



Developer Reference Manual



Toolkit Developer Reference Manual

Note

Before using this information and the product it supports, read the information in "Notices" on page 570

This edition applies to version 4.1.2 of IBM FileNet Image Services Toolkit (part number 5724-R95) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 1984, 2008.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

TABLE OF CONTENTS

0. REVISION HISTORY	1
1. INTRODUCTION	2
2. CONFIGURATION NOTES	3
3. PROGRAMMING NOTES	12
4. CONCEPTS	15
5. MODULE SUMMARY	20
6. BES DECLARATIONS	23
7. BES SUBROUTINES	45
7.1. BES_abort_image	45
7.2. BES_abort_image_companion	46
7.3. BES_alloc_batch_name	46
7.4. BES_alloc_ids	46
7.5. BES_close_batch	47
7.6. BES_close_connection	47
7.7. BES_close_csum_image	47
7.8. BES_close_image	48
7.9. BES_close_image_companion	49
7.10. BES_commit_batch_now	49
7.11. BES_create_batch	50
7.12. BES_create_doc	51
7.13. BES_create_image	52
7.14. BES_create_image_companion	53
7.15. BES_create_image_index	54
7.16. BES_delete_batch	54
7.17. BES_delete_doc	55
7.18. BES_delete_image	55
7.19. BES_delete_image_companion	56
7.20. BES_delete_image_index	56
7.21. BES_delete_object_data	57
7.22. BES_enqueue_batch	58
7.23. BES_extended	59
7.24. BES_find_batches	60
7.25. BES_find_docs	61
7.26. BES_find_images	62

7.27.	BES_find_object_data	62
7.28.	BES_folder_create_node	64
7.29.	BES_folder_delete_node	65
7.30.	BES_folder_find_children	66
7.31.	BES_folder_get_node	67
7.32.	BES_folder_update_node	68
7.33.	BES_free_info_rec	69
7.34.	BES_free_object_data	69
7.35.	BES_get_image_companion_ids	70
7.36.	BES_get_image_index	70
7.37.	BES_get_info	71
7.38.	BES_logoff	72
7.39.	BES_logon	72
7.40.	BES_logon_extended	73
7.41.	BES_modify_image_index	73
7.42.	BES_move_doc	74
7.43.	BES_move_image	75
7.44.	BES_open_batch	75
7.45.	BES_open_connection	76
7.46.	BES_open_csum_image	76
7.47.	BES_open_image	77
7.48.	BES_open_image_companion	78
7.49.	BES_put_info	79
7.50.	BES_put_object_data	79
7.51.	BES_query_index_dir	80
7.52.	BES_read_image	80
7.53.	BES_read_image_companion	81
7.54.	BES_return_docs	82
7.55.	BES_return_index_dir	82
7.56.	BES_sync_commit	82
7.57.	BES_update_batch	83
7.58.	BES_update_doc	83
7.59.	BES_update_image	84
7.60.	BES_update_image_companion	85
7.61.	BES_update_index_total	86
7.62.	BES_verify_image	86
7.63.	BES_write_image	87
7.64.	BES_write_image_companion	88
8.	CKS DECLARATIONS	89
9.	CKS SUBROUTINES	89
9.1.	CKS_compute_csum	89
10.	CSM DECLARATIONS	91
11.	CSM SUBROUTINES	99
11.1.	CSM_breakup_object	100

11.2.	CSM_chg_refcnt	102
11.3.	CSM_close_connection	102
11.4.	CSM_close_csum_object	103
11.5.	CSM_close_delete_object	103
11.6.	CSM_close_object	103
11.7.	CSM_copy_object	104
11.8.	CSM_create_object	105
11.9.	CSM_delete_object	106
11.10.	CSM_find_objects	107
11.11.	CSM_get_cache_attributes	108
11.12.	CSM_get_obj_attr_2	108
11.13.	CSM_get_object_attributes	109
11.14.	CSM_logoff	109
11.15.	CSM_logon	109
11.16.	CSM_lookup_cache	110
11.17.	CSM_modify_object_attributes	110
11.18.	CSM_move_object	111
11.19.	CSM_open_connection	112
11.20.	CSM_open_csum_object	112
11.21.	CSM_open_object	113
11.22.	CSM_read_object	114
11.23.	CSM_rename_object	115
11.24.	CSM_resize_object	115
11.25.	CSM_write_object	116
12.	DOC DECLARATIONS	118
13.	DOC SUBROUTINES	140
13.1.	DOC_alloc_queue	140
13.2.	DOC_allocate_ids	141
13.3.	DOC_async_migrate_from_od	141
13.4.	DOC_batch_abort	143
13.5.	DOC_batch_close	143
13.6.	DOC_batch_create	143
13.7.	DOC_batch_get_csum_doc_hdr_size	145
13.8.	DOC_batch_get_doc_hdr_size	146
13.9.	DOC_batch_set_checksum	146
13.10.	DOC_batch_write_doc_hdr	147
13.11.	DOC_batch_write_page	148
13.12.	DOC_cancel_async_migrate	149
13.13.	DOC_cancel_migrate	149
13.14.	DOC_cancel_migrate_from_surf	149
13.15.	DOC_close_connection	150
13.16.	DOC_commit_document	150
13.17.	DOC_commit_option_doc	151
13.18.	DOC_create_annotation	152
13.19.	DOC_create_annotation_id	152
13.20.	DOC_delete_annotation	154

13.21.	DOC_delete_documents	155
13.22.	DOC_find_by_family_id	155
13.23.	DOC_find_by_family_name	156
13.24.	DOC_find_documents	156
13.25.	DOC_find_documents_w_sds	157
13.26.	DOC_find_surfaces	158
13.27.	DOC_free_queue	158
13.28.	DOC_get_and_lock_annotation	159
13.29.	DOC_get_annotations	159
13.30.	DOC_get_boot_number	161
13.31.	DOC_get_doc_surf_location	161
13.32.	DOC_get_ssn_of_docs	162
13.33.	DOC_is_annotated	162
13.34.	DOC_is_migrated	163
13.35.	DOC_logoff	163
13.36.	DOC_logon	164
13.37.	DOC_migrate_from_optical_disk	164
13.38.	DOC_migrate_to_optical_disk	166
13.39.	DOC_migrate_w_ssn_from_od	166
13.40.	DOC_modify_document_attribute	168
13.41.	DOC_open_connection	169
13.42.	DOC_prefetch_from_optical_disk	169
13.43.	DOC_read_queue	171
13.44.	DOC_set_surface_inbox_priority	172
13.45.	DOC_update_annotation	172
14.	DTI DECLARATIONS	174
15.	DTI SUBROUTINES	181
15.1.	DTI.UTC_to_local_struct	181
15.2.	DTI_addtime	182
15.3.	DTI_addtimeUTC	182
15.4.	DTI_dtnum_to_string	184
15.5.	DTI_dtnum_to_string_no_pad	184
15.6.	DTI_dtnum_to_struct	184
15.7.	DTI_get_cur.UTC_struct	185
15.8.	DTI_get_cur.datetime_num	185
15.9.	DTI_get_cur.datetime_string	186
15.10.	DTI_get_cur.datetime_struct	186
15.11.	DTI_getmask	186
15.12.	DTI_local_to.UTC_struct	187
15.13.	DTI_make4y	187
15.14.	DTI_mask_strlen	188
15.15.	DTI_mask_value	188
15.16.	DTI_mask_verify	188
15.17.	DTI_oldmask	189
15.18.	DTI_string_to_dtnum	189
15.19.	DTI_string_to_dtnum_c	190

15.20.	DTI_string_to_struct	190
15.21.	DTI_string_to_struct_c	191
15.22.	DTI_struct_to_dtnum	191
15.23.	DTI_struct_to_string	191
15.24.	DTI_struct_to_string_no_pad	192
15.25.	DTI_subtime	192
15.26.	DTI_subtimeUTC	193
16.	FFI DECLARATIONS	195
17.	FFI SUBROUTINES	204
17.1.	FFI_annot_decode	204
17.2.	FFI_annot_encode	205
17.3.	FFI_annot_free	205
17.4.	FFI_build_uncomp_text_header	206
17.5.	FFI_get_page_attr	206
17.6.	FFI_get_page_type	207
17.7.	FFI_image_decode	207
17.8.	FFI_image_encode	208
17.9.	FFI_text_decode	209
17.10.	FFI_text_encode	209
18.	FP DECLARATIONS	211
19.	FP SUBROUTINES	214
19.1.	FP_abs	214
19.2.	FP_add	214
19.3.	FP_assign	215
19.4.	FP_dbl2num	215
19.5.	FP_divide	215
19.6.	FP_eql	215
19.7.	FP_geq	215
19.8.	FP_getmode	216
19.9.	FP_gtr	216
19.10.	FP_isundef	216
19.11.	FP_leq	216
19.12.	FP_long2num	217
19.13.	FP_lss	217
19.14.	FP_multiply	217
19.15.	FP_neg	217
19.16.	FP_neq	217
19.17.	FP_num2dbl	218
19.18.	FP_num2long	218
19.19.	FP_num2mask	218
19.20.	FP_num2sci	219
19.21.	FP_num2str	220
19.22.	FP_num2unslong	220
19.23.	FP_parsenum	220

19.24.	FP_retsign	220
19.25.	FP_round	221
19.26.	FP_rounddisp	221
19.27.	FP_setmode	221
19.28.	FP_setundef	221
19.29.	FP_str2num	221
19.30.	FP_subtract	222
19.31.	FP_trunc	222
19.32.	FP_unslong2num	222
20.	IMS DECLARATIONS	223
21.	IMS SUBROUTINES	224
21.1.	IMS_free_domain_names	224
21.2.	IMS_free_organization_names	224
21.3.	IMS_free_svc_names	224
21.4.	IMS_get_message	224
21.5.	IMS_get_svc_handle	225
21.6.	IMS_get_version_number	226
21.7.	IMS_list_domain_names	226
21.8.	IMS_list_organization_names	227
21.9.	IMS_list_svc_names	227
21.10.	IMS_logoff	228
21.11.	IMS_logon	228
21.12.	IMS_release_svc_handle	229
21.13.	IMS_reserve_svc_handle	229
21.14.	IMS_shmat	230
21.15.	IMS_str_to_svc	230
21.16.	IMS_svc_to_str	230
22.	INX DECLARATIONS	232
23.	INX SUBROUTINES	269
23.1.	INX_check_dict_ts	269
23.2.	INX_close_DIR	269
23.3.	INX_close_connection	269
23.4.	INX_close_folder	270
23.5.	INX_copy_folder	270
23.6.	INX_create_DIR	271
23.7.	INX_create_folder	271
23.8.	INX_delete_DIR	272
23.9.	INX_delete_DIRS	272
23.10.	INX_delete_folder	273
23.11.	INX_delete_folders_by_id	273
23.12.	INX_dir_to_doc_desc	274
23.13.	INX_dir_to_doc_desc2	274
23.14.	INX_file_doc	274
23.15.	INX_file_doc_after	275

23.16.	INX_find_DIRS	275
23.17.	INX_find_folders	276
23.18.	INX_free_DIR	277
23.19.	INX_free_folder_ids	277
23.20.	INX_free_folders	277
23.21.	INX_free_matches	278
23.22.	INX_get_all_cluster_descs	278
23.23.	INX_get_all_dcl_descs	278
23.24.	INX_get_all_index_descs	279
23.25.	INX_get_all_key_descs	279
23.26.	INX_get_all_menu_descs	279
23.27.	INX_get_and_lock_folder	280
23.28.	INX_get_cluster_desc	280
23.29.	INX_get_dcl_desc	281
23.30.	INX_get_dict_ts	281
23.31.	INX_get_docs_in_folder	282
23.32.	INX_get_folder_atts	282
23.33.	INX_get_folder_content_desc	283
23.34.	INX_get_folders_by_archivedate	284
23.35.	INX_get_index_desc	284
23.36.	INX_get_key_desc	285
23.37.	INX_get_menu_desc	285
23.38.	INX_get_num_cluster_id	286
23.39.	INX_get_single_DIR	286
23.40.	INX_get_single_match	287
23.41.	INX_get_str_cluster_id	287
23.42.	INX_import_DIRS	287
23.43.	INX_index_data_size	289
23.44.	INX_interrupt_query	290
23.45.	INX_logoff	290
23.46.	INX_logon	290
23.47.	INX_move_folder	291
23.48.	INX_open_connection	291
23.49.	INX_reorder_folder	292
23.50.	INX_terminate_query	292
23.51.	INX_unfile_doc	292
23.52.	INX_unfile_docs	293
23.53.	INX_update_DIR	293
23.54.	INX_update_folder	294
23.55.	INX_where_filed	295
24.	PRI DECLARATIONS	296
25.	PRI SUBROUTINES	320
25.1.	PRI_cancel_request	320
25.2.	PRI_close_connection	320
25.3.	PRI_find_requests	320
25.4.	PRI_find_requests2	321

25.5.	PRI_free_find_requests2	323
25.6.	PRI_free_printer_attrs	323
25.7.	PRI_free_printer_attrs2	323
25.8.	PRI_free_printer_status	324
25.9.	PRI_free_service_status	324
25.10.	PRI_get_cache_name	324
25.11.	PRI_get_printer_attrs	324
25.12.	PRI_get_printer_attrs2	325
25.13.	PRI_get_printer_status	325
25.14.	PRI_get_service_status	326
25.15.	PRI_is_printed	326
25.16.	PRI_logoff	326
25.17.	PRI_logon	326
25.18.	PRI_modify_request	327
25.19.	PRI_open_connection	327
25.20.	PRI_print_data	328
25.21.	PRI_print_docs	328
26.	PRS DECLARATIONS	330
27.	PRS SUBROUTINES	338
27.1.	PRS_free_query	338
27.2.	PRS_parse	338
28.	SEC DECLARATIONS	342
29.	SEC SUBROUTINES	407
29.1.	SEC_add_member	407
29.2.	SEC_add_object	407
29.3.	SEC_change_password	409
29.4.	SEC_change_password_without_logon	410
29.5.	SEC_check_access	410
29.6.	SEC_check_membership_wid	411
29.7.	SEC_cleanup_invalid_logons	411
29.8.	SEC_delete_member	411
29.9.	SEC_delete_object	412
29.10.	SEC_find_groups_owning_member	412
29.11.	SEC_find_members_of_group	413
29.12.	SEC_find_object	414
29.13.	SEC_find_object_without_membership	416
29.14.	SEC_find_system_defaults	417
29.15.	SEC_free	417
29.16.	SEC_gateway_logon	418
29.17.	SEC_get_concurrent_limits	419
29.18.	SEC_get_id	420
29.19.	SEC_get_last_logon_info	420
29.20.	SEC_get_message_without_logon	420
29.21.	SEC_get_name	421

29.22.	SEC_get_security_info	421
29.23.	SEC_get_security_info_using_cache	421
29.24.	SEC_get_service	422
29.25.	SEC_get_stats	422
29.26.	SEC_id_to_name	423
29.27.	SEC_logoff	423
29.28.	SEC_logon	424
29.29.	SEC_name_to_id	428
29.30.	SEC_pooled_logon	429
29.31.	SEC_shared_logon	431
29.32.	SEC_update_object	434
29.33.	SEC_update_system_defaults	435
29.34.	SEC_validate_session	436
30.	SQI DECLARATIONS	437
31.	SQI SUBROUTINES	446
31.1.	SQI_begin_transaction	446
31.2.	SQI_bind_result	447
31.3.	SQI_bind_value	448
31.4.	SQI_create_cursor	448
31.5.	SQI_dbms_version	449
31.6.	SQI_describe	449
31.7.	SQI_drop_cursor	450
31.8.	SQI_exec_cursor	450
31.9.	SQI_exec_sql	451
31.10.	SQI_fetch	451
31.11.	SQI_full_use_logon	452
31.12.	SQI_logoff	452
31.13.	SQI_max_blob_size	453
31.14.	SQI_n_describe	453
31.15.	SQI_prepare_blob_update	453
32.	WQS DECLARATIONS	455
33.	WQS SUBROUTINES	466
33.1.	WQS_close_queue	466
33.2.	WQS_count_entries	467
33.3.	WQS_create_queue	467
33.4.	WQS_create_workspace	467
33.5.	WQS_delete_and_insert_entry	468
33.6.	WQS_delete_entry	469
33.7.	WQS_delete_queue	469
33.8.	WQS_delete_workspace	469
33.9.	WQS_empty_queue	470
33.10.	WQS_end_dump	470
33.11.	WQS_flush_act_log	471
33.12.	WQS_get_queue_desc	471

33.13.	WQS_get_queue_names	471
33.14.	WQS_get_server_name	472
33.15.	WQS_get_table_name	473
33.16.	WQS_get_workspace_info	473
33.17.	WQS_get_workspace_names	473
33.18.	WQS_insert_entry	474
33.19.	WQS_logoff	474
33.20.	WQS_logon	475
33.21.	WQS_open_queue	475
33.22.	WQS_qlogon	476
33.23.	WQS_read_dump	476
33.24.	WQS_read_entry	477
33.25.	WQS_read_queue	477
33.26.	WQS_retarea	478
33.27.	WQS_start_dump	479
33.28.	WQS_update_entry	479
33.29.	WQS_update_queue	480
33.30.	WQS_update_selected	481
33.31.	WQS_update_workspace	482
34.	MISC DECLARATIONS	483
APPENDIX A:	INDEX	502
APPENDIX B:	NOTICES	570

0. REVISION HISTORY

- Z3 - 10/08/08 GTG Add comments to PRS_parse that F_PAGES with NULL value really means single page.
- Z2 - 09/10/08 CAB Add comments to DOC_prefetch_from_optical_disk.
- Z1 - 08/20/08 GTL Limited PRS_parse filter string conditions.
- Z - 08/01/08 GTL Updated to 4.1.2 levels.
Updated to meet Bluewash requirements.
Added 'Notices' Appendix.
- Y - 10/18/07 YJC Updated all entrypoints to 4.1.1 levels.
- X - 10/18/07 YJC Added Solaris10 section for kernel parameters.
- W - 09/18/07 GTL Removed INX_QUERY_BACKWARD define.
Added PRS_parse key and filter constraints comments.
- V - 05/22/07 JTL Updated all entrypoints to 4.1.0 levels.
- U - 08/14/06 CAB Update INX_delete_DIR comment.
- T - 08/19/05 AP Made INX entry point correction.
- S - 07/20/05 YJC Made 4.0.40 Updates.
- R - 02/22/05 JNL Removed all references of SEC_remove_inheritance. This function is no longer available in WAL.
- Q - 01/20/05 GTL More misc updates for 4.0.30. Replaced "NT" with "Windows".
Reformat page margins to accommodate Windows printout.
Updates kernel parameters settings.
- P - 12/09/04 RNF Updated for 4.0.30. Updated WAL_ROOT.
Added shared memory conflict.
Replaced "WAL" with "ISTK".
- O - 06/10/04 CAS Updated all doc files and WAL manual to 4.0.20.
- N - 12/18/02 JNL Removed unsupported entry point.
- M - 07/16/02 CAS Updated all doc files and WAL manual to 4.0.0.
- L - 03/26/02 JNL Updated all readme files and WAL manual to 3.6.30.
- K - 06/07/01 JNL Updated all entrypoints and doc to 3.6.10 levels.
- J - 01/04/01 JNL Updated all entrypoints and doc to 3.6.0 levels.
- I - 08/08/00 JNL Updated all entrypoints to 3.5.0 levels.
- H - 06/30/97 SKN Updated all entrypoints to 3.4.0 levels.
- G - 12/19/96 SKN Updated all entrypoints to 3.3.1 levels.
- F - 06/13/95 KBB Added discussion of WAL_ROOT, LD_LIBRARY_PATH and SHLIB_PATH. Updated SEC_find_object() comments to include discussion of SEC_OPTOBJ_CLASS_KEY.
- E - 06/01/95 KBB Removed SQI_logon. Only SQI_full_use_logon is used for logging on to SQI. Added discussion of hosts and service file updates.
- D - 05/20/95 KBB Added discussion of kernel parameter values, error log use, and IPC key values.
- C - 03/03/95 KBB Removed discussion about all processes running as same user.

Added discussion of wal_daemon and wal_purge.
B - 01/17/95 KBB Added SQI section. Updated all entrypoints to 3.2.0 levels.
A - 02/17/93 SRT Clarify index services. Added FFI description.
Added SEC_name_to_id and SEC_id_to_name.
X3 - 01/28/93 SRT Clarify requirements for batch committal in BES.
X2 - 01/03/93 SRT Added concepts section
X1 - 12/05/92 SRT Initial Draft

1. INTRODUCTION

This document contains descriptions of the ISTK formerly known as WorkFlo Application Library (WAL) subroutines and declarations used by those subroutines. These subroutines provide a broad variety of functions to support the storage, indexing, transfer, and retrieval of data objects.

The first few sections of this document make general comments about using this toolkit. These sections should be read from start to finish, since they contain material you need to know for the remainder of the document. The remaining sections describe the entry points supported in alphabetical order. You will probably want to skip around in these sections to look up entry points which are of interest to you.

This document has a cross reference index at the end, and also cross references each use of a data structure or type by placing:

"(pg<pagenum>.<linenum>)" in the right hand margin past column 80. <pagenum> is the page number where the item is defined, and <linenum> is the line number on that page. The right hand margin cross reference can be used with a paper copy of this document to quickly find a referenced definition.

If you electronically view this document with a terminal which has a width of only 80 characters, you may want to create a second copy of this document by running column remove to strip off characters past column 80 (the cross reference material). Doing so will prevent the display from wrapping lines, and the material lost is not significant since you can use a search command to find needed information. The strip command is "colrm 81 <filename1 >filename2" where "filename1" is the input file and "filename2" is the output file. (Note that 'colrm' command can only be executed on an AIX platform.)

Viewing this document with a Windows(R) editor such as Microsoft(R) Word or WordPad, you may want to reduce the margins to prevent the display or hardcopy printout from wrapping lines. Select 'Page Setup' under the 'File' tool menu and adjust the left or right margin to a value that will work for your environment.

2. CONFIGURATION NOTES

Services and Hosts File Updates

The following entries should be added to the /etc/services file for UNIX(R) or \$(ETC_PATH)/services file for Windows (where \$(ETC_PATH) maps to drive:\Windows_root_directory\system32\drivers\etc location):

```
tms      32768/tcp
cor      32769/tcp
nch      32770/udp
```

(If you are installing on a system already running the FileNet(R) IMS, these entries should already exist.)

Any host running the FileNet IMS that the ISTK applications will communicate with should also be listed in the /etc/hosts file for UNIX or \$(ETC_PATH)/hosts file for Windows. In addition, when installing ISTK in a network environment which uses a router with broadcast disabled, the following should be added for each target IMS in order for proper NCH operation to occur:

```
<IP addr of domain:organization> <domain-organization>-nch-server
```

where <IP addr of domain:organization> is the IP address of the FileNet domain and organization identifying the target IMS system, and <domain-organization> is that same domain and organization in lower case and separated with a '-' character. For example, if the domain "Mydom" and organization "Myorg" was at the IP address 123.45.6.78 the entry would look as follows:

```
123.45.6.78    mydom-myorg-nch-server
```

WAL_ROOT

By default ISTK is installed into the following directory path:

```
UNIX: /fnsw/client
```

```
Windows: C:\Program Files\FileNet\FNSW\client
```

On Solaris, HPUX, and Windows this default install directory can be changed (but not on AIX).

The Windows installation automatically saves the installation

directory path in a registry value named WAL_ROOT. This may be found in the registry as:

```
HKEY_LOCAL_MACHINE\SOFTWARE\FileNet\WAL\CurrentVersion\WAL_ROOT
```

This registry value MUST be set for ISTK to function on Windows systems.

The default value for WAL_ROOT on Windows is:

```
Windows default WAL_ROOT = "C:\Program Files\FileNet\FNSW"
```

On Solaris and HPUX systems WAL_ROOT only needs to be set if ISTK is NOT installed in the above default location. On such non-standard installations the user is responsible for changing their environment to include WAL_ROOT. See the section below on "Moving the /fnsw/client directory" for more details on Solaris and HPUX WAL_ROOT settings.

Examples of paths are given in the remainder of this document. You can substitute your value for WAL_ROOT in these examples to locate the file in question. If your setting of WAL_ROOT is identical to the default ISTK installation directory, then you can use the default examples given.

Interprocess Communication Parameters

The Image Services Toolkit (ISTK) make heavy use of various aspects of SystemV Interprocess Communication (IPC) constructs. Proper measures should be taken to ensure the operating system can handle the amount of IPC resources necessary for proper operation. The three IPC resources are shared memory, semaphores, and message queues. Listed below are the minimum values for the various resources.

AIX:

On AIX systems, the IPC resources are defaulted to their max values. As a result, the kernel does not need to be tuned to have these values increased. The values used by the ISTK software are listed here for completeness.

Shared Memory:

Segment size (SHMMAX): 2GB

Semaphores:

Identifiers (SEMMNI): 131072

Message Queues:

Identifiers (MSGMNI): 131072

HPUX:

On HPUX systems, the only item that typically needs to be increased is the maximum number of message queue identifiers. This value defaults to 50 and should be increased to 2000. The kernel can be tuned by running the 'sam' utility. Please refer to the documentation for the 'sam' utility for details on configuring the kernel.

Shared Memory:

Segment size (SHMMAX): 16MB

Semaphores:

Identifiers (SEMMNS): 500

Identifiers (SEMMNI): 500

Message Queues:

Identifiers (MSGMNI): 2048

Solaris:

On Solaris systems, it has been found that the operating system does not allow a shared memory segment to be configured above 16MB. As a result, the segment size is 16MB, and the variable associated with the number of segments per process should be set to a significant value (15 to 20). The program 'sysdef' can be used to get the current values for these parameters. If changes

are necessary, the /etc/system file should be updated with the appropriate values and the system should be rebooted. Please refer to the system documentation before making updates to the /etc/system file.

Shared Memory:

Segment size (SHMMAX): 16MB

Segments per process (SHMSEG): 15 to 20

Semaphores:

Identifiers (SEMMNI): 500

Identifiers (SEMMNS): 500

Identifiers (SEMMSL): 512

Message Queues:

Identifiers (MSGMNI): 2048

Solaris10:

On Solaris10 systems, many kernel parameters are either automatically configured or can be controlled by resource controls. Obsolete parameters can still be included in the /etc/system file on a Solaris system. If so, the parameters are used to initialize the default resource control values as in the previous releases. To check the kernel parameter settings, use 'prctl \$\$' in the shell prompt. If you don't set values, it will use the default values. For more detailed descriptions of the resource controls, see "Solaris Tunable Parameters Reference Manual". Note: SHMSEG and SEMMNS are removed in Solaris 10. The project.max-shm-memory resource control limits the total amount of shared memory of one project, whereas previously, the shmsys:shminfo_shmmax parameter limited the size of a single shared memory segment.

Shared Memory in Solaris10:

default value (1/4 of physical memory)

Semaphores in Solaris10:

Identifiers (SEMMNI or project.max-sem-ids): 500

Identifiers (SEMMSL or process.max-sem-nsems): default value(512)

Message Queues in Solaris10:

Identifiers (MSGMNI or project.max-msg-ids): 2048

Shared Memory Segment Size

The size of a shared memory segment created by ISTK can be configured. By default it is 16 MB. To change this default size, create the following file containing the desired size (in megabytes):

Windows: WAL_ROOT\client\tmp\FNSHMSEGSZ

Example: C:\Program Files\FileNet\FNSW\client\tmp\FNSHMSEGSZ

UNIX: WAL_ROOT/tmp/FNSHMSEGSZ

Example: /fnsw/client/tmp/FNSHMSEGSZ

FNSHMSEGSZ is short for "FileNet Shared Memory Segment Size"

For example, to set the shared memory segment size at 20 megabytes (assuming ISTK is installed in the default path), do the following steps:

Completely terminate all ISTK applications.

UNIX:

```
/fnsw/client/wal_purge
```

```
cd /fnsw/client/tmp
```

```
echo 20 > FNSHMSEGSZ
```

Windows:

```
cd "C:\Program Files\FileNet\FNSW\client\tmp"
```

```
echo 20 > FNSHMSEGSZ
```

Run `wal_ipc` and select menu option "A" to verify that the new size is used. This option also shows how many shared memory segments are currently in use by ISTK. The option can also be run from the command prompt:

```
wal_ipc -A
```

Be careful not to use the lower-case 'a' option, as this will not give the desired results.

System event/error logs

When any of the ISTK libraries encounter an error, they generate what is known as a `sys_log` entry (system log entry). In the IS release, these entries are sent to the standard error and also to the file:

```
UNIX: /fnsw/local/logs/elogs/elogYYYYMMDD
```

```
Windows: \fnsw_loc\logs\elogs\elYYYYMMDD
```

The `sys_log` messages generated from the ISTK libraries are also sent to a file. By default, this file is named:

```
UNIX: /fnsw/client/logs/walYYYYMMDD
```

```
Windows: C:\Program Files\FileNet\FNSW\client\walYYYYMMDD
```

The location of this file can be changed by setting the environment variable `WAL_LOG_DIR` to specify a different directory. NOTE: The filename will still be `walYYYYMMDD`!

WINDOWS example: setting environment `WAL_LOG_DIR` to `C:\logs\mylogs` creates log files named `C:\logs\mylogs\walYYYYMMDD`.

UNIX example: setting environment `WAL_LOG_DIR` to `/logs/mylogs` creates log files named `/logs/mylogs/walYYYYMMDD`.

Note: `WAL_LOG_DIR` overrides `WAL_ROOT`. If `WAL_LOG_DIR` is set, then `WAL_ROOT` is ignored when determining where to write the ISTK `sys_log` file.

WINDOWS systems can be configured to automatically add a ".txt" extension to the ISTK log file. This allows the user to quickly open

the file by clicking on it in Windows Explorer. To automatically add the ".txt" extension, create the following "trigger" file:

Windows: WAL_ROOT\client\tmp\syslog.txt

Example: C:\Program Files\FileNet\FNSW\client\tmp\syslog.txt

UNIX: WAL_ROOT/tmp/syslog.txt

Example: /fnsw/client/tmp/syslog.txt

Note: Writing to the ISTK sys_log file can be tested by running the wal_ipc tool. When this tool is executed, it automatically creates an "information-only" message in the ISTK log stating that the tool has executed. The wal_ipc utility presents the user with a selection menu. For purposes of testing the sys_log entry, no selection is necessary - simply enter "q" to quit from the menu.

Output to stderr can be turned off by creating the file:

Windows: WAL_ROOT\client\tmp\nocons

Example: C:\Program Files\FileNet\FNSW\client\tmp\nocons

UNIX: WAL_ROOT/tmp/nocons

Example: /fnsw/client/tmp/nocons

If this file exists, sys_log entries will ONLY be written to the system log file, not to stderr.

It is the responsibility of the site administrator for the ISTK software to manage the filesystem where the ISTK sys_log files are located. Over time the number of files can grow quite large, so measures should be taken to either backup the files and/or remove them from the corresponding directory.

Moving the /fnsw/client directory

If installed on Solaris or HP-UX systems, the /fnsw/client directory tree can be relocated to another directory. This can be accomplished by setting the WAL_ROOT environment variable to the new location. Please note that all subdirectories under the current /fnsw/client directory must remain in their respective places relative to the new "root" location. In order for the shared libraries to be found at runtime, however, the associated platform's shared library path variable must also be set. Consider the following example:

The user wishes to move the /fnsw/client tree to /walproduct. They must set their WAL_ROOT environment variable to (csh is assumed):

```
setenv WAL_ROOT /walproduct
```

For example, with the above value for WAL_ROOT the ISTK shared libraries would be found in:

```
/walproduct/shobj
```

Next the platform's shared library lookup variable must be set. On HPUX this is SHLIB_PATH while on Solaris this is LD_LIBRARY_PATH. NOTE: Remember, this option is only available to HPUX and Solaris platforms.

```
setenv SHLIB_PATH ${WAL_ROOT}/shobj /* for HPUX */
```

```
setenv LD_LIBRARY_PATH ${WAL_ROOT}/shobj /*for Solaris */
```

Once this setup has been performed, the user can operate out of the the /walproduct "tree". Also note that any setting of WAL_LOG_DIR (described above) will override the setting of WAL_ROOT.

WARNING: On Solaris systems, setting LD_LIBRARY_PATH to locate shared libraries does not get used during the shared library resolution of set-uid and set-gid programs. As a result, the programs wal_purge and wal_daemon must be executed by the root user (with the proper LD_LIBRARY_PATH and/or WAL_ROOT settings). Since wal_daemon is a background process which monitors shared resource activity (message queue usage), it must be invoked prior to all other ISTK applications. To do this, run "nohup wal_daemon >& <outputfile> &". NOTE: This is only a requirement if moving the /fnsw/client tree, and is not an issue on HPUX systems.

The user must be sure they do not have multiple copies of the ISTK product installed in different locations on the same machine. Mixing ISTK applications using different WAL_ROOT locations on the same machine is NOT supported.

On AIX systems ISTK must be installed in the default location /fnsw/client. It cannot be moved.

On Windows you can choose where to install the ISTK during installation. It is not advised to move the install directories after installation. To move ISTK on Windows systems, you would need to uninstall current version, then re-install to the new location.

Windows Shared Memory Conflicts

Sometimes the addresses that ISTK uses for shared memory may conflict with other third-party libraries that are also linked to the ISTK application. When such a problem occurs, ISTK will generate errors when trying to create a new shared memory segment. The particular error displayed in the ISTK error log will indicate some kind of problem with shared memory. For example:

```
fnc_shmget failed for key=0xcfaa0000 from_catch_segv=0
```

First, the addresses used by ISTK for shared memory can be examined using `wal_ipc`. Terminate any running ISTK applications then run `"wal_ipc -A"` (make sure to use a capital "A", not lower case). The output shows information on ISTK's use of shared memory, including the number of shared memory segments and their addresses.

For example, the output may appear as:

```
Current configured segment size: 0x01000000 bytes (16 MB)
```

```
Shared Memory Address Manager Information
```

Address	Shm id	Creator
0 0x45000000	FNSHM_464e0000	Shared address manager
1 0x46000000	FNSHM_cfaa0000	FileNet client software
2 0x47000000	FNSHM_cfaa0001	FileNet client software

This example shows three ISTK shared memory segments along with their starting addresses. The first segment (called the address manager) at address 0x45000000 is small and only contains information used by ISTK to locate the other segments. The other two segments are the real data segments. As shown by the `wal_ipc` output, each of these segments is 16 MB in size (the segment size can be modified - see the discussion of `FNSHMSEGSZ` in the section "Shared Memory Segment Size" above).

Any of these shared memory addresses may conflict with the base address of other non-ISTK libraries linked to the ISTK applications.

There are three possible ways to avoid shared memory conflicts.

1. Use the Microsoft "rebase" command to change the base loader addresses of the other .dll files linked to the application. See the Microsoft MSDN documentation for help on this problem.

2. Change the starting shared memory address used by SysV for ISTK shared memory. The start shared memory address is the address listed above in the wal_ipc output for the address manager segment (0x45000000). This start address can be changed by creating the following registry value under HKEY_LOCAL_MACHINE:

```
SOFTWARE\FileNet\WAL\CurrentVersion\StartShmAddress
```

This entry must be type REG_DWORD and it should contain the hex value of the first shared memory segment address. For example, StartShmAddress could be set to 0x48000000.

Note: All ISTK applications must be stopped before changing StartShmAddress!

Once set, the new start address can be verified by running "wal_ipc -A" as above.

3. By default ISTK only allocates its shared memory when it is necessary to do so. So if some other library has already used the address required by ISTK, then ISTK fails.

ISTK can be configured to allocate its shared memory segments immediately as the ISTK application is loaded. The idea here is that once ISTK has setup its shared memory, then any other library using the same address will automatically get based at some other address by Windows, thus avoiding the conflict.

To force ISTK to immediately allocate its shared memory, create the following registry entry under HKEY_LOCAL_MACHINE:

```
SOFTWARE\FileNet\WAL\CurrentVersion\SharedMemSegments
```

This entry must be type REG_DWORD. Set it to 1 to enable memory allocation. Set it to 0 (zero) to disable memory allocation (the default behavior).

}

3. PROGRAMMING NOTES

UNIX Interprocess Communication

This section applies to UNIX ISTK only.

In order for interprocess communication to work properly, the program /fnsw/client/bin/wal_daemon must run as the root user. It is

responsible for monitoring unused IPC resources created on behalf of the ISTK applications. When it has determined a need to clean up these resources, wal_daemon must be able to verify that these resources are no longer in use, and removes them accordingly. In addition, the program /fnsw/client/bin/wal_purge should also run as the root user. This program is used to clear out ISTK shared memory, semaphores, and message queues completely! No ISTK applications should be running when this program is executed. These programs can be modified by setting the owner to root (using chown) and the set-user-id bit (using chmod).

If a program allocates shared memory, the virtual address of this memory must not conflict with virtual addresses of shared memory allocated by ISTK. Use "IMS_shmat" to set the virtual address to work properly with ISTK. If you use the system shmat call instead, you may get aborts from ISTK subroutines which try to access ISTK shared memory.

The ISTK libraries make use of various SystemV IPC keys. These same key values must not be used by the ISTK programmer. The keys are:

UNIX Shared Memory:

0x464d0000 shared memory address manager segment
0xcfaa000# data segments ('#' specifies a number starting at 0)

UNIX Semaphore sets:

0x464d0000 semaphore set for address manager.
0xcfba0000 semaphore set for ISTK libraries.
0xcfba0004 semaphore set for ISTK log files.

The program wal_purge will remove the segments 0xcfaa000# and the semaphore sets 0xcfba0000 and 0xcfba0004. It will not remove the segment and semaphore set associated with the address manager (0x464d0000) since that segment and semaphore set could be in use by the IMS which is also running on the same server. In addition, if the program creates its own shared memory segments, the virtual addresses for these segments are stored in the address manager. The IMS_shmat call is responsible for inserting these addresses into the address manager so that the virtual addresses do not conflict with those used by the ISTK libraries. If a need arises to remove these IPC keys, the user must use the native utility "ipcrm" to do so.

Please refer to your local system documentation for how to use "ipcrm".

Windows Interprocess Communication

This section applies to Windows ISTK only.

ISTK on Windows also uses many interprocess resources. However, unlike the UNIX version of ISTK, there is no `wal_daemon` or `wal_purge` process (as described in the previous UNIX IPC section). The IPC resources created by ISTK on Windows are automatically removed by the Windows operating system when all of the ISTK processes terminate. There is no need to run a "wal_purge" on Windows to cleanup after ISTK.

ISTK uses mapped files as shared memory. If a program allocates a mapped file, the virtual address of this mapped file must not conflict with virtual addresses of mapped files allocated by ISTK. If you use the system `CreateFileMapping` and `MapViewOfFileEx` calls, you may get aborts from ISTK subroutines which try to access ISTK mapped files. If you have problems with mapped files, use "IMS_shmat" to set the virtual address to work properly with ISTK. This routine will check the mapped file space used by ISTK and assign an address that does not conflict with ISTK.

Logon and Logoff

The first access across the network via an ISTK subroutine must be either an `SEC_logon` or `IMS_logon` (which calls `SEC_logon`) call. Failure to do so will cause a security error to be returned from the improperly called routine. Note that an access across the network may occur whenever a handle returned by a "logon" routine is input to an entry point. The modules "FP", "CKS" don't use handles so these are the only exceptions to this restriction.

`IMS_logoff` or `SEC_logoff` should be called after (not before or during) all accesses across the network. Calling `SEC_logoff` prior to making an access across the network may cause a security error (see the description of `SEC_logon` and `SEC_logoff` for more details).

The handles returned by any "logon" routine (`DOC_logon`, `BES_logon`, `CSM_logon`, etc.) are only valid in the process which called the logon routine. The handle can't be passed to another process and used there.

The handles returned by a "logon" done by a parent process may not be used by a child process. The software does not have the capability to support concurrent use of a handle, so you may not logon to a

service, fork the process, and then use the logon handle in the child process.

General Information

The ISTK libraries are thread safe on Windows, but not on UNIX. The use of multiple threads in a single process in conjunction with ISTK library calls on UNIX systems can cause hangs of that process, and other peculiar behavior.

Many entry points return a pointer to a pointer to a piece of memory. Whenever you see this (a "***" after the type), check to see how the memory is managed. In some cases (for example, `INX_get_all_dcl_descs`), the memory is read only and the client should not modify or return it. In other cases (for example, `INX_find_folders`), the memory is allocated by ISTK in the local heap, and must be returned by the application calling the indicated deallocate routine (`INX_free_folders`).

Any entry point which does network activity can get an error due to the network being down, or the target station being down. These errors are recognized by the error category being equal to "err_SPP" (the check is `err_category(error) == err_SPP`), and some applications will want to retry these errors to provide fault tolerance.

When compiling applications which use ISTK, the symbol "`_ANSI_C_SOURCE`" may be `#define`d in the program prior to the include of the ISTK header files to make the compiler check types on each ISTK entry point. The definition of this symbol may also be omitted if the compiler being used is not capable of handling ANSI C procedure prototype definitions.

4. CONCEPTS

This section describes the major concepts which must be understood when using this manual. This section should be read completely before proceeding to subsequent sections.

system serial number

A number which identifies a FileNet imaging system (abbreviated SSN). Each system has one SSN assigned to it. No two systems have the same SSN.

page

A page is a stream of bytes which was originally used to hold the data corresponding to a single sheet of paper. A page has since been expanded to hold many types of data, however, and the only restrictions are related to the page size due to performance reasons. Note that a page shouldn't be too small (just a few bytes), because the overhead required to store a page would make the ratio of overhead to user data too high. Additionally, an entire page must be retrieved from optical disk at the same time, so making the page too large can cause performance problems if a program typically needs only a very small portion of a page. Currently the largest pages used are 9 megabytes, the smallest are around 100 bytes, and the average page is about 80 kilobytes.

document

A document is a collection of related pages, and also some summary information which consists primarily of "indexes" (see next item). A document is identified by a document id, which is a number assigned to the document when the document is created, and also a system serial number.

index

An index is a value associated with a document used to identify the contents of the document. An index has a name which identifies the type or category of the index (such as "account_name"), and a value which is an attribute of a document (such as "Fred Smith"). A document can have one or more indexes associated with it.

document class

A document class is a category of documents. Each document class may have a different set of indexes and other attributes associated with it. For example, a bank may want one document class for loans, and another for savings deposits.

optical disk surface

An optical disk surface, or just "surface", is a single side of an optical disk. A surface is identified by a surface id, which is a number assigned to the surface when the surface is first used, and also a system serial number. No two surfaces have the same system serial number and surface id.

optical disk family

An optical disk family is a sequence of optical disks. An optical disk is a member of only one family. A family keeps track of which

optical disks for that family are currently being written. When one of the disks being written fills up, another is automatically assigned to take its place. A document class will write to a specified optical disk family, so different classes can be set up to write to different optical disk families, and thus keep documents of different types on different sets of disks. An optical disk family is only used when documents are being written to optical disk. Families are not referenced or used when documents are read from optical disk.

folder

A folder is a container used to hold a list of documents. A folder is given a name by the user when created. Putting a document into a folder involves only saving the document id in the folder's list of documents.

annotation

An annotation is a small note associated with a location on a page. Think of an annotation as a comment written on a sheet of paper with an arrow pointing to the item the comment refers to.

capability

A capability is a structure which allows a user exclusive access to an object such as a document, folder, annotation, etc. When a routine is called to lock or open an object, it returns the capability structure to the application program. Then the application can call another routine or set of routines to modify and unlock/close the object, and this saves the new version and releases the lock on the object, which also makes the capability no longer valid for additional updates. Examples of capability structures are the `INX_capability_typ`, `ASE_capability_typ`, and `BES_batch_cap_typ` structures described later in this document.

magnetic disk cache

Several magnetic disk caches exist which can be used to store pages of documents. These caches are managed by the Cache Service Manager (CSM) described later in the document.

index database

The index database contains the folders and indexes of documents the user has defined, and also has the definition of the document classes and index types, maximum index lengths, etc. The index database is managed by Index Services (INX) described later in the document.

There is only one index database on a system.

doc locator database

The document locator database contains the locations of the documents on optical disk, and also the definitions of the optical disk families, surfaces, and annotations. The document locator database is also referred to as the MKF permanent database or just permanent database. The doc locator database is managed by the Document Service (DOC) described later in this document. There is only one doc locator database on a system.

transient database

The transient database is a typically small database which holds items for relative short periods of time. The largest component of the transient database are the records which define the locations of pages in the magnetic disk cache. The transient database is used by several entities, including the Document Service and the Cache Service. There can be one or more transient databases on a system.

OSAR

OSAR stands for "optical storage and retrieval". It refers to a hardware unit which contains one or more optical drives, a storage rack to hold optical disks, and a robotic arm which can move disks between slots in the storage rack and the optical disk drives.

retrieval

Retrieval is the act of copying a document from wherever it's located to the workstation which is requesting it. Retrieval will typically cause a document to get copied from optical disk to a magnetic disk cache, and then from the cache to the local workstation. Retrievals can be optimized so that the document is already in cache when the retrieval is made, and thus the first step which is the time consuming one is omitted.

committal

Committal is the act of creating a document and making it available from the Document Service. It requires putting the pages in the appropriate cache and inserting a record into the doc locator database which defines the document. It also may or may not post a request to write the document to optical disk. Note that committal is considered complete when the cache and database updates are done, and the write to optical disk, if requested, occurs afterwards.

cataloging

Cataloging a document is the act of putting a record into the index database which defines the document, which makes this document available from the Index Service.

Note that for each document, one record is put into the index database (via cataloging) to indicate what the document is, and another record is put into the doc locator database (via committal) to indicate where (which optical surface and the offset on that surface) the document is.

clustering

Clustering is a method of writing documents to optical disk so that documents in a group get written to the same optical disk, even though the group is not all written in the same period of time. The advantage of clustering is that if the entire group of documents must be retrieved, then only one optical disk needs to be mounted in a drive, and retrieval performance is better. For example, if a hospital wants to retrieve all documents for one patient at the same time, then documents could be clustered on the patient name, so that all documents with the same patient name are written to the same optical disk.

In the above example, the patient name is called the "cluster index", because it is the index which determines which documents are written on the same optical disk, and a group of documents for a single patient is called a "cluster". To invoke clustering, an index must be designated as the cluster index, and then a document class must be created which has clustering enabled on the cluster index.

When clustering is used, the system will set aside space on optical disk in anticipation of new documents being added to existing clusters, and will start writing on new blank disks before all the space on an older disk is written. But when clustering is not used, a single optical disk will be written from start to finish before the next blank optical disk is used. Note that this difference implies that use of clustering is a space/performance tradeoff, and the use of clustering must be analyzed carefully before it is enabled.

cluster space

A cluster space is an indication of which application is using clustering. For example, a hospital may have two applications, one for patients, and another for employees. If the hospital wanted to have patient records clustered on patient name, and also wanted to have employee records clustered on employee name, but didn't want

employee records to be on the same disks as the patient records, it would create two different cluster spaces, one for each application.

Only one cluster space is supported at the current time, but the cluster space parameter is defined in the various interfaces so that multiple cluster spaces can be supported in the future.

connection

A network connection (or just connection) is a link between one process on one machine and another process on another machine. The procedure calls listed in this manual utilize connections to link application programs on a user's workstation with server processes on a central server.

Each module which makes procedure calls across the network has an entry point to open a connection, and another to close a connection. An application may use these entry points to open and close connections explicitly, or may omit use of these routines and then connections will be opened and closed automatically on every procedure call.

The advantage of an application calling open and close connection explicitly is that it can issue multiple procedure calls between the open and close of the connection, which results in better performance. An application which uses explicit connections, however, should not keep them open for long periods of time or when they are not being used, because opening a connection dedicates a server process to the application program, and the number of server processes is limited and may run out.

5. MODULE SUMMARY

ISTK is divided into a number of modules, each of which is a collection of callable entry points for a specific functional area. These modules are as follows:

BES - Batch Entry Services

Provides a set of entry points and data structures to support a high volume, batch oriented, document acquisition application. Batch Entry Services uses the concept of a batch, into which images are scanned and then assembled into documents, pages are verified for being correctly input, indexes are added, and then the document is committed with the Document Service and cataloged with the Index Service.

CKS - Checksum Services

Provides an entry point and data structures to perform checksumming on specified data.

CSM - Cache Server Manager

Provides a set of entry points and data structures for high speed magnetic disk based storage and retrieval of document pages. Cache Services is used by the other services as a means of storing document pages, and provides for optional least-recently-used deletion of pages.

DOC - Document Services

Provides a set of entry points and data structures for the storage, retrieval, and management of image and text documents on a virtual hierarchical mass storage system. Document Services has facilities for committing documents with or without migration to optical disk, migration of documents from optical disk to cache, annotating document pages, and miscellaneous reporting and maintenance functions.

DTI - Date Time Interface

Provides a set of entry points and data structures for converting dates and times from one format to another (string to number, etc.).

FFI - FileNet Formats Interface

Provides a set of entry points and data structures to process objects which are documented as streams of bytes in other modules into usable formats. Includes routines to compress/decompress image/text pages.

FP - Floating Point Numbers

Provides a set of entry points and data structures for converting to and from FileNet format floating point numbers and the client stations native format or printable string format.

IMS - Image Management Services

Provides a set of entry points and data structures to facilitate access to ISTK. It contains handle management routines to reduce the number of logon calls required, plus miscellaneous functions.

INX - Index Services

Provides a set of entry points and data structures for the creation, deletion, and modification of document and folder information contained in the index database. It also contains routines to create and modify index and document class information.

PRI - Print Services

Provides a set of entry points and data structures for the printing and outbound faxing of documents and text objects. Print Services is actually a spooling facility which can control any number of physical print and fax server stations elsewhere on the network.

PRS - Parse Services

Provides a set of entry points to allow for string queries to be passed to Index Services.

SEC - Security Services

Provides a set of entry points and data structures for acquiring system wide and object specific security attributes. Note that object (e.g. document, page, queue, etc.) security is also enforced by the other services.

SQL - SQL Services

Provides a set of entry points for accessing various relational databases using SQL (Structured Query Language) as the interface language.

WQS - WorkFlo Queue Services

Provides a set of entry points and data structures for creating, maintaining, and using WorkFlo queues, which are used to customize applications for the routing of documents between users.

6. BES DECLARATIONS

```

/*****
*
*
*
*****/

/*
*
*   Batch Entry Services provides its clients with batch-oriented
*   document entry functions.
*
*   These functions provide the client with the ability to:
*
*       o Create a batch.
*       o Add images to a batch.
*       o Assemble images into documents.
*       o Index documents.
*       o Validate certain conditions on batches, images,
*         and documents.
*       o Enqueue a batch for committal.
*       o Retrieve batch contents.
*       o Delete a batch.
*/

#ifndef BES_defs
#define BES_defs

#include <AS_externals.h>
#include <SEC.defs>
#include <FP.defs>
#include <INX.defs>

/*****
*
*   C data definitions for Batch Services (BES).
*
*****/

/* Batch size restrictions */

#define BES_MAX_BATCHDOCS      5000      /* Max documents per batch */
#define BES_MAX_BATCHPAGES    5000      /* Max pages per batch (document
                                         header not counted as a page). */

/* The following declarations identify types of batches according
```

+++ BES - Batch Entry Services +++

to who created them. */

```
typedef char BES_batch_name_typ;
#define BES_BATCH 0 /* regular batch */
#define BES_QUICK 1 /* quick batch */
#define BES_FORM 2 /* fill-in document */
#define BES_WPS 3 /* WP document */
#define BES_TEST 4 /* test batch */
#define BES_DEW 5 /* WorkFlo */
#define BES_ARCH 6 /* ? */
#define BES_PC 7 /* PC */
#define BES_FAX 8 /* FAX */
#define BES_DDT 9 /* DDTs */
#define BES_FOLDER 10 /* folder */
#define BES_DEL 11 /* Deletion - for system use only. */
```

```
#define BES_MAX_NAME_TYPE BES_FOLDER
```

```
/* The following declarations define the characters batch names start
with for service generated batch names. */
```

```
#define BES_NAME_TYPE_BATCH 'b' /* batch prefix */
#define BES_NAME_TYPE_QUICK 'q' /* quick batch prefix */
#define BES_NAME_TYPE_FORM 'f' /* form prefix */
#define BES_NAME_TYPE_WPS 'w' /* WP prefix */
#define BES_NAME_TYPE_TEST 't' /* test batch prefix */
#define BES_NAME_TYPE_DEW 'x' /* WFL prefix */
#define BES_NAME_TYPE_ARCH 'a' /* ? */
#define BES_NAME_TYPE_PC 'p' /* PC */
#define BES_NAME_TYPE_FAX 'X' /* FAX prefix */
#define BES_NAME_TYPE_DDT 'd' /* DDTs tape transfer prefix */
#define BES_NAME_TYPE_FOLDER 'F' /* Folder */
#define BES_NAME_TYPE_DEL 'D' /* Deletion - for system use only. */
```

```
/* Batch status indicators. */
```

```
typedef short BES_open_status_typ;
```

```
#define BES_AVAILABLE 0
#define BES_BUSY 1
```

```
/* Number of phases */
```

```
#define BES_NUM_PHASES 11
```

```
/* Batch phase identifiers */
```

```
typedef short BES_phase_num_typ;
```

```
#define BES_DEFINE 0 /* batch creation */
#define BES_SCAN 1 /* scan */
```

+++ BES - Batch Entry Services +++

```

#define BES_IMAGE_VER          2      /* image verification */
#define BES_RESCAN             3      /* rescan */
#define BES_ASSEMBLY          4      /* manual assembly */
#define BES_INDEX              5      /* index */
#define BES_INDEX_VER         6      /* index verification */
#define BES_BATCH_TOTAL       7      /* batch total */
#define BES_COMMIT            8      /* committal */
#define BES_CATALOG           9      /* catalog */
#define BES_RECOMMIT         10     /* recommittal */

/*          Phase status identifiers.          */

typedef short          BES_phase_status_typ;
#define BES_NOT_STARTED    0      /* uninitiated */
#define BES_IN_PROGRESS    1      /* running */
#define BES_INTERRUPTED    2      /* stopped before done */
#define BES_DONE           3      /* completed */
#define BES_NOT_NECESSARY  4      /* not required */
#define BES_HAS_ERRORS     5      /* error(s) occurred */
#define BES_OPTIONAL       6      /* phase is optional */

/* Committal information. Note that both queue typedefs have the same
   values stored in them. */

typedef short          BES_commit_queue_typ;
#define BES_OCR_QUEUE      4      /* batch in OCR queue */
#define BES_INPROGRESS_QUEUE 3      /* worked on by daemon */
#define BES_COMMIT_QUEUE   2      /* committal queue */
#define BES_UNCOMMIT_QUEUE 1      /* errors from committal */
#define BES_NO_QUEUE       0      /* not in any queue yet */

typedef long           BES_execute_com_typ;
#define BES_ASYNCHRONOUS    0      /* enqueue batch */
#define BES_SYNCHRONOUS     1      /* synchronous commit */
#define BES_FAST_ASYNCHRONOUS 2      /* fast asynchronous commit */
#define BES_SYNCHRONOUS_NO_WAIT 3      /* synchronous, no wait */

typedef short          BES_migrate_com_typ;
#define BES_NORMAL_COMMIT   1      /* migrate to disk */
#define BES_NO_MIGRATE      2      /* keep on mag. disk */

/*          Filter information.          */

typedef long           BES_relop_typ;
#define BES_EQUAL           0      /* equal */
#define BES_GEQ             1      /* greater or equal */
#define BES_GT              2      /* greater than */

```

+++ BES - Batch Entry Services +++

```
/*          Image source          */

typedef short          BES_source_typ;
#define BES_SOURCE_UNKNOWN    INX_SRC_UNKNOWN
#define BES_MANUAL            INX_SRC_MANUAL
#define BES_OCR                INX_SRC_OCR
#define BES_APERTURE          INX_SRC_APERTURE

/*          Image types. (obsolete--use FN_page_header.h defines)  */

/*
#define BES_IMAGE              FN_IMAGE_PAGE_TYPE
#define BES_TEXT                FN_TEXT_PAGE_TYPE
#define BES_FILLIN_PAGE        FN_FORM_PAGE_TYPE
#define BES_COMPOSITE           FN_COMPOSITE_PAGE_TYPE
#define BES_SEP_SHEET           FN_SEP_SHEET
#define BES_OTHER               FN_OTHER_PAGE_TYPE
*/

/*          Image verification states.          */

typedef short          BES_verify_stat_typ;
#define BES_UNSEEN              0
#define BES_GOOD                1
#define BES_BAD                 2
#define BES_REQUIRED            3
#define BES_DELETE              4

/*          Index status states          */

typedef short          BES_index_stat_typ;
#define BES_SUCCESS              0
#define BES_HAS_ERROR            1
#define BES_UNKNOWN              2

/*          Misc defines          */

#define BES_FILTER_ALL           -1
#define BES_MAX_PAGES_PER_DOC    1000
#define BES_MAX_INDICES_PER_DOC  224
#define BES_FIRST_BATCH_ID       1
#define BES_MAX_INDEX_LEN        239
#define BES_MAX_TEXT_DATA_LEN    239

/* Options for opening a batch */

typedef unsigned short  BES_open_options_typ;
#define BES_OPEN_NORMAL          0
```


+++ BES - Batch Entry Services +++

```

#define BES_OVERRIDE                1
#define BES_OPEN_READONLY          0x1357

/*          Sort definitions          */

typedef short                       BES_sort_order_typ;
#define BES_SEQUENTIAL              0
#define BES_REVERSE                  1

/*****
 *
 *          M I S C   T Y P E D E F S
 *
 *****/

typedef char                        *BES_handle_typ;
typedef unsigned long               BES_batch_id_typ;

/* Filter declaration for finding batches.  Each item in the filter can
   have a value in the group of values identified by the type, or
   can have the value "BES_FILTER_ALL" to mean don't use a particular item
   as a filter, and match any value. */

typedef struct {
    BES_open_status_typ  open_flag;      /* only return batches with this          (pg24.37)
                                         value for the open flag. */
    BES_commit_queue_typ queue;         /* only return batches in this          (pg25.25)
                                         queue. */
    BES_phase_num_typ   this_phase;     /* only return batches with this as     (pg24.46)
                                         the current phase. */
    BES_phase_num_typ   next_phase;     /* only return batches with this as     (pg24.46)
                                         the next phase. */
    BES_phase_num_typ   which_status;   /* which phase the status belows       (pg24.46)
                                         is for */
    BES_phase_status_typ phase_status;  /* only return batches with this       (pg25.13)
                                         status for phase which_status */
} BES_filter_typ;

/* The batch capabilities structure allows a user exclusive access to the
   batch.  This structure is returned from a BES_create_batch or BES_open_batch
   call, and then passed in to subsequent calls to acquire access to the
   batch. */

typedef struct {
    BES_batch_id_typ      batch_id;      (pg27.19)
    ASE_time_typ         open_time;     (pg489.31)
}

```

+++ BES - Batch Entry Services +++

```

    SEC_id_typ          open_user_id;          (pg373.32)
    ASE_ssn_typ         open_ssn;             (pg483.47)
} BES_batch_cap_typ;

```

/* The BES_stat_hdr_desc_typ record below contains items which are defined for a batch which the user cannot update directly. Note that many items are described as "from doc class", which means that the value is taken from the document class description for the document class this batch is assigned to. */

```

typedef struct {
    BES_open_status_typ  open_flag;          /* BES_AVAILABLE, BES_BUSY */      (pg24.37)
    bool                delete_flag;        (pg491.22)
#ifdef FDOS
    short               unused2;           /* unused, contents unspecified */
#endif
    BES_commit_queue_typ  queue;           /* BES_COMMIT_QUEUE, ...*/        (pg25.25)
    ASE_time_typ         qtime;           /* time inserted into queue */     (pg489.31)
    ASE_time_typ         create_time;      /* time batch created */           (pg489.31)
    SEC_id_typ           create_user_id;   /* id of user who created batch */ (pg373.32)
    bool                 tab_out_flag;     /* value of tab_out_flag from     (pg491.22)
                                     document class of batch */
    INX_dcl_name_typ     doc_class_name;   /* all docs in this batch are for (pg239.47)
                                     this document class */
    short                num_indices;      /* #indices from doc class */
    short                num_reqd_indcs;   /* #required indices in doc class */
    short                num_menus;       /* #menus in doc class */
    char                 indexing_form[ASE_MAX_ANYNAMELEN + 1]; /* Form name
                                     for this document class */
    SEC_access_restrictions access_restrct; /* access restrictions for batch */ (pg385.2)
    INX_cluster_space_typ cluster_space;   /* cluster space number */         (pg243.31)
    INX_index_id_typ     cluster_index;    /* index id of cluster index or   (pg235.34)
                                     INX_INVALID_INDEX_ID if not
                                     clustering */
    unsigned long        fam_id;          /* Family to write docs to. */
    char                 wfl_queue_name[ASE_MAX_WFQUEUEUENAME + 1];
                                     /* WorkFlo queue from doc class */
    char                 wfl_sys_name[ASE_MAX_WFSYSNAME + 1];
                                     /* WorkFlo system from doc class */
    short                retent_disp;     /* retent_disp from doc class */
    short                retent_base;     /* retent_base from doc class */
    short                retent_offset;   /* retent_offset from doc class */
    long                 act_pages;       /* number of pages in batch */
    long                 act_docs;        /* number of documents in batch */
    long                 next_doc_id;     /* Starts at 1 for first document
                                     and increments by one for each
                                     document created. Note that
                                     a "doc_id" in batch services is

```

+++ BES - Batch Entry Services +++

a batch relative number, and
will not work if used as an
input to a DOC or INX call. */

} BES_stat_hdr_desc_typ;

/* The BES_dyn_hdr_desc_typ structure contains the items which are defined
for a batch which a user can directly set by calling BES_update_batch.
These entail options on how the batch will be processed and status of the
completion of the batch. Note that many of the items in this structure
are for recording information only, and are not checked or used by
Batch Services. Such items are documented as "not checked by Batch
Services". */

```
typedef struct {
    char          batch_name[ASE_MAX_BIG_O_LEN+1]; /* Name of batch */
    long          exp_pages; /* Expected number of pages in batch.
                             Set to zero upon batch creation.
                             Not checked by Batch Services. */
    long          exp_docs; /* Expected number of docs in batch.
                             Set to zero upon batch creation.
                             May be changed by caller. Not
                             checked by Batch Services. */
    unsigned short pages_per_doc; /* Pages per document. Initialized to
                                   the value from the document class.
                                   May be changed by caller. Not
                                   checked by Batch Services. */
    bool          double_sided; /* If pages of batch are double sided. (pg491.22)
                                Initialized to FALSE. Not
                                checked by Batch Services. */
    bool          verify_images; /* If an image verification should be (pg491.22)
                                done on each image. Initialized
                                to value from document class. If
                                equal to TRUE, then BES_verify_image
                                must be called on each image prior
                                to committal, or else a committal
                                attempt will be rejected. */
    bool          verify_indices; /* If an index verification should be (pg491.22)
                                done on each index. Initialized to
                                value of "verify_indexes" from
                                document class. If equal to TRUE
                                and the "verify_flag" in the index
                                description in INX is also TRUE,
                                then batch services will require
                                that "phase_status[BES_INDEX_VER]"
                                of the BES_doc_desc_typ record for
                                each document be equal to BES_DONE
                                prior to committal, or else a
                                committal attempt will be rejected.
```

+++ BES - Batch Entry Services +++

Note that a copy of "verify_flag" is also in the BES_ixdir_desc_typ record. */

BES_migrate_com_typ committal_type; /* Controls migrate to od. Set to (pg25.38)
 BES_NORMAL_COMMIT upon batch creation. */

bool batch_total; /* If batch totals are to be checked. (pg491.22)
 Set to value from document class upon batch creation. Having this flag and "batch_tot_flag" in the BES_ixdir_desc_typ record equal to TRUE causes the exp_total and act_total values in the BES_ixdir_desc_typ record to be compared when a commit is attempted, and the commit to be rejected with an error if they aren't equal. */

BES_phase_num_typ this_phase; /* Current phase. Set to BES_DEFINE (pg24.46)
 upon batch creation. Not checked by Batch Services. */

BES_phase_num_typ next_phase; /* Current phase. Set to BES_SCAN (pg24.46)
 upon batch creation, and set to BES_RECOMMIT if a commit attempt fails. Not checked by Batch Services. */

BES_phase_status_typ phase_status[BES_NUM_PHASES]; /* Phase statuses. (pg25.13)
 Set to BES_NOT_STARTED upon batch creation. If a committal fails, phase_status[BES_COMMIT] is set to BES_HAS_ERRORS. Each phase status must be set to BES_DONE or BES_NOT_NECESSARY prior to committal or else the committal will fail. */

ASE_time_typ phase_lst_time[BES_NUM_PHASES]; /* Last time phase (pg489.31)
 worked on. Set to 0 upon batch creation. Not checked by Batch Services. */

SEC_id_typ phase_user_id[BES_NUM_PHASES]; /* User id who (pg373.32)
 performed phase. Set to SEC_NULL_GROUP_ID upon batch creation. Not checked by Batch Services, except that it can't be set to SEC_UNDEFINED_ID. */

BES_sort_order_typ sort_order; /* Order pages sorted in. Set to (pg27.6)
 BES_SEQUENTIAL upon batch creation. Not checked by batch services. */

+++ BES - Batch Entry Services +++

```

bool          ocr_enable;      /* Whether to do optical character      (pg491.22)
                                recognition on pages in batch.  Set
                                to FALSE upon batch creation.  Not
                                checked by Batch Services (OCR
                                capabilities expected to be in
                                client software--not in BES). */

bool          auto_index;     /* Whether to do automatic indexing.   (pg491.22)
                                Set to FALSE upon batch creation.
                                Not checked by Batch Services (auto
                                indexing capabilities expected to
                                be in client software--not in BES)*/

long          migrate_delay;  /* The committed document pages will
                                not be migrated to optical disk
                                until after migrate_delay number
                                of seconds had elapsed after the
                                batch was committed.  This field is
                                only meaningful if committal_type
                                is BES_NORMAL_COMMIT.
                                Note also that if this field
                                is non-zero, FAST_ASYNCHRONOUS
                                committals will automatically be
                                converted to regular ASYNCHRONOUS
                                committals.
                                When a batch is created, this
                                field will be set to the value
                                as specified in the document class,
                                and can be updated later. */

ASE_service_name_typ migr_dest_cache; /* If non-null, pages of the committed (pg486.28)
                                documents will be migrated to the
                                cache specified in this field.
                                This field is initialized to NULL
                                when the batch is created, and can
                                be updated. */

long          reserved_1;     /* reserved--set to zero */
long          reserved_2;     /* reserved--set to zero */
long          reserved_3;     /* reserved--set to zero */
long          reserved_4;     /* reserved--set to zero */
} BES_dyn_hdr_desc_typ;

typedef struct {
    BES_dyn_hdr_desc_typ  d; /* Part of header user can update      */ (pg31.38)
    BES_stat_hdr_desc_typ s; /* Part of header user cannot update */ (pg29.4)
} BES_hdr_desc_typ;

/* The BES_ixdir_desc_typ record contains information about each index for
the document class.  All elements of this record except "act_total" are
copies of the information from index services at the time the batch was
created. */

```

+++ BES - Batch Entry Services +++

```
typedef struct {
    INX_index_id_typ      index_id;      /* Index number */                (pg235.34)
    char                  index_name[ASE_MAX_BIG_O_LEN+1]; /* index name */
    INX_value_type_typ    index_type;     /* index type */                (pg235.48)
    unsigned short        max_string_len; /* max length of index if string */
    char                  menu_name[ASE_MAX_ANYNAMELEN+1]; /* applicable
                                                                    menu name for index, if any */
    bool                  verify_flag;     /* if want this index to be
                                                                    verified. See "verify_indices"
                                                                    flag in BES_dyn_hdr_desc_typ. */    (pg491.22)
    bool                  required_flag;   /* if this index is required. */    (pg491.22)
    bool                  batch_tot_flag;  /* TRUE=>add up all values for this
                                                                    index for each document in the
                                                                    batch, and compare it against
                                                                    "exp_total" when batch is
                                                                    committed. See "batch_total"
                                                                    in BES_dyn_hdr_desc_typ. */
    bool                  up_case_flag;   /* If index is upper case */        (pg491.22)
    BES_source_typ        source;         /* BES_MANUAL, BES_OCR,... */      (pg26.3)
    FP_number             exp_total;      /* expected total for index */      (pg211.20)
    FP_number             act_total;      /* actual total for index */        (pg211.20)
    char                  mask[ASE_MAX_ANYDESCLEN+1]; /* mask for index */
} BES_idxdir_desc_typ;
```

/* The BES_image_desc_typ record describes an image in a batch */

```
typedef struct {
    ASE_doc_id_typ        image_id;       /* Values are 100,000 and up */    (pg483.37)
    unsigned long         image_length;   /* #bytes in image */
    short                 image_type;     /* Image type. Not checked by
                                                                    batch services. Set to any
                                                                    value desired. */
    BES_verify_stat_typ  image_ver_stat; /* image verification status */    (pg26.22)
    ASE_doc_id_typ        doc_id;        /* 0 if not in a doc., or 1..n for
                                                                    1st through nth document in
                                                                    batch. */                          (pg483.37)
    ASE_page_number_typ  page_id;        /* page number if in document */    (pg483.41)
    bool                  end_of_doc;     /* not used--meaningless */        (pg491.22)
    unsigned short        index_len;     /* 0 if no index */
} BES_image_desc_typ;
```

```
typedef struct {
    char                  index_value[BES_MAX_INDEX_LEN + 1];
} BES_image_ix_desc_typ;
```

/* New index_type for implementing packed index values */
#define BES_INDEX_TYPE_PACKED 1

+++ BES - Batch Entry Services +++

```
#define BES_INDEX_TYPE_NULL 2
```

```
/* Packed indexes, which are stored in the index_value field of the
batch_ixval table, will have the following structure:
```

```
    index_id      short
    index_type    short
    data_status   short
    index_len     short
    index_value   [index_len bytes of data]
```

```
*/
```

```
/* The BES_ixval_desc_typ record holds an index associated with a document. */
```

```
typedef struct {
    INX_index_id_typ      index_id;          /* index id */                (pg235.34)
    INX_value_type_typ    index_type;        /* type of index, must be either (pg235.48)
                                         INX_VT_FPNUM, INX_VT_ASCII,
                                         INX_VT_DATE, INX_VT_MENU. No
                                         other values are used. */
```

```
/* The index_value has the following contents based upon index_type:
```

```
    INX_VT_ASCII - NULL terminated string (no 2 byte length field)
    INX_VT_MENU  - index_value[0] is the numeric value of the menu item,
                  and index_value[1] is a NULL.
    INX_VT_DATE  - ASE_date_typ
    INX_VT_FPNUM - FP_number
```

```
Note that "index_value" will need to be typecast for assignments
to fields of types INX_VT_DATE and INX_VT_FPNUM. */
```

```
char          index_value[BES_MAX_INDEX_LEN + 1];
```

```
BES_index_stat_typ    data_status;        /* Used by OCR. Not checked or (pg26.31)
                                         used by batch services. */
```

```
} BES_ixval_desc_typ;
```

```
typedef struct {
    INX_index_id_typ      index_id;          (pg235.34)
    INX_value_type_typ    index_type;        (pg235.48)
    unsigned short        index_len;
    char                  index_value[BES_MAX_INDEX_LEN + 1];
    BES_index_stat_typ    data_status;        (pg26.31)
} BES_mkf_ixval_desc_typ; /* Used in BES_ixval_cursor_rec_typ */
```

```
typedef struct {
    ASE_doc_id_typ      doc_id;          /* Batch relative document id. Values (pg483.37)
```

+++ BES - Batch Entry Services +++

```

                                range from 1 for first document in batch
                                to "n" for nth document in batch. Note
                                that is not the document id which will
                                be used to access the document in either
                                the index or doclocator databases. */
unsigned char      unused;      /* unused, contents unspecified */
INX_doc_type_typ  doc_type;     /* document type */                                (pg234.11)
long              num_pages;    /* #pages in document */
short            num_indices;   /* #indices in document */
short            num_reqd_indcs; /* #required indices (not used!) */
BES_phase_num_typ phase_num;    /* phase of this document */                                (pg24.46)
error_typ        phase_error;  /* error tuple if error on this doc */          (pg493.26)
BES_phase_status_typ phase_status[BES_NUM_PHASES]; /* phase statuses */      (pg25.13)
INX_fam_id_typ   fam_id;       /* family id of document (not used!) */      (pg243.25)
INX_cluster_key_typ cluster_key; /* cluster key for this document. If
                                clustering is used, this value must
                                be computed by calling either
                                INX_num_cluster_id or
                                INX_str_cluster_id depending on the
                                type of index used for clustering.
                                If clustering is not used, then all
                                elements of this structure must be
                                set to zero. */
long              ocr_schema;  /* Values are client defined. This field
                                is used to store information about the
                                document while in the batch. This
                                field is not checked or used by
                                Batch Services and is for client use
                                only. */

} BES_doc_desc_typ;

typedef struct {
    unsigned long  batch_id;
    unsigned long  batch_name_id;
    unsigned long  quick_name_id;          /* not used */
    unsigned long  form_name_id;          /* not used */
    unsigned long  wps_name_id;           /* not used */
    unsigned long  test_name_id;          /* not used */
    unsigned long  dew_name_id;           /* not used */
    unsigned long  batch_open_id;
    unsigned long  folder_id;
} BES_ctl_desc_typ;

typedef struct BES_doc_list_typ {
    BES_doc_desc_typ  d;          /* document description record */          (pg34.30)
    ASE_doc_id_typ*   page_ary;   /* ptr to array of image ids or 0 */        (pg483.37)
    BES_ixval_desc_typ* ixval_ary; /* ptr to array of index values or 0*/      (pg33.37)
    struct BES_doc_list_typ* next_doc; /* ptr to next BES_doc_list_typ or 0*/      (pg35.1)
}

```


+++ BES - Batch Entry Services +++

```
} BES_doc_list_typ;

/*****
 *
 * get_info/put_info related typedefs
 *
 *****/

#define BES_INVALID_BATCH_ID      0

/* put_info/get_info type */

#define BES_INVALID_INFO 0
#define BES_INFO_BATCH  1
#define BES_INFO_DOC    2
#define BES_INFO_IMAGE  3
#define BES_INFO_IDXVAL 4

typedef unsigned short  BES_info_typ;

typedef struct
{
  ASE_doc_id_typ      doc_id;      /* Values are 1 to n for 1st to nth doc */ (pg483.37)
  unsigned short      index_id;
} BES_ixval_spec_typ;

typedef struct
{
  BES_info_typ        spec_type;      (pg35.19)
  union
  {
    ASE_doc_id_typ    doc_id;      /* Values are 1 to n for 1st to nth doc */ (pg483.37)
    ASE_doc_id_typ    image_id;   /* Values are 100,000 and up */ (pg483.37)
    BES_ixval_spec_typ ixval_spec; (pg35.25)
  } spec;
} BES_info_spec_typ;

typedef struct BES_info_rec_typ
{
  BES_info_spec_typ      info_spec;      (pg35.36)
  unsigned short         info_len;
  char                   *info_text;
  struct BES_info_rec_typ *next_rec_p;  (pg35.44)
} BES_info_rec_typ;

/*****
 *
 * folder related typedefs

```

+++ BES - Batch Entry Services +++

```

*
*****/

#define BES_ROOT_FOLDER_ID 0
#define BES_FIRST_FOLDER_ID 1
#define BES_INVALID_FOLDER_ID 0xFFFFFFFF

#define BES_MAX_NODE_NAME_LEN 60
#define BES_MAX_FOLDER_INFO_LEN 600
typedef unsigned long BES_node_id_typ;
typedef char BES_folder_node_name_typ [BES_MAX_NODE_NAME_LEN];

typedef struct {
    BES_folder_node_name_typ base_node_name; (pg36.11)
    BES_node_id_typ node_id; (pg36.10)
    BES_batch_id_typ batch_id; /* If batch_id = 0, then this (pg27.19)
    * node is not a batch
    */
    BES_node_id_typ parent_id; (pg36.10)
    unsigned long last_update; /* creation/update time
    * in 32 bit in seconds
    */
    unsigned short info_len;
    byte info_data[BES_MAX_FOLDER_INFO_LEN]; (pg491.23)
} BES_folder_desc_typ;

/*****
*
* Expanded info repository related typedefs
*
*****/

#define BES_INVALID_OBJECT_TYPE 0
#define BES_INVALID_SEQUENCE_ID 0

#define BES_OBJECT_TYPE_ALL 0xFFFFFFFFE
#define BES_OBJECT_ID_ALL 0xFFFFFFFFE
#define BES_SEQUENCE_ID_ALL 0xFFFFFFFFE

#define BES_OBJECT_TYPE_BATCH 1
#define BES_OBJECT_TYPE_DOC 2
#define BES_OBJECT_TYPE_IMAGE 3
#define BES_OBJECT_TYPE_PHASE 4

#define BES_MAX_OBJECT_DATA_LEN 800

typedef unsigned long BES_object_type_typ;

```

+++ BES - Batch Entry Services +++

```

typedef unsigned long   BES_object_id_typ;
typedef unsigned long   BES_sequence_id_typ;

typedef struct
{
    BES_object_type_typ      spec_type;           (pg36.48)
    BES_object_id_typ        object_id;          (pg37.1)
    BES_sequence_id_typ      sequence_id;        /* Must be greater than 0 */ (pg37.2)
} BES_object_spec_typ;

typedef struct BES_object_rec_typ
{
    BES_object_spec_typ      object_spec;        (pg37.9)
    unsigned short           data_len;
    byte                     *data;            (pg491.23)
    struct BES_object_rec_typ *next_rec_p;      (pg37.17)
} BES_object_rec_typ;

/*****
 *
 * Image Companion Cache Data related typedefs
 *
 *****/

#define   BES_MIN_COMPANION_ID   1
#define   BES_MAX_COMPANION_ID   20

typedef struct
{
    ASE_image_id_typ         image_id;          /* Image Id. */ (pg483.33)
    ASE_page_number_typ      companion_id;     /* Companion Id. */ (pg483.41)
    unsigned long            data_length;      /* Length of companion data
                                                * in bytes
                                                */
} BES_companion_desc_typ;

/*****
 *
 * Level for Extended Indexing.  This value can be set
 * through the BES_extended entry point and is associated with
 * BES session.  The level is set to BES_level_0 by default.
 * BES_level_0: No extended indexes
 * BES_level_1: New extended indexes (F_DOCLOCATION and F_DOCFORMAT
 * support RCI 1448, 1450)
 *
 *****/

#define   BES_level_0   0

```

+++ BES - Batch Entry Services +++

```
#define BES_level_1 1

#define BES_MIN_LEVEL BES_level_0
#define BES_MAX_LEVEL BES_level_1

/*****
 *
 *          E R R O R   D E F I N E S
 *
 *****/

#define BES_generic 0
#define BES_xmain 1
#define BES_ZOTHER_ERROR 2
#define BES_XIN_PROTOCOL 3

/* Error three-tuples. */

#define BES_err_bad_version err_encode(err_BES, BES_xmain, 1)
/*$M Incorrect abstract link version for BES. */

#define BES_err_bad_sess_id err_encode(err_BES, BES_generic, 3)
/*$M Invalid Batch Entry Services session number. */

#define BES_err_too_many_ids err_encode(err_BES, BES_generic, 4)
/*$M Attempt to allocate too many image identifiers. */

#define BES_err_no_resources err_encode(err_BES, BES_generic, 5)
/*$M Cannot perform this operation. No resources available. */

#define BES_err_batch_exists err_encode(err_BES, BES_generic, 6)
/*$M This batch already exists. */

#define BES_err_batch_not_found err_encode(err_BES, BES_generic, 7)
/*$M This batch does not exist. */

#define BES_err_batch_busy err_encode(err_BES, BES_generic, 8)
/*$M This batch is already in use. */

#define BES_err_batch_not_open err_encode(err_BES, BES_generic, 9)
/*$M This batch is not open. */

#define BES_err_image_exists err_encode(err_BES, BES_generic, 10)
/*$M This image already exists. */

#define BES_err_image_not_found err_encode(err_BES, BES_generic, 11)
/*$M This image does not exist. */
```

+++ BES - Batch Entry Services +++

```
#define BES_err_no_transaction      err_encode(err_BES, BES_generic, 12)
/*$M There is no transaction on this image. */

#define BES_err_already_in_transaction  err_encode(err_BES, BES_generic, 13)
/*$M Can't do requested operation when transaction in process on image */

#define BES_err_doc_exists          err_encode(err_BES, BES_generic, 14)
/*$M This document already exists. */

#define BES_err_bad_pages           err_encode(err_BES, BES_generic, 15)
/*$M Attempt to put page into new document without removing from old. */

#define BES_err_doc_not_found       err_encode(err_BES, BES_generic, 16)
/*$M Document does not exist. */

#define BES_err_ixdir_not_found     err_encode(err_BES, BES_generic, 17)
/*$M Column name record does not exist. */

#define BES_err_internal_rpc_error  err_encode(err_BES, BES_generic, 18)
/*$M Internal RPC error. */

#define BES_err_non_debugging       err_encode(err_BES, BES_generic, 19)
/*$M Debugging not turned on. */

#define BES_err_not_logged_on_to_db err_encode(err_BES, BES_generic, 20)
/*$M Not logged on to BES and/or MKF database. */

#define BES_err_invalid_batch_type  err_encode(err_BES, BES_generic, 21)
/*$M Invalid batch type. */

#define BES_err_ctl_not_found       err_encode(err_BES, BES_generic, 22)
/*$M MKF Ctl record not found.
Your data base was ok when the system was booted, but now the 'ctl'
MKF record is missing. Did someone delete it with MKF_tool? */

#define BES_err_ixval_not_found     err_encode(err_BES, BES_generic, 23)
/*$M Index value record not found */

#define BES_err_bad_relop           err_encode(err_BES, BES_generic, 24)
/*$M The 'relop' parameter passed to BES_find_batches has an invalid value */

#define BES_err_too_many_pages      err_encode(err_BES, BES_generic, 25)
/*$M Attempt to create document with too many pages. */

#define BES_err_too_many_indices    err_encode(err_BES, BES_generic, 26)
/*$M Attempt to create document with too many indices. */
```

+++ BES - Batch Entry Services +++

```
#define BES_err_bad_batch_total      err_encode(err_BES, BES_generic, 27)
/*$M Attempt to compute batch totals on non-numeric field. */

#define BES_err_ixval_cnt            err_encode(err_BES, BES_generic, 28)
/*$M Invalid parameter passed to BES_update_doc: num_indices
When changing the 'num_indices' field of a document, the index values
must be passed to the procedure (array ptr must be non-null). */

#define BES_err_page_cnt            err_encode(err_BES, BES_generic, 29)
/*$M Invalid parameter passed to BES_update_doc: num_pages
When changing the 'num_pages' field of a document, the page array
must be passed to the procedure (array ptr must be non-null). */

#define BES_err_bad_handle          err_encode(err_BES, BES_generic, 30)
/*$M Invalid handle passed to BES. */

#define BES_err_invalid_queue       err_encode(err_BES, BES_generic, 31)
/*$M Attempt to enqueue batch to invalid queue. */

#define BES_err_phase_incomplete    err_encode(err_BES, BES_generic, 32)
/*$M Attempt to commit batch when phase(s) not complete. */

#define BES_err_image_not_verified  err_encode(err_BES, BES_generic, 33)
/*$M Attempt to commit batch when image(s) not verified. */

#define BES_err_wrong_queue         err_encode(err_BES, BES_generic, 34)
/*$M Can't open batch when queue not equal to uncommit(1) or none(0). */

#define BES_err_no_required_index   err_encode(err_BES, BES_generic, 35)
/*$M Can't find required index for document when batch committed. */

#define BES_err_commit_batch_total  err_encode(err_BES, BES_generic, 36)
/*$M Batch total invalid when attempt made to commit batch. */

#define BES_err_index_not_verified  err_encode(err_BES, BES_generic, 37)
/*$M Index not verified when attempt made to commit batch. */

#define BES_err_batch_name_too_long err_encode(err_BES, BES_generic, 38)
/*$M Attempt to create a batch with a batch name which is too long. */

#define BES_err_invalid_batch_cap   err_encode(err_BES, BES_generic, 40)
/*$M Attempt to read/write an image with an invalid batch capability. */

#define BES_err_connection_already_open err_encode(err_BES, BES_generic, 41)
/*$M A connection has previously been opened. */

#define BES_err_connection_not_open err_encode(err_BES, BES_generic, 42)
/*$M This connection is not open. */
```

+++ BES - Batch Entry Services +++

```
#define BES_err_invalid_bulk_data_src    err_encode(err_BES, BES_generic, 43)
/*$M Invalid bulk data source. Should be bulk data immediate. */

#define BES_err_string_too_long          err_encode(err_BES, BES_generic, 44)
/*$M String passed across network exceeds maximum length. */

#define BES_err_batch_too_large          err_encode(err_BES, BES_generic, 45)
/*$M Too many documents or pages in this batch. */

#define BES_err_bad_image_batch_id       err_encode(err_BES, BES_generic, 46)
/*$M Corrupted record in 'batch_image' table.
'batch_id2' is non-null, but does not match 'batch_id' in the batch_image
table. */

#define BES_err_image_in_doc              err_encode(err_BES, BES_generic, 47)
/*$M Can't delete image--image is in document.
An attempt has been made to delete an image in a document but not the
document itself. The image can't be deleted unless the document is
deleted too. This error indicates a programming problem. */

#define BES_err_image_index_exists        err_encode(err_BES, BES_generic, 48)
/*$M This image already has in index associated with it. */

#define BES_err_invalid_image_index      err_encode(err_BES, BES_generic, 49)
/*$M The image index value cannot exceed 239 bytes. */

#define BES_err_no_image_index           err_encode(err_BES, BES_generic, 50)
/*$M This image does not have an associated index value. */

#define BES_err_session_busy              err_encode(err_BES, BES_generic, 51)
/*$M This batch entry session is in use by another client. */

#define BES_err_internal                  err_encode(err_BES, BES_generic, 52)
/*$M Internal BES error. */

#define BES_err_bad_index_type            err_encode(err_BES, BES_generic, 53)
/*$M Invalid index type. */

#define BES_err_commit_failed             err_encode(err_BES, BES_generic, 54)
/*$M Committal failed. Check error status in documents. */

#define BES_err_batch_not_locked          err_encode(err_BES, BES_generic, 55)
/*$M Override flag cannot be TRUE if batch not locked. */

#define BES_err_imagebuf_not_alloc        err_encode(err_BES, BES_generic, 56)
/*$M Image buffer in read,write, or update not allocated */
```

+++ BES - Batch Entry Services +++

```
#define BES_err_cross_committal      err_encode(err_BES,BES_generic,57)
/*$M Error in committing to a compatible target IMS.      */

#define BES_err_too_many_images      err_encode(err_BES, BES_generic, 58)
/*$M Attempt to create too many images for a batch. */

#define BES_err_readonly_batch      err_encode(err_BES,BES_generic,59)
/*$M Attempt update operation on a batch opened as read only. */

#define BES_err_batch_overridden      err_encode(err_BES,BES_generic,60)
/*$M Batch is overridden by another user. */

#define BES_err_batch_for_committal  err_encode(err_BES,BES_generic,61)
/*$M Access of read-only batch is denied; batch is in committal. */

#define BES_err_invalid_write      err_encode(err_BES,BES_generic,62)
/*$M Write image is only permitted in update or create image. */

#define BES_err_not_create_image      err_encode(err_BES,BES_generic,63)
/*$M Create image index is only allowed during image creation. */

#define BES_err_doc_without_page      err_encode(err_BES,BES_generic,64)
/*$M Document has no page. */

#define BES_err_bad_batch_name      err_encode(err_BES, BES_generic, 65)
/*$M Attempt to create a batch with a NULL or invalid batch name */

#define BES_err_already_in_doc      err_encode(err_BES, BES_generic, 66)
/*$M Attempt to move an image which is currently assigned to a doc */

#define BES_err_ixval_len      err_encode(err_BES, BES_generic, 67)
/*$M Length of the index value is greater than the declared maximum. */

#define BES_err_invalid_migr_delay_val  err_encode(err_BES, BES_generic, 68)
/*$M The value of migrate_delay must be greater than or equal to -1 and less than
2147483648 */

#define BES_err_invalid_image_id      err_encode(err_BES, BES_generic, 69)
/*$M Image id must be greater than ASE_INVALID_DOC_ID. */

#define BES_err_debug_only      err_encode(err_BES, BES_generic, 70)
/*$M Message is for BES information and/or debugging purposes only. */

#define BES_err_in_exclusive_use      err_encode(err_BES,BES_xmain,10)
/*$M BES is being used exclusively. */

#define BES_err_exclusive_use_denied  err_encode(err_BES,BES_xmain,11)
/*$M Exclusive use is denied. */
```


+++ BES - Batch Entry Services +++

```
#define BES_err_not_exclusive_logon      err_encode(err_BES,BES_xmain,12)
/*$M Not exclusive logon; Exclusive logoff is denied. */

#define BES_err_invalid_info_spec        err_encode(err_BES,BES_xmain,13)
/*$M The specified BES_info_spec contains invalid data, or is inconsistent with
other data. */

#define BES_err_invalid_info_type        err_encode(err_BES,BES_xmain,14)
/*$M Invalid info_type. */

#define BES_err_end_of_info_list          err_encode(err_BES,BES_xmain,15)
/*$M The link-list of info rec has too few elements. */

#define BES_err_invalid_name_format      err_encode(err_BES,BES_xmain,16)
/*$M The name is not in the expected format of obj:domain:org. */

#define BES_err_folder_no_parent          err_encode(err_BES, BES_generic, 71)
/*$M Parent Folder Node does not exist */

#define BES_err_folder_bad_path_format    err_encode(err_BES, BES_generic, 72)
/*$M Bad Folder path format */

#define BES_err_folder_exceed_max_info_len err_encode(err_BES, BES_generic, 73)
/*$M Maximum Folder info data length exceeded */

#define BES_err_folder_parent_is_batch    err_encode(err_BES, BES_generic, 74)
/*$M Parent Folder is a batch. This is not allowed */

#define BES_err_folder_dup_node           err_encode(err_BES, BES_generic, 75)
/*$M Attempted to create a duplicate folder node */

#define BES_err_folder_no_node_found      err_encode(err_BES, BES_generic, 76)
/*$M No folder found */

#define BES_err_invalid_object_seq_no     err_encode(err_BES, BES_generic, 77)
/*$M Invalid object sequence number. Object sequence number 0 is invalid */

#define BES_err_object_data_too_large     err_encode(err_BES, BES_generic, 78)
/*$M Object data too large. Max. length of object data is 800 */

#define BES_err_invalid_object_type       err_encode(err_BES, BES_generic, 79)
/*$M Invalid object type. 0 is invalid object type */

#define BES_err_object_not_in_batch       err_encode(err_BES, BES_generic, 80)
/*$M Object not in batch */

#define BES_err_folder_exceed_max_base_name_len err_encode(err_BES, BES_generic, 81)
/*$M Folder node name exceeded the MAX limit */
```

+++ BES - Batch Entry Services +++

```
#define BES_err_folder_has_changed      err_encode(err_BES, BES_generic, 82)
/*$M Folder name has been changed. The update failed */

#define BES_err_invalid_object_id      err_encode(err_BES, BES_generic, 83)
/*$M Invalid Object Id. */

#define BES_err_object_data_not_found  err_encode(err_BES, BES_generic, 84)
/*$M Object data not found */

#define BES_err_folder_node_not_empty  err_encode(err_BES, BES_generic, 85)
/*$M Attempt to delete a folder which is not empty */

#define BES_err_folder_no_children_found err_encode(err_BES, BES_generic, 86)
/*$M No children found for a given folder node */

#define BES_err_invalid_move_phase      err_encode(err_BES, BES_generic, 87)
/*$M Invalid current phase (BES_COMMIT, BES_CATALOG or BES_RECOMMIT) of the
    source or destination batch for moving document between batches */

#define BES_err_folder_invalid_parameter err_encode(err_BES, BES_generic, 88)
/*$M Invalid parameter was passed to folder APIs */

#define BES_err_passed_null_ptr        err_encode(err_BES, BES_generic, 89)
/*$M Passed null pointer for input or output for BES APIs. */

#define BES_err_invalid_doc_id         err_encode(err_BES, BES_generic, 90)
/*$M Invalid document id passed to BES APIs. */

#define BES_err_folder_batch_has_child err_encode(err_BES, BES_generic, 91)
/*$M When updating folder node to a batch type, it was found that it had a
    child node */

#define BES_err_object_data_invalid_parameter err_encode(err_BES, BES_generic, 92)
/*$M Invalid parameter was passed to object data APIs */

#define BES_err_image_companion_invalid_parameter err_encode(err_BES, BES_generic, 93)
/*$M Invalid parameter was passed to image companion APIs */

#define BES_err_invalid_companion_id   err_encode(err_BES, BES_generic, 94)
/*$M Invalid image companion Id. ( should be between 1 and 20 ) */

#define BES_err_companion_not_found    err_encode(err_BES, BES_generic, 95)
/*$M Image companion does not exist. */

#define BES_err_invalid_last_companion_id err_encode(err_BES, BES_generic, 96)
/*$M Invalid last image companion Id. ( should be between 0 and 19 ) */
```

+++ BES - Batch Entry Services +++

```
#define BES_err_no_companion_found      err_encode(err_BES, BES_generic, 97)
/*$M No companion object found for specified image. */

#define BES_err_move_doc_same_batch     err_encode(err_BES, BES_generic, 98)
/*$M Invalid call to BES_move_doc to move a document to the same batch */

#define BES_err_folder_parent_is_ancestor err_encode(err_BES, BES_generic, 99)
/*$M When doing a folder update, the parent_id is found to be an ancestor
of the folder node. Or if the parent_id is the same as the folder
node id, this error will also be returned. This will prevent loops
in the folder node parent child relationships. */

#define BES_err_image_companion_exists  err_encode(err_BES, BES_generic, 100)
/*$M The companion image already exists when calling BES_create_image_companion.*

#define BES_err_invalid_level          err_encode(err_BES, BES_generic, 101)
/*$M Invalid extended level */

#define BES_err_no_local_batch_service  err_encode(err_BES, BES_generic, 102)
/*$M No local Batch Services */

#define BES_err_buf_size                err_encode(err_BES, BES_generic, 103)
/*$M Buffer size is too big */

#define BES_err_text_data_size          err_encode(err_BES, BES_generic, 104)
/*$M Text_data field greater than max size allocated */

#endif
```

7. BES SUBROUTINES

7.1. BES_abort_image

```
error_typ
BES_abort_image(ims_sess_id, batch_cap_p)
  ASE_session_number_typ  ims_sess_id; /* IN: session number */           (pg483.24)
  BES_batch_cap_typ      *batch_cap_p; /* IN: batch capabilities */       (pg28.3)

/* BES_abort_image closes the image and throws away all information
about changes being made to this image. If BES_update_image was
being called, then the image goes back to it's state before the
update. If BES_create_image is called, the image information is
deleted.
*/
```

+++ BES - Batch Entry Services +++

7.2. BES_abort_image_companion

```
error_typ
BES_abort_image_companion ( ims_sess_id, batch_cap_p )
ASE_session_number_typ    ims_sess_id; /* IN: session number */      (pg483.24)
BES_batch_cap_typ        *batch_cap_p; /* IN: batch capabilities */  (pg28.3)

/* BES_abort_image_companion closes the companion object and throws away all
information about changes being made to this companion object. If
BES_update_image_companion was being called, then the companion goes back to
it's state before the update. If BES_create_image_companion is called, the
companion object information is deleted.

ERRORS:

    BES_err_batch_not_open
    BES_err_no_transaction
    BES_err_invalid_batch_cap
    BES_err_image_companion_invalid_parameter
*/
```

7.3. BES_alloc_batch_name

```
error_typ
BES_alloc_batch_name(ims_sess_id, batch_type, batch_name)
ASE_session_number_typ    ims_sess_id; /* IN: session number */      (pg483.24)
BES_batch_name_typ        batch_type; /* IN: batch type */          (pg24.3)
char                       batch_name[]; /* OUT: batch name */

/* BES_alloc_batch_name provides the client with the ability
to generate a unique batch name.

ERRORS:

    BES_err_invalid_batch_type
*/
```

7.4. BES_alloc_ids

```
error_typ
BES_alloc_ids(ims_sess_id, num_ids, base_id_p)
ASE_session_number_typ    ims_sess_id; /* IN: session number */      (pg483.24)
long                       num_ids; /* IN: number of ids to allocate.
max allowed is BES_MAX_BATCHDOCS (5000). */
ASE_doc_id_typ            *base_id_p; /* OUT: first id of ids allocated */ (pg483.37)
```

+++ BES - Batch Entry Services +++

/* BES_alloc_ids provides the client with the ability to acquire a block of unique integers to be used as image identifiers. The values of the ids are *base_id_p through *base_id_p + num_ids - 1.

ERRORS:

BES_err_too_many_ids

*/

7.5. BES_close_batch

error_typ

BES_close_batch(ims_sess_id, batch_cap_p)

ASE_session_number_typ ims_sess_id; /* IN: Session number */ (pg483.24)

BES_batch_cap_typ *batch_cap_p; /* IN: Batch capability structure */ (pg28.3)

/* BES_close_batch closes the batch and the current callers lock on the batch, if any.

ERRORS:

BES_err_batch_not_open

*/

7.6. BES_close_connection

error_typ

BES_close_connection(ims_sess_id)

ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)

/* BES_close_connection provides the caller with the ability to close a network connection. */

7.7. BES_close_csum_image

error_typ

BES_close_csum_image(ims_sess_id, batch_cap_p, csum_exists, csum)

ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)

BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)

bool csum_exists; /* IN: whether a checksum is to be put on this image. */ (pg491.22)

long csum; /* IN: the checksum of the image if csum_exists = TRUE. */

+++ BES - Batch Entry Services +++

/* BES_close_csum_image makes the changes of the transaction on the image permanent with an indivisible operation. If BES_close_csum_image is not called and the system is rebooted, all transaction changes for the image are lost. Note that if an image is being updated by calling BES_update_image, BES_write_image multiple times with pieces of the image, and BES_close_csum_image, you can't generate a corrupt image by writing the first half of the image and then encountering a system reboot.

BES_csum_image allows setting a checksum on the image data which has been previously created. The checksum must be computed with the CKS_compute_csum routine (see the CKS module). If checksums are used, it's recommended that the checksum be computed at the earliest possible moment to maximize their effectiveness.

When a checksum is put on an image which is committed, the checksum is inspected prior to writing the image to optical disk. If the checksum is incorrect, an error will be generated and the page will not be written to optical disk.

Images in a batch must be consistent as to whether the images are check summed - either all of the images are check-summed, or none of the images are check-summed. To enforce checksum consistency, BES_close_csum_image will ignore the check sum if the batch already contains non-checksummed images. Similarly, a call to BES_close_csum_image with csum_exists = FALSE (or a call to BES_close_image), and the batch contains check-summed images, will cause the check-sum of all images to be removed.

ERRORS:

BES_err_batch_not_open
BES_err_no_transaction
BES_err_invalid_batch_cap

*/

7.8. BES_close_image

error_typ

BES_close_image(ims_sess_id, batch_cap_p)

ASE_session_number_typ ims_sess_id; /* IN: session number */

(pg483.24)

BES_batch_cap_typ *batch_cap_p; /* IN: batch capabilities */

(pg28.3)

/* BES_close_image makes the changes of the transaction on the image permanent with an indivisible operation. If BES_close_image is not called and the system is rebooted, all transaction changes for the image are lost. Note that if an image is being updated by calling BES_update_image, BES_write_image multiple times with pieces of the image, and BES_close_image, you can't generate a corrupt image by writing

+++ BES - Batch Entry Services +++

the first half of the image and then encountering a system reboot.

ERRORS:

BES_err_batch_not_open
BES_err_no_transaction
BES_err_invalid_batch_cap

*/

7.9. BES_close_image_companion

error_typ

BES_close_image_companion (ims_sess_id, batch_cap_p)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capabilities */ (pg28.3)

/* BES_close_image_companion makes the changes of the transaction on the companion object permanent with an indivisible operation. If BES_close_image_companion is not called and the system is rebooted, all transaction changes for the companion object are lost. Note that if an companion object is being updated by calling BES_update_image_companion, BES_write_image_companion multiple times with pieces of the companion object, and BES_close_image_companion, you can't generate a corrupt companion object by writing the first half of the companion object and then encountering a system reboot.

ERRORS:

BES_err_batch_not_open
BES_err_no_transaction
BES_err_invalid_batch_cap
BES_err_image_companion_invalid_parameter

*/

7.10. BES_commit_batch_now

error_typ

BES_commit_batch_now(ims_sess_id, batch_cap_p)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capabilities */ (pg28.3)

/* BES_commit_batch_now provides the client with the ability to synchronously commit a batch. In order for this routine to execute, the batch must be in a state which is acceptable for committal. See the description of BES_enqueue_batch for the requirements of batches to be committed.

+++ BES - Batch Entry Services +++

If the batch is in an acceptable state, committal will be attempted.
If all documents are successfully committed, then the batch is deleted.
If any document fails committal, the batch remains open and will contain the documents which failed committal.

The batch is placed in the IN_PROGRESS queue. This allows committal to be automatically restarted by the daemon in the event of a crash. The client will not be notified of completion if the system crashes.

If a network error or cache full error occurs while the batch is committing, this routine will retry the function forever until the commit is successful.

ERRORS:

BES_err_batch_not_open
BES_err_phase_incomplete
BES_err_image_not_verified
BES_err_commit_batch_total
BES_err_ixval_not_found
BES_err_no_required_index
BES_err_index_not_verified

*/

7.11. BES_create_batch

error_typ

```
BES_create_batch(ims_sess_id, batch_name, dcl_name, batch_hdr_p, batch_cap_p)
  ASE_session_number_typ  ims_sess_id; /* IN: Session number */ (pg483.24)
  char                    batch_name[]; /* IN: Name of batch */
  char                    dcl_name[]; /* IN: Name of document class batch
                                     will belong to. */
  BES_hdr_desc_typ       *batch_hdr_p; /* OUT: Header for batch created */ (pg31.43)
  BES_batch_cap_typ      *batch_cap_p; /* OUT: Capability for this batch. (pg28.3)
                                     Used to access this batch on
                                     subsequent calls to this
                                     library */
```

/* BES_create_batch provides the client with the ability to create a batch with the specified name. Upon success, BES_create_batch returns a newly created batch header record and batch capabilities. The batch is left open upon return so that the client is the only user with the ability to update or add to this batch.

The batch capabilities structure is a key which identifies the batch (by batch id) and the caller who has this batch open, and also authorizes only

+++ BES - Batch Entry Services +++

this caller to update the batch. This structure is passed in to all other entry points which require an open batch as a prerequisite to the call. This structure is saved on magnetic disk, and will be in effect even after the system is rebooted. Note that the capabilities structure can be overridden in the BES_open_batch call with the open_mode flag.

The maximum number of pages which may be put into a batch is BES_MAX_BATCHPAGES (5000). Therefore in a single batch you can put 5000 one page documents, or 10 999 page documents, etc.

ERRORS:

BES_err_no_resources
BES_err_batch_exists
BES_err_batch_name_too_long

*/

7.12. BES_create_doc

error_typ

BES_create_doc(ims_sess_id, batch_cap_p, doc_desc_p, page_list_ary,
index_values_ary, doc_id_p)

ASE_session_number_typ	ims_sess_id;	/* IN: session number */	(pg483.24)
BES_batch_cap_typ	*batch_cap_p;	/* IN: batch capabilities */	(pg28.3)
BES_doc_desc_typ	*doc_desc_p;	/* IN/OUT: document description.	(pg34.30)
		All fields passed in except doc_id. Doc_id field set on return. */	
ASE_doc_id_typ	page_list_ary[];	/* IN: array of image ids which are to become pages of the document */	(pg483.37)
BES_ixval_desc_typ	index_values_ary[];	/* IN: array of index values */	(pg33.37)
ASE_doc_id_typ	*doc_id_p;	/* OUT: Document id of document created (same as doc_desc_p->doc_id) */	(pg483.37)

/* BES_create_doc provides the client with the ability to create a document from images and indices. Batch totals, if any, will be updated with the incoming index values.

ERRORS:

BES_err_batch_not_open
BES_err_doc_exists
BES_err_no_resources
BES_err_bad_pages
BES_err_too_many_pages

+++ BES - Batch Entry Services +++

BES_err_too_many_indices

*/

7.13. BES_create_image

error_typ

BES_create_image(ims_sess_id, batch_cap_p, image_desc_p)

ASE_session_number_typ	ims_sess_id;	/* IN: session number */	(pg483.24)
BES_batch_cap_typ	*batch_cap_p;	/* IN: batch capability */	(pg28.3)
BES_image_desc_typ	*image_desc_p;	/* IN: image descriptor defining image to create */	(pg32.41)

/* BES_create_image creates an image in the cache, leaves it open, and initiates a "transaction" on the image. All changes made under control of the transaction do not take effect until either BES_close_image or BES_close_csum_image is made. These changes include the new *image_desc_p passed in on this call, BES_write_image calls, and anything which requires an open image to execute.

In *image_desc_p, the fields are used as follows:

The following are input by the client:

image_id
image_length
image_type
image_ver_stat

The following are set to zero since other entry points set these values:

doc_id
page_id
index_len

The following is not used:

end_of_doc

Only one image can be open at a time. Note that the indication that the image is open is saved in the session (ims_sess_id), and subsequent calls which read and write images do not take an image id or description as an argument.

When calling BES_create_image() through the WAL library, it uses one DBP buffer. When it is used in combination with DOC_batch_create, CSM_create_object, CSM_open_object, or CSM_open_csum_object, there could be a need to increase the number of DBP buffers on the IMS server. There is the

+++ BES - Batch Entry Services +++

potential for deadlocking the IMS if multiple applications are running at the same time and each program calls more than one of the entry points above and there are few DBP buffers on the IMS server.

ERRORS:

BES_err_batch_not_open
BES_err_image_exists
BES_err_no_resources
BES_err_already_in_transaction

*/

7.14. BES_create_image_companion

error_typ

```
BES_create_image_companion ( ims_sess_id, batch_cap_p, companion_desc_p )
ASE_session_number_typ    ims_sess_id;          /* IN: session number */      (pg483.24)
BES_batch_cap_typ        *batch_cap_p;         /* IN: batch capability */   (pg28.3)
BES_companion_desc_typ   *companion_desc_p; /* IN: companion descriptor (pg37.35)
                               defining image companion
                               to create */
```

/* BES_create_image_companion creates an companion object in the cache, leaves it open, and and initiates a "transaction" on the companion data. All the changes made under control of the transaction do not take effect until BES_close_image_companion call is made. These changes include the new *companion_desc_p passed in on this call, BES_write_image_companion calls, and anything which requires an open companion object to execute.

In *companion_desc_p, the fields are used as follows:

The following are input by the client:
image_id
companion_id
data_length

Only one companion object or image can be open at a time. Note that the indication that the companion object or image is open is saved in the session (ims_sess_id), and subsequent calls which read and write companion object do not take an companion description as an argument.

ERRORS:

BES_err_batch_not_open
BES_err_image_not_found
BES_err_image_companion_exists
BES_err_no_resources

+++ BES - Batch Entry Services +++

BES_err_already_in_transaction
BES_err_invalid_companion_id
BES_err_image_companion_invalid_parameter

*/

7.15. BES_create_image_index

error_typ

BES_create_image_index(ims_sess_id, batch_cap_p, ix_len, ix_val)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)
unsigned short ix_len; /* IN: index length */
char ix_val[]; /* IN: index value */

/* BES_create_image_index provides the caller with the ability to store indexing information with an associated image. This function will be useful with, for example, aperture card scanning, where the indexing information is part of the image itself.

The image MUST be open by virtue of BES_create_image call. This change does not get saved until BES_close_image or BES_close_csum_image is called.

Note: Use BES_modify_image_index to change the image index after it has been created.

ERRORS:

BES_err_batch_not_open
BES_err_batch_busy
BES_err_no_transaction
BES_err_not_create_image
BES_err_image_index_exists
BES_err_invalid_image_index

*/

7.16. BES_delete_batch

error_typ

BES_delete_batch(ims_sess_id, batch_cap_p)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)

/* BES_delete_batch deletes all state related to the previously opened batch. All images associated with the batch are deleted from the cache.

+++ BES - Batch Entry Services +++

ERRORS:

BES_err_batch_not_open
*/

7.17. BES_delete_doc

error_typ

```
BES_delete_doc(ims_sess_id, batch_cap_p, doc_id, del_images_flag)
  ASE_session_number_typ  ims_sess_id;      /* IN: session number */      (pg483.24)
  BES_batch_cap_typ       *batch_cap_p;     /* IN: batch capabilities */  (pg28.3)
  ASE_doc_id_typ          doc_id;           /* IN: document id */        (pg483.37)
  bool                    del_images_flag; /* IN: TRUE=>delete images,   (pg491.22)
                                     FALSE=>don't delete images--
                                     images can be put into
                                     different documents. */
```

/* BES_delete_doc provides the client with the ability to delete a document. The function will unconditionally delete all assembly and indexing information associated with the document.

If del_images_flag is FALSE, then the images associated with the doc will remain in the bes cache, and their images records in the transient database will not be deleted, but will be updated to indicate that they belong to no document.

ERRORS:

BES_err_batch_not_open
BES_err_doc_not_found
*/

7.18. BES_delete_image

error_typ

```
BES_delete_image(ims_sess_id, batch_cap_p, image_id)
  ASE_session_number_typ  ims_sess_id;      /* IN: session number */      (pg483.24)
  BES_batch_cap_typ       *batch_cap_p;     /* IN: batch capability */    (pg28.3)
  ASE_doc_id_typ          image_id;        /* IN: image id of image to delete */ (pg483.37)
```

/* BES_delete_image deletes the image descriptor associated with the specified image and frees space occupied by the image data.

The image to be deleted must not be part of a document. To

+++ BES - Batch Entry Services +++

delete an image that is part of a document, the client must first update the document to not contain the image and then delete the image.

ERRORS:

BES_err_batch_not_open
BES_err_image_not_found
BES_err_image_in_doc

*/

7.19. BES_delete_image_companion

error_typ

BES_delete_image_companion (ims_sess_id, batch_cap_p, image_id, companion_id)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)
ASE_image_id_typ image_id; /* IN: image id of image, companion (pg483.33)
 object is associated with */
ASE_page_number_typ companion_id; /* IN: companion id of the companion (pg483.41)
 object to delete */

/* BES_delete_image_companion deletes the companion descriptor associated with the specified image and specified companion_id and frees space occupied by the companion object data.

It returns error if the specified companion object is in transaction.

ERRORS:

BES_err_batch_not_open
BES_err_image_not_found
BES_err_image_companion_invalid_parameter

*/

7.20. BES_delete_image_index

error_typ

BES_delete_image_index(ims_sess_id, batch_cap_p, image_id)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)
ASE_doc_id_typ image_id; /* IN: image id */ (pg483.37)

/* BES_delete_image_index provides the caller with the ability to delete the index data associated with a specified image. The image need not be open.

+++ BES - Batch Entry Services +++

ERRORS:

BES_err_batch_not_open
BES_err_batch_busy
BES_err_image_not_found

*/

7.21. BES_delete_object_data

error_typ

```
BES_delete_object_data (ims_sess_id, batch_cap_p, object_type, object_id,
                        num_of_sequence_ids, sequence_ids)
ASE_session_number_typ    ims_sess_id;          /* IN: session number */(pg483.24)
BES_batch_cap_typ        *batch_cap_p;         /* IN: batch capability */(pg28.3)
BES_object_type_typ      object_type;          /* IN: type of object */(pg36.48)
BES_object_id_typ        object_id;            /* IN: object id */      (pg37.1)
unsigned long            num_of_sequence_ids;  /* IN: number of
                                                sequence id's
                                                in sequence_id
                                                array */
BES_sequence_id_typ      sequence_ids[];       /* IN: An array of      (pg37.2)
                                                * sequence id's to
                                                * delete */
```

/*

BES_delete_object_data provides the client with the means to delete associated information with a batch record, a batch document record, a batch image record, a batch phase or a user defined object type. Multiple buffers may have been associated with one batch object by the using of sequence numbers.

A deletion can be done of all the associated data with a specific batch (batch_cap_p). This can be done by setting the object_type to BES_OBJECT_TYPE_ALL, object_id to BES_OBJECT_ID_ALL and number of sequence id to zero (0). A deletion may also be done for all the associated data with a specific batch and object type. This can be done by setting the object_id to BES_OBJECT_ID_ALL and number of sequence id to zero (0). To delete all entries associated with a batch, object_type, and object_id, the number of sequence ids should be set to zero(0). Individual records may also be deleted by specifying the object_type, object_id, number of sequence ids and an array of sequence ids.

Input:

ims_sess_id - session number returned by logon
batch_cap_p - batch capability
object_type - object type if set to NULL all

+++ BES - Batch Entry Services +++

information associated with
a this batch will be deleted
object_id - object_id if set to NULL all
num_of_sequence_ids - Number of sequence id's in sequence_id array
sequence_ids - Array of sequence id's to delete

ERRORS:

BES_err_batch_not_open
BES_err_no_such_object
BES_err_invalid_object_seq_no

*/

7.22. BES_enqueue_batch

error_typ

BES_enqueue_batch(ims_sess_id, batch_cap_p, queue_selector)

ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability structure */ (pg28.3)
BES_commit_queue_typ queue_selector; /* IN: queue type */ (pg25.25)

/* BES_enqueue_batch provides the client with the ability to
place a previously-opened batch into the specified queue.

BES_enqueue_batch will typically be used to enqueue a
batch for committal but can also be used to return a
batch that failed committal to the uncommit queue.

BES_enqueue_batch can also be used to place a batch in the
OCR queue. No validation is done by BES if the OCR queue
is specified.

If the commit queue is specified, the batch will be enqueued for
committal provided that the batch is in an acceptable state.
A batch is in an acceptable state if it satisfies the following
requirements:

Each document in the batch must be correctly defined, i.e. all
its pages must exist and values must have been assigned to all
required indexing fields.

If the verify_images field in the batch header is TRUE,
this routine will check that the image_ver_stat field
in each BES_image_desc_typ record is equal to BES_GOOD.

If the batch_total field in the batch header is TRUE, and an index also
has the batch_tot_flag in the BES_ixdir_desc_typ record set to true, then
the batch totals for this index will be verified (see the description of

+++ BES - Batch Entry Services +++

batch_total in the BES_dyn_hdr_desc_typ record).

If the verify_indices field in the batch header is TRUE and at least one index of the document class also has the verify_flag set to TRUE (this flag is available in either the BES_ixdir_desc_typ and INX_dcl_index_desc_typ records, and has the same value in both places), this routine checks that the "phase_status[BES_INDEX_VER]" of each BES_doc_desc_typ record is equal to BES_DONE. Note that this requirement is not made of the phase_status array in the BES_dyn_hdr_desc_typ record.

For every batch, all elements of the phase_status array of the BES_dyn_hdr_desc_typ record except those for the BES_COMMIT, BES_CATALOG, and BES_RECOMMIT phases must be either BES_DONE or BES_NOT_NECESSARY. Note that this requirement is not made of the phase_status array in the BES_doc_desc_typ record.

If all the above conditions are met, the batch is enqueued for committal and closed. If any condition is not met, the batch is not enqueued and it also remains open for further modifications.

ERRORS:

BES_err_batch_not_open
BES_err_invalid_queue
BES_err_phase_incomplete
BES_err_image_not_verified
BES_err_commit_batch_total
BES_err_ixval_not_found
BES_err_no_required_index
BES_err_index_not_verified

*/

7.23. BES_extended

error_typ

BES_extended(ims_sess_id, level)

ASE_session_number_typ ims_sess_id; /* IN: session number */
short level;

(pg483.24)

/* BES_extended provides the client with the extended indexes. This entry point should be called after BES_logon and is affect for that session. The following are some possible level values as defined in the BES.defs file:

BES_level_0 (0)
BES_level_1 (1)

+++ BES - Batch Entry Services +++

The level is set to BES_level_0 by default.

BES_level_0: No extended indexes

BES_level_1: New extended indexes (F_DOCLOCATION and F_DOCFORMAT support RCI 1448, 1450)

ERRORS:

BES_err_invalid_level

*/

7.24. BES_find_batches

error_typ

BES_find_batches(ims_sess_id, batch_name, relop, filter_p, max_hdrs,
num_hdrs_p, batch_hdrs, done_p)

ASE_session_number_typ	ims_sess_id;	/* IN: Session number */	(pg483.24)
char	batch_name[];	/* IN: Starting batch name. Use zero length string for all batches */	
BES_relop_typ	relop;	/* IN: BES_EQUAL, BES_GEQ, or BES_GT */	(pg25.44)
BES_filter_typ	*filter_p;	/* IN: Filter conditions. All batches returned satisfy conditions in this filter. */	(pg27.39)
long	max_hdrs;	/* IN: maximum # headers to return */	
long	*num_hdrs_p;	/* OUT: actual # headers returned */	
BES_hdr_desc_typ	batch_hdrs[];	/* OUT: headers returned */	(pg31.43)
bool	*done_p;	/* OUT: TRUE=>no more headers exist for given inputs, FALSE=>more headers may exist. */	(pg491.22)

/* BES_find_batches returns batch headers that match the specified filter. This entry point, used for reports, allows the client to access one or more batch headers without opening the batches.

The client can use this entry point to, for example, return the headers of all active batches, all uncommitted batches, and all batches with a particular error state.

The batch_name and relop parameters specify the starting point for the search. BES_find_batches will return a sequence of no more than max_hdrs. The *done_p boolean is set to TRUE when no batch headers satisfying the condition are found. To completely satisfy the search conditions, the call should be made repeatedly while BES_find_batches sets the *done_p to FALSE.

ERRORS:

+++ BES - Batch Entry Services +++

BES_err_bad_relop
BES_err_batch_not_found

*/

7.25. BES_find_docs

error_typ

```
BES_find_docs(ims_sess_id, batch_cap_p, starting_doc_id, max_docs,
              get_pages_flag, get_indices_flag, num_docs_p, doc_list_p_p, done_p)
ASE_session_number_typ  ims_sess_id;      /* IN: session number */      (pg483.24)
BES_batch_cap_typ      *batch_cap_p;     /* IN: batch capabilities */  (pg28.3)
ASE_doc_id_typ         starting_doc_id;   /* IN: find docs with doc_id (pg483.37)
                                greater than this. Set to
                                0 to find all documents */
long                   max_docs;          /* IN: maximum documents to
                                return */
bool                   get_pages_flag;    /* IN: if want pages (image ids) (pg491.22)
                                returned. */
bool                   get_indices_flag;  /* IN: if want indicies returned*/ (pg491.22)
long                   *num_docs_p;       /* OUT: #documents returned */
BES_doc_list_typ      **doc_list_p_p;    /* OUT: documents returned */  (pg35.1)
bool                   *done_p;          /* OUT: TRUE=>more documents exist. (pg491.22)
                                FALSE=>all documents
                                returned already. */
```

/* BES_find_docs provides the client with the ability to find one or more document descriptors. It will return at most max_docs descriptors. The doc_id in the first descriptor will be greater than starting_doc_id.

The client can specify whether to return the page and/or index value array in the BES_doc_list_typ record.

Memory for the BES_doc_list_typ is allocated by BES, and the caller should return this memory when done by calling "BES_return_docs".

BES_find_docs sets a boolean (done_p) to indicate whether or not all document descriptors satisfying the query have been returned. If done_p is FALSE, the doc_id field from the last returned document descriptor should be the starting_doc_id in the subsequent call.

ERRORS:

BES_err_doc_not_found

*/

+++ BES - Batch Entry Services +++

7.26. BES_find_images

error_typ

```
BES_find_images(ims_sess_id, batch_cap_p, starting_image_id, max_images,
                num_images_p, image_descs, done_p)
ASE_session_number_typ  ims_sess_id;      /* IN: session number */      (pg483.24)
BES_batch_cap_typ       *batch_cap_p;     /* IN: batch capabilities */  (pg28.3)
ASE_doc_id_typ          starting_image_id; /* IN: starting image id. Image
                                descriptors containing
                                image ids greater than
                                this number will be
                                returned. */
long                    max_images;        /* IN: max #images to return */
long                    *num_images_p;    /* OUT: #images returned */
BES_image_desc_typ      image_descs[];    /* OUT: images returned. Client
                                must allocate this memory
                                large enough to hold
                                max_images. */
bool                    *done_p;          /* OUT: FALSE=>more exist for
                                query, TRUE=>no more
                                for this query. */
```

/* BES_find_images returns an array of image descriptors.
The image_id in the first image descriptor will be greater
than starting_image_id.

BES_find_images sets a boolean (done_p) to indicate whether
or not all image descriptors satisfying the query have been
returned. If done_p is FALSE, the client should use the
image_id in the last image descriptor as the
starting_image_id in the subsequent call.

ERRORS:

```
BES_err_batch_not_open
BES_err_image_not_found
```

*/

7.27. BES_find_object_data

error_typ

```
BES_find_object_data (ims_sess_id, batch_cap_p, object_type, object_id,
                     sequence_id, last_spec_p, num_to_get, num_returned_p,
                     object_rec_p_p, done_p)
ASE_session_number_typ  ims_sess_id;      /* IN: session number */      (pg483.24)
BES_batch_cap_typ       *batch_cap_p;     /* IN: batch capability */    (pg28.3)
BES_object_type_typ     object_type;      /* IN: type of object */     (pg36.48)
BES_object_id_typ       object_id;        /* IN: object id. */         (pg37.1)
```

+++ BES - Batch Entry Services +++

```

BES_sequence_id_typ      sequence_id;      /* IN: starting sequence id. to (pg37.2)
                        *      get associated data for
                        */
BES_object_spec_typ      *last_spec_p;      /* IN: return object data records (pg37.9)
                        *      beginning with the record
                        *      logically after this
                        *      record. */
unsigned long            num_to_get;      /* IN: maximum number of records
                        *      to return
                        */
unsigned long            *num_returned_p; /* OUT: number of records
                        *      returned
                        */
BES_object_rec_typ      **object_rec_p_p; /* OUT: linked list of records (pg37.17)
                        *      of BES_object_rec_typ
                        */
bool                    *done_p;         /* OUT: If TRUE (1), no more (pg491.22)
                        *      records for this query
                        */

```

/*
BES_find_object_data provides the client with the means to retrieve associated information with a batch record, a batch document record, a batch image record, a batch phase or user defined object type. Multiple buffers may have been associated with one batch object by using the sequence numbers. A specific data record may be retrieved by specifying the object_type, object_id, sequence_id and setting the num_to_get to 1. If *done_p is returned as FALSE, the caller may specify the data from the last record in object_rec_pp linked list as the input last_spec_p to continue the query from where it left off. Memory for the object_rec_p_p linked list is allocated by batch services, and the caller must return this memory by calling BES_free_object_data. BES_SEQUENCE_ID_ALL, BES_OBJECT_ID_ALL and BES_OBJECT_TYPE_ALL should be used for sequence_id, object_id and object_type for wild card queries.

Input:

```

ims_sess_id - session number returned by logon
batch_cap_p - batch capability
object_type - object type if set to NULL all
              object data associated with
              a this batch will be returned
object_id   - object id. if set to NULL all
              object data associated with this
              batch and object type will be returned
sequence_id - starting sequence id to get associated
              data from.
num_to_get  - maximum number of records to return.

```

+++ BES - Batch Entry Services +++

`last_spec_p` - return object data records beginning with the record logically after this one. Input pointer as NULL to start.

Output:

`num_returned_p` - number of records returned in the `object_rec_p_p` linked list
`object_rec_p_p` - linked list of object data
`done_p` - If a TRUE (1) value is returned, then there are no more records for this query

ERRORS:

`BES_err_batch_not_open`
`BES_err_no_resources`
`BES_err_invalid_object_sequence_no`

*/

7.28. `BES_folder_create_node`

`error_typ`

```
BES_folder_create_node (ims_sess_id, full_node_name, info_len, info_data,
                        is_batch, node_id_p, creation_time_p)
ASE_session_number_typ    ims_sess_id;      /* IN: session number */      (pg483.24)
char                      *full_node_name;  /* IN: node name */
unsigned long             info_len;         /* IN: length of info data */
any_ptr                   info_data;       /* IN: user info data */    (pg491.24)
bool                      is_batch;        /* IN: is batch node */    (pg491.22)
BES_node_id_typ           *node_id_p;      /* OUT: generate node id */ (pg36.10)
unsigned long             *creation_time_p; /* OUT: creation time */
```

/*

Creates a node in the `batch_folder_node` table. If the `is_batch` flag is set to TRUE (1), the base name of `full_node_name` will be used to query the `batch_header` table to find out if the batch exist and to get the `batch_id`. The parent node can not be a batch node.

Input:

`ims_sess_id` - session number returned by logon
`full_node_name` - The `full_node_name` string specifies the name of the node in an absolute path format. For example a `full_node_name` could be `/top_folder` or `/top_folder/batch1`. This assumes that the parent node exist. The `/top_folder` node must be created prior to `/top_folder/batch1`. The `full_node_name` string must always start with the slash, `/` character.
`info_len` - The `info_len` variable is used to specify the length of the `info_data` buffer. The value of `info_len` and `info_data` may be changed using `BES_folder_update_node`. The maximum length of the info buffer is 600.

+++ BES - Batch Entry Services +++

info_data - Contains user info data to be associated with this node.
is_batch - If FALSE (0), then this node is a non- batch. If the is_batch parameter is set to TRUE (1), then the batch will automatically be cleaned up. When a batch has been committed successfully and all the entries in the batch table have been deleted, the associated batch node(s) in the batch_folder_node table will also be deleted.

Output:

node_id_p - A pointer to the node number which uniquely identifies this batch folder node.
creation_time_p - Creation time in seconds

ERRORS:

BES_err_folder_no_parent
BES_err_folder_bad_path_format
BES_err_folder_exceed_max_info_len
BES_err_batch_not_found
BES_err_folder_parent_is_batch
BES_err_folder_dup_node
BES_err_folder_invalid_parameter

*/

7.29. BES_folder_delete_node

error_typ

BES_folder_delete_node (ims_sess_id, full_node_name, node_id)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
char *full_node_name; /* IN: folder name */
unsigned long node_id; /* IN: folder id */

/*

BES_folder_delete_node allows a client to delete empty folder nodes. Deleting of folder nodes must be done with care since another client may be directly or indirectly referencing the nodes being deleted.

Note: Using a full_node_name instead of node_id is less efficient. When possible, a node_id should be used because it only requires one query of the database. The full_node_name will require N+1 queries where N is equal to the number of levels of nodes.

Input:

ims_sess_id - session number returned by logon
full_node_name - specifies which folder node to be delete

+++ BES - Batch Entry Services +++

`node_id` - If `full_node_name` is set to a NULL pointer then the `node_id` parameter is used as the search value for the deletion.

ERRORS:

`BES_err_folder_bad_path_format`
`BES_err_folder_no_node_found`
`BES_err_folder_node_not_empty`

*/

7.30. `BES_folder_find_children`

`error_typ`

```
BES_folder_find_children(ims_sess_id, full_parent_name, parent_id,
                        last_child, max_children, num_children_p,
                        children_desc, done_p)
ASE_session_number_typ    ims_sess_id;          /* IN: session number */    (pg483.24)
char                     *full_parent_name;      /* IN: parent name */
unsigned long             parent_id;              /* IN: parent id */
char                     *last_child;           /* IN: last child */
unsigned long             max_children;          /* IN: max children */
unsigned long             *num_children_p;       /* OUT: number of children
                                                    */
BES_folder_desc_typ      children_desc[];       /* OUT: folder description (pg36.25)
                                                    */
bool                     *done_p;                (pg491.22)
```

/*

`BES_folder_find_children` will return a list of children folder nodes. The children description will be returned in a code point (in most cases alphanumeric) order. Only the immediate children will be returned.

Note: Using a `full_parent_name` instead of `parent_id` is less efficient. When possible, a `parent_id` should be used because it only requires one query of the database. The `full_parent_name` will require N+1 queries where N is equal to the number of levels of nodes.

Input:

`ims_sess_id` - session number returned by logon
`full_parent_name` - Full path name of the parent node. To get the children of the root node use the / as the `full_parent_name`. If the `full_parent_name` is set to a NULL pointer use the `parent_id` parameter.
`parent_id` - If the `full_parent_name` is set to a NULL

+++ BES - Batch Entry Services +++

pointer then use the `parent_id` to do the search.

`last_child` - This is typically set to a NULL string value; but if more than one of the `BES_folder_find_children` calls are needed to get a complete list of the children, the `last_child` parameter is used as a pointer to know where to search from. The children folder nodes after the `last_child` will be returned.

`max_children` - maximum number of children folder node descriptions to return

Output:

`num_children_p` - Actual number of children folder node descriptions returned

`children_desc` - An array of children folder nodes. This space must be allocated prior to calling `BES_folder_find_children`. An array of `BES_folder_desc_typ` with `max_children` elements must be allocated by the client.

`done_p` - TRUE if no more children can be found

ERRORS:

`BES_err_folder_no_parent`
`BES_err_folder_bad_path_format`
`BES_err_folder_no_children_found`
`BES_err_folder_invalid_parameter`

*/

7.31. `BES_folder_get_node`

`error_typ`

`BES_folder_get_node (ims_sess_id, full_node_name, node_id, folder_desc_p)`

<code>ASE_session_number_typ</code>	<code>ims_sess_id;</code>	<code>/* IN: session number */</code>	<code>(pg483.24)</code>
<code>char</code>	<code>*full_node_name;</code>	<code>/* IN: node name */</code>	
<code>unsigned long</code>	<code>node_id;</code>	<code>/* IN: node id */</code>	
<code>BES_folder_desc_typ</code>	<code>*folder_desc_p;</code>	<code>/* OUT: folder description */</code>	<code>(pg36.25)</code>

/*

`BES_folder_get_node` allows a client to get a single folder node record based on the full node name (ie. `/xx1/batch2`) or the node id.

Note: Using a `full_node_name` instead of `node_id` is less efficient. When possible, a `node_id` should be used because it only requires one query of the database. The `full_node_name` will require N+1 queries where N is equal to the number of levels of nodes.

Input:

`ims_sess_id` - session number returned by logon

+++ BES - Batch Entry Services +++

Input:

ims_sess_id - session number returned by logon
object_rec_p_p - linked list of object data of type BES_object_rec_typ
returned by BES_find_object_data

*/

7.35. BES_get_image_companion_ids

error_typ

BES_get_image_companion_ids (ims_sess_id, batch_cap_p, image_id,
last_companion_id, num_to_get, num_returned_p,
companion_ids_p, done_p)

ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)

BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)

ASE_image_id_typ image_id; /* IN: image id */ (pg483.33)

ASE_page_number_typ last_companion_id; /* IN: the last companion id (pg483.41)

returned, get the companion
ids greater than this
companion_id
If 0 then get all companion
ids */

unsigned long num_to_get; /* IN: number of companion ids
to return */

unsigned long *num_returned_p; /* OUT: actual number of companion
ids returned */

ASE_page_number_typ *companion_ids_p; /* OUT: array of companion ids (pg483.41)
to save the returned
companion_ids */

bool *done_p; /* OUT: If TRUE, no more companion (pg491.22)
objects for this query */

/* BES_get_image_companion_ids returns number of companion objects and
associated companion ids for specified image.

ERRORS:

BES_err_batch_not_open

BES_err_image_not_found

BES_err_no_resources

BES_err_invalid_last_companion_id

BES_err_image_companion_invalid_parameter

*/

7.36. BES_get_image_index

error_typ

BES_get_image_index(ims_sess_id, batch_cap_p, image_id, max_ix_len,

+++ BES - Batch Entry Services +++

```

        act_ix_len_p, ix_val_p, done_p)
ASE_session_number_typ  ims_sess_id;    /* IN:  session number */           (pg483.24)
BES_batch_cap_typ       *batch_cap_p;    /* IN:  batch capability */         (pg28.3)
ASE_doc_id_typ          image_id;        /* IN:  image id */                 (pg483.37)
unsigned short          max_ix_len;      /* IN:  maximum index length */
unsigned short          *act_ix_len_p;    /* OUT: actual index length */
char                    ix_val_p[];      /* OUT: index value */
bool                    *done_p;        /* OUT: TRUE=>no more indexes for
                                         this query, FALSE=>more
                                         indexes for query */
/* BES_get_image_index provides the caller with the ability to
   retrieve indexing information associated with a specified
   image. The image need not be open.

   The ix_val_p pointer must reference a memory area which is at least
   max_ix_len bytes in size. The maximum size needed for an image
   index is BES_MAX_INDEX_LEN bytes.

   ERRORS:

   BES_err_batch_not_open
   BES_err_batch_busy
   BES_err_image_not_found
   BES_err_no_image_index
*/

```

7.37. BES_get_info

```

error_typ
BES_get_info(ims_sess_id, batch_cap_p, info_type, last_spec_p, num_to_get,
             num_returned_p, info_rec_p_p, done_p)
ASE_session_number_typ  ims_sess_id;    /* IN:  session number */           (pg483.24)
BES_batch_cap_typ       *batch_cap_p;    /* IN:  batch capability */         (pg28.3)
BES_info_typ            info_type;        /* IN:  information type */         (pg35.19)
BES_info_spec_typ       *last_spec_p;    /* IN:  return info records
                                         beginning with the record
                                         logically after this one.
                                         Input pointer as NULL to
                                         start at beginning. */
unsigned short          num_to_get;      /* IN:  max #records to get */
unsigned short          *num_returned_p; /* OUT: #records returned */
BES_info_rec_typ        **info_rec_p_p;  /* OUT: linked list of records
                                         returned. */           (pg35.44)
bool                    *done_p;        /* OUT: TRUE=>no more records for
                                         query, FALSE=>more records
                                         exist. */           (pg491.22)

```

+++ BES - Batch Entry Services +++

/* BES_get_info provides the client with the ability to retrieve info associated with a batch, a document in a batch, an image in a batch, an index type associated with the class of the batch, or an index value associated with a specific document of a batch.

If *done_p is returned as FALSE, the caller may specify the data from the last record in info_rec_p_p linked list as the input last_spec_p to continue the query from where it left off.

Memory for the info_rec_p_p linke list is allocated by batch services, and the caller must return this memory by calling BES_free_info.

ERRORS:

BES_err_batch_not_open
BES_err_no_resources

*/

7.38. BES_logoff

error_typ

BES_logoff(ims_sess_id)

ASE_session_number_typ ims_sess_id; /* IN: session number */

(pg483.24)

/* Logoff terminates a Batch Entry Services session and invalidates a session on the server. No further calls can be made to Batch Entry Services until another BES_logon call is performed.

If any cache object was in the process of being created or modified and the create/modify is not complete when BES_logoff is called, the create/modify will be terminated and the cache object will be closed or deleted as is necessary to put the cache back into its original state.

If any batch was open when logoff is called, the batch will be left open and the client can continue using it by re-logging on and then using the same BES_batch_cap_typ record as was used previously.

*/

7.39. BES_logon

error_typ

BES_logon(service_name_p, leave_open, sess_id_p, timeout_p)

ASE_service_name_typ *service_name_p; /* IN: session number */

(pg486.28)

bool leave_open; /* IN: leave connection open */

(pg491.22)

ASE_session_number_typ *sess_id_p; /* OUT: session number */

(pg483.24)

+++ BES - Batch Entry Services +++

unsigned int *timeout_p; /* OUT: obsolete (may be NULL) */

/* Logon establishes a Batch Entry Service session and returns a number that identifies the session to the client. The number must be presented to subsequent calls to Batch Entry Services.

ERRORS:

 BES_err_no_permission
*/

/* #endif */

7.40. BES_logon_extended

error_typ

BES_logon_extended(service_name_p, leave_open, sess_id_p, timeout_p, level)
 ASE_service_name_typ *service_name_p; /* IN: session number */ (pg486.28)
 bool leave_open; /* IN: leave connection open */ (pg491.22)
 ASE_session_number_typ *sess_id_p; /* OUT: session number */ (pg483.24)
 unsigned int *timeout_p; /* OUT: obsolete (may be NULL) */
 short level;

/* BES_logon_extended combines BES_logon with BES_extended

*/

7.41. BES_modify_image_index

error_typ

BES_modify_image_index(ims_sess_id, batch_cap_p, image_id, ix_len, ix_val)
 ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
 BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)
 ASE_doc_id_typ image_id; /* IN: image id of image to change */ (pg483.37)
 unsigned short ix_len; /* IN: length of image index */
 char ix_val[]; /* IN: image index value */

/* BES_modify_image_index provides the caller with the ability to modify the index data associated with a specified image. The image need not be open.

ERRORS:

 BES_err_batch_not_open
 BES_err_batch_busy
 BES_err_image_not_found
 BES_err_no_image_index

+++ BES - Batch Entry Services +++

BES_err_invalid_image_index

*/

7.42. BES_move_doc

error_typ

BES_move_doc(ims_sess_id, src_batch_cap_p, dst_batch_cap_p, old_doc_id,
new_doc_id_p)

ASE_session_number_typ	ims_sess_id;	/* IN: session number */	(pg483.24)
BES_batch_cap_typ	*src_batch_cap_p;	/* IN: source batch capability */	(pg28.3)
BES_batch_cap_typ	*dst_batch_cap_p;	/* IN: destination batch capability */	(pg28.3)
ASE_doc_id_typ	old_doc_id;	/* IN: old document id */	(pg483.37)
ASE_doc_id_typ	*new_doc_id_p;	/* OUT: new document id returned */	(pg483.37)

/* BES_move_doc moves documents between batches. The restrictions on moving document between batches are as follows

1. Both batches must be opened in exclusive write mode.

2. Both batches can not be in commit, catalog or recommit phase.

If the source and destination batches have same document class, then index values will be transferred from old document to new document, and batch totalling will be performed on both source and destination batches, if batch_total flag is turned on.

If the source and destination batches have different document class, then index values will be dropped, and batch totalling will not be performed on the destination batch, if batch_total flag is turned on.

The document id's are batch relative. The values range from 1 to n. It is necessary to return the new document id, since it is probable that the document id's may overlap between batches.

Input :

ims_session_id - session number returned by logon
src_batch_cap_p - source batch capability
dst_batch_cap_p - destination batch capability
old_doc_id - old document id

Output :

new_doc_id_p - new document id

Errors returned :

- BES_err_batch_not_open
- BES_err_doc_not_found
- BES_err_invalid_move_phase
- BES_err_batch_tot_invalid_move

+++ BES - Batch Entry Services +++

- BES_err_move_doc_same_batch

*/

7.43. BES_move_image

error_typ

BES_move_image(ims_sess_id, batch_cap_p1, batch_cap_p2, image_id)

ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p1; /* IN: capability for source batch */ (pg28.3)
BES_batch_cap_typ *batch_cap_p2; /* IN: capability for destination */ (pg28.3)
ASE_doc_id_typ image_id; /* IN: image id of image to move */ (pg483.37)

/* BES_move_image moves a image from one batch to another. The image must not already be part of a document when this routine is called.

ERRORS:

BES_err_batch_not_open
BES_err_batch_busy
BES_err_image_not_found
BES_err_no_image_index
BES_err_invalid_image_index
BES_err_already_in_doc

*/

7.44. BES_open_batch

error_typ

BES_open_batch(ims_sess_id, batch_name, open_mode, batch_hdr_p, batch_cap_p)

ASE_session_number_typ ims_sess_id; /* IN: Session number */ (pg483.24)
char batch_name[]; /* IN: Name of batch */
BES_open_options_typ open_mode; /* IN: BES_OPEN_NORMAL, BES_OVERRIDE, (pg26.47)
or BES_OPEN_READONLY */
BES_hdr_desc_typ *batch_hdr_p; /* OUT: Header of batch opened */ (pg31.43)
BES_batch_cap_typ *batch_cap_p; /* OUT: Capability record used to (pg28.3)
access this batch with
subsequent calls. */

/* BES_open_batch provides the client with exclusive update privileges to a specified batch, and returns the batch header and the batch capabilities. The batch capabilities structure is passed to all subsequent calls which update the open batch.

The batch capabilities structure is a key which identifies the batch (by batch id) and the caller who has this batch open, and also authorizes only this caller to update the batch (if update access requested). This

+++ BES - Batch Entry Services +++

structure is passed in to all other entry points which require an open batch as a prerequisite to the call. This structure is saved on magnetic disk, and will be in effect even after the system is rebooted.

Inputting `open_mode` as `BES_OPEN_NORMAL` attempts to open and lock the batch so that the caller has exclusive access to the batch. Using `BES_OPEN_NORMAL` when the batch is already opened and locked by someone else will return the error `BES_err_batch_busy`.

Inputting `open_mode` as `BES_OVERRIDE` allows a client with sufficient privilege to override the open and locked status someone else has put on the batch, and open and lock the batch for the current caller. This parameter must be used if the workstation goes down while a batch is open so that the batch may be reopened and processing can continue. When `BES_OVERRIDE` is used, the other user who had the batch open/locked will abruptly receive `BES_err_batch_not_open` errors to all subsequent calls, because two clients will not be allowed to update the same batch at the same time. If `BES_OVERRIDE` is input when the batch is not opened, then the error `BES_err_batch_not_locked` is returned and the override attempt fails.

Passing `open_mode` as `BES_OPEN_READONLY` will allow a user to read the batch, but not update it. This value opens the batch but does not lock it, so another user will not be stopped from opening the batch using `BES_OPEN_NORMAL` for `open_mode`.

ERRORS:

`BES_err_batch_not_found`
`BES_err_batch_busy`
`BES_err_batch_not_locked`

*/

7.45. `BES_open_connection`

`error_typ`

`BES_open_connection(ims_sess_id)`

`ASE_session_number_typ ims_sess_id; /* IN: session number */`

(pg483.24)

`/* BES_open_connection opens a network connection. Connections should not be left open for long periods (i.e. minutes). */`

7.46. `BES_open_csum_image`

`error_typ`

`BES_open_csum_image(ims_sess_id, batch_cap_p, image_id, image_desc_p, csum_exists_p, csum_p)`

+++ BES - Batch Entry Services +++

```

ASE_session_number_typ  ims_sess_id;    /* IN:  session number */           (pg483.24)
BES_batch_cap_typ       *batch_cap_p;  /* IN:  batch capabilities */       (pg28.3)
ASE_doc_id_typ          image_id;      /* IN:  image id */                 (pg483.37)
BES_image_desc_typ      *image_desc_p; /* OUT: image description */         (pg32.41)
bool                    *csum_exists_p; /* OUT: check-sum present */        (pg491.22)
long                    *csum_p;      /* OUT: check-sum value if
                                     csum_exists_p is TRUE,
                                     0 otherwise */

```

/* BES_open_csum_image opens the specified image for read only and returns the associated image descriptor record, a boolean indicating whether the image has check-sum computed, and the check-sum value.

Note: Multiple BES_open_csum_image calls can be done on the same image, but BES_update_image cannot be done at the same time as BES_open_csum_image. Do not call BES_open_image and leave an image open for long periods of time if BES_update_image calls are being done concurrently.

Only one image can be open at a time. Note that the indication that the image is open is saved in the session (ims_sess_id), and subsequent calls which read and write images do not take an image id or description as an argument.

When calling BES_open_csum_image() through the WAL library, it uses one DBP buffer. When it is used in combination with DOC_batch_create, CSM_create_object, CSM_open_object, or CSM_open_csum_object, there could be a need to increase the number of DBP buffers on the IMS server. There is the potential for deadlocking the IMS if multiple applications are running at the same time and each program calls more than one of the entry points above and there are few DBP buffers on the IMS server.

ERRORS:

```

BES_err_batch_not_open
BES_err_image_not_found
BES_err_already_in_transaction

```

*/

7.47. BES_open_image

error_typ

```

BES_open_image(ims_sess_id, batch_cap_p, image_id, image_desc_p)
ASE_session_number_typ  ims_sess_id;    /* IN:  session number */           (pg483.24)
BES_batch_cap_typ       *batch_cap_p;  /* IN:  batch capabilities */       (pg28.3)
ASE_doc_id_typ          image_id;      /* IN:  image id */                 (pg483.37)

```

+++ BES - Batch Entry Services +++

BES_image_desc_typ *image_desc_p; /* OUT: image description */ (pg32.41)

/* BES_open_image opens the specified image for read only
and returns the associated image descriptor record.

Note: Multiple BES_open_image calls can be done on the same image, but
BES_update_image cannot be done at the same time as BES_open_image. Do
not call BES_open_image and leave an image open for long periods of time
if BES_update_image calls are being done concurrently.

Only one image can be open at a time. Note that the indication that
the image is open is saved in the session (ims_sess_id), and subsequent
calls which read and write images do not take an image id or description
as an argument.

When calling BES_open_image() through the WAL library, it
uses one DBP buffer. When it is used in combination with
DOC_batch_create, CSM_create_object, CSM_open_object, or
CSM_open_csum_object, there could be a need to increase
the number of DBP buffers on the IMS server. There is the
potential for deadlocking the IMS if multiple applications
are running at the same time and each program calls more than one of
the entry points above and there are few DBP buffers on
the IMS server.

ERRORS:

BES_err_batch_not_open
BES_err_image_not_found
BES_err_already_in_transaction

*/

7.48. BES_open_image_companion

error_typ

BES_open_image_companion (ims_sess_id, batch_cap_p, image_id, companion_id,
 companion_desc_p)

ASE_session_number_typ	ims_sess_id;	/* IN: session number */	(pg483.24)
BES_batch_cap_typ	*batch_cap_p;	/* IN: batch capabilities */	(pg28.3)
ASE_image_id_typ	image_id;	/* IN: image id */	(pg483.33)
ASE_page_number_typ	companion_id;	/* IN: companion id */	(pg483.41)
BES_companion_desc_typ	*companion_desc_p;	/* OUT: companion description */	(pg37.35)

/* BES_open_image_companion opens the specified companion object for read only
and returns the associated companion descriptor record.

ERRORS:

BES_err_batch_not_open

+++ BES - Batch Entry Services +++

BES_err_image_not_found
BES_err_already_in_transaction
BES_err_companion_not_found
BES_err_invalid_companion_id
BES_err_image_companion_invalid_parameter

*/

7.49. BES_put_info

error_typ

BES_put_info (ims_sess_id, batch_cap_p, num_to_put, info_rec_p)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)
unsigned short num_to_put; /* IN: #records to store */
BES_info_rec_typ *info_rec_p; /* IN: linked list of records */ (pg35.44)

/* BES_put_info provides the client with the ability to associate additional information with a batch record, a batch document record, a batch image record, a batch class index type record, or a batch document index value record.

Format of the additional information thus added is defined and interpreted by the client. Batch Services will simply treat this information as a sequence of bytes.

ERRORS:

BES_err_batch_not_open
BES_err_no_resources

*/

7.50. BES_put_object_data

error_typ

BES_put_object_data (ims_sess_id, batch_cap_p, num_to_put,
object_rec_p)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)
unsigned long num_to_put; /* IN: number of records to put */
BES_object_rec_typ *object_rec_p; /* IN: linked list of records (pg37.17)
* of BES_object_rec_typ */

/*

BES_put_object_data provides the client with the means to associate new additional information or update existing additional information with a batch record, a batch document record, a batch image record, or a batch phase. Users can define additional object types and associate

+++ BES - Batch Entry Services +++

information with user defined objects.

Multiple buffers may be associated with one batch object or user defined object by using sequence numbers.

Input:

ims_sess_id - session number returned by logon
batch_cap_p - batch capability
num_to_put - number of records to store\update
object_rec_p - a linked list of records of BES_object_rec_typ.

ERRORS:

BES_err_batch_not_open
BES_err_invalid_object_sequence_no
BES_err_invalid_object_type
BES_err_invalid_object_id
BES_err_object_data_too_large
BES_err_object_not_in_batch
BES_err_no_resources

*/

7.51. BES_query_index_dir

error_typ

BES_query_index_dir(ims_sess_id, batch_cap_p, num_indices_p, index_descs_p_p)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: capability */ (pg28.3)
long *num_indices_p; /* OUT: #indexes */
BES_ixdir_desc_typ **index_descs_p_p; /* OUT: index descriptions */ (pg32.24)

/* BES_query_index_dir provides the caller with the ability to retrieve the directory of indices associated with the specified batch.

Memory for index_descs_p_p is allocated by batch services, and must be returned by calling BES_return_index_dir.

ERRORS:

BES_err_batch_not_open
BES_err_ixdir_not_found

*/

7.52. BES_read_image

error_typ

BES_read_image(ims_sess_id, batch_cap_p, offset, num_bytes, bytes_read_p, buf_p)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)

+++ BES - Batch Entry Services +++

```
BES_batch_cap_typ      *batch_cap_p; /* IN: batch capabilities */           (pg28.3)
unsigned long          offset; /* IN: offset from start of image,
                           in bytes, to start reading. */
long                  num_bytes; /* IN: number of bytes to read */
long                  *bytes_read_p; /* OUT: actual number of bytes read */
any_ptr               buf_p; /* OUT: buffer to read data into. */ (pg491.24)
```

/* BES_read_image attempts to read num_bytes bytes from an image on which there is a current transaction. Reading begins at the specified offset. The actual number of bytes read will be less than the number of bytes requested if the offset plus num_bytes goes beyond the end of the image.

ERRORS:

```
BES_err_batch_not_open
BES_err_no_transaction
BES_err_invalid_batch_cap
```

*/

7.53. BES_read_image_companion

error_typ

```
BES_read_image_companion ( ims_sess_id, batch_cap_p, offset, num_bytes,
                           bytes_read_p, buf_p )
ASE_session_number_typ  ims_sess_id; /* IN: session number */           (pg483.24)
BES_batch_cap_typ      *batch_cap_p; /* IN: batch capabilities */           (pg28.3)
unsigned long          offset; /* IN: offset from start of companion,
                           in bytes, to start reading. */
unsigned long          num_bytes; /* IN: number of bytes to read */
unsigned long          *bytes_read_p; /* OUT: actual number of bytes read */
any_ptr               buf_p; /* OUT: buffer to read data into. */ (pg491.24)
```

/* BES_read_image_companion attempts to read num_bytes bytes from an companion object on which there is a current transaction. Reading begins at the specified offset. The actual number of bytes read will be less than the number of bytes requested if the offset plus num_bytes goes beyond the end of the image.

ERRORS:

```
BES_err_batch_not_open
BES_err_no_transaction
BES_err_invalid_batch_cap
BES_err_image_companion_invalid_parameter
```

*/

+++ BES - Batch Entry Services +++

7.54. BES_return_docs

```
error_typ
BES_return_docs(ims_sess_id, doc_list_p p)
  ASE_session_number_typ  ims_sess_id;      /* IN: session number */      (pg483.24)
  BES_doc_list_typ        **doc_list_p p;    /* IN: value returned by    (pg35.1)
                                     BES_find_documents */

/* BES_return_docs returns memory allocated by BES_find_docs.
   The "next_doc" fields in the BES_doc_list_typ structure must
   not be modified between the calls to BES_find_docs and
   BES_return_docs.
*/
```

7.55. BES_return_index_dir

```
error_typ
BES_return_index_dir(ims_sess_id, index_descs_p p)
  ASE_session_number_typ  ims_sess_id;      /* IN: session number */      (pg483.24)
  BES_ixdir_desc_typ      **index_descs_p p; /* IN: index descriptions */  (pg32.24)

/* BES_return_index_dir returns memory allocated by BES_query_index_dir. */
```

7.56. BES_sync_commit

```
error_typ
BES_sync_commit(ims_sess_id, batch_cap_p)
  ASE_session_number_typ  ims_sess_id;      /* IN: session number */      (pg483.24)
  BES_batch_cap_typ       *batch_cap_p;     /* IN: batch capabilities */  (pg28.3)

/* BES_sync_commit provides the client with the ability to
   synchronously commit a batch. In order for this routine to
   execute, the batch must be in a state which is acceptable
   for committal. See the description of BES_enqueue_batch for
   the requirements of batches to be committed.
```

If the batch is in an acceptable state, committal will be attempted.
If all documents are successfully committed, then the batch is deleted.
If any document fails committal, the batch remains open and will contain
the documents which failed committal.

The batch is placed in the IN_PROGRESS queue. This allows
committal to be automatically restarted by the daemon in
the event of a crash. The client will not be notified of
completion if the system crashes.

+++ BES - Batch Entry Services +++

This routine returns an error if the network is down or the cache is full. The client program may wish to retry these errors after displaying the error condition to the user.

ERRORS:

BES_err_batch_not_open
BES_err_phase_incomplete
BES_err_image_not_verified
BES_err_commit_batch_total
BES_err_ixval_not_found
BES_err_no_required_index
BES_err_index_not_verified
CSM_no_resources
Network errors

*/

7.57. BES_update_batch

error_typ

BES_update_batch(ims_sess_id, batch_dyn_hdr_p, batch_cap_p)

ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_dyn_hdr_desc_typ *batch_dyn_hdr_p; /* IN: new values for batch_hdr */ (pg31.38)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)

/* BES_update_batch provides the client with the ability to update fields in the dynamic batch header.

ERRORS:

BES_err_batch_not_open

*/

7.58. BES_update_doc

error_typ

BES_update_doc(ims_sess_id, batch_cap_p, doc_desc_p, page_list_ary,
index_values_ary)

ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)
BES_doc_desc_typ *doc_desc_p; /* IN: document description */ (pg34.30)
ASE_doc_id_typ page_list_ary[]; /* IN: pages in document */ (pg483.37)
BES_ixval_desc_typ index_values_ary[]; /* IN: indexes in document */ (pg33.37)

/* BES_update_doc provides the client with the ability to update a previously created document. The client can alter the page composition and/or the document indexing information.

+++ BES - Batch Entry Services +++

If `page_list_ary` is a NULL pointer, `BES_update_doc` will not update page composition. If `index_values_ary` is a NULL pointer, `BES_update_doc` will not alter the indexing information. To remove all indices, pass in a value for `index_values_ary` and set `doc_desc_p->num_indices` to zero.

The `index_values_ary` passed in is considered a complete list of indices, and any index value, that existed from previous `BES_create_doc`, or `BES_update_doc` commands will be removed. Therefore, to change a single, or few indices, all previous values must be read by the client and passed in via the `index_value_ary`, as well as the changed values. To remove an index value, read all index values, and create an array of just the index values to be retained, to pass as the `index_values_ary`. If a value is nulled, the index value is still considered to be present when the required indices are tested.

ERRORS:

`BES_err_batch_not_open`
`BES_err_doc_not_found`
`BES_err_no_resources`
`BES_err_bad_pages`
`BES_err_too_many_pages`
`BES_err_too_many_indices`

*/

7.59. `BES_update_image`

`error_typ`

`BES_update_image(ims_sess_id, batch_cap_p, image_desc_p)`

<code>ASE_session_number_typ</code>	<code>ims_sess_id;</code>	<code>/* IN: session number */</code>	(pg483.24)
<code>BES_batch_cap_typ</code>	<code>*batch_cap_p;</code>	<code>/* IN: batch capabilities */</code>	(pg28.3)
<code>BES_image_desc_typ</code>	<code>*image_desc_p;</code>	<code>/* IN: new image description */</code>	(pg32.41)

/* `BES_update_image` opens the image for reads and writes, and initiates a "transaction" on the image. All changes made under control of the transaction do not take effect until either `BES_close_image` or `BES_close_csum_image` is made. These changes include the new `*image_desc_p` passed in on this call, `BES_write_image` calls, and anything which requires an open image to execute.

If another client has this image open via `BES_read_image`, this routine will wait for up to 30 seconds for the `BES_read_image` caller to close the image. If 30 seconds wait is exceeded this routine returns an error.

+++ BES - Batch Entry Services +++

Only one image can be open at a time. Note that the indication that the image is open is saved in the session (`ims_sess_id`), and subsequent calls which read and write images do not take an image id or description as an argument.

When calling `BES_update_image()` through the WAL library, it uses one DBP buffer. When it is used in combination with `DOC_batch_create`, `CSM_create_object`, `CSM_open_object`, or `CSM_open_csum_object`, there could be a need to increase the number of DBP buffers on the IMS server. There is the potential for deadlocking the IMS if multiple applications are running at the same time and each program calls more than one of the entry points above and there are few DBP buffers on the IMS server.

In `*image_desc_p`, the fields are used as follows:

The following are input by the client:

`image_id`
`image_length`
`image_type`
`image_ver_stat`

The following are set to zero since other entry points set these values:

`index_len`

The following is not used:

`end_of_doc`

ERRORS:

`BES_err_batch_not_open`
`BES_err_image_not_found`
`BES_err_no_resources`
`BES_err_already_in_transaction`
`CSM_object_busy`

*/

7.60. `BES_update_image_companion`

`error_typ`

```
BES_update_image_companion ( ims_sess_id, batch_cap_p, companion_desc_p )
ASE_session_number_typ      ims_sess_id;          /* IN: session number */      (pg483.24)
BES_batch_cap_typ           *batch_cap_p;        /* IN: batch capabilities */  (pg28.3)
BES_companion_desc_typ      *companion_desc_p; /* IN: new companion
                                                                    description */      (pg37.35)
```

+++ BES - Batch Entry Services +++

/* BES_update_image_companion opens the companion_object for reads/writes, and initiates a "transaction" on the companion object. All changes made under control of the transaction do not take effect until after BES_close_image_companion call is made. These changes include the new *companion_desc_p passed in on this call, BES_write_image_companion calls, and anything which requires an open companion object to execute.

Only one companion object or image can be open at a time. Note that the indication that the companion object or image is open is saved in the session (ims_sess_id), and subsequent calls which read and write companion object do not take an companion id., image id or companion description as an argument.

ERRORS:

BES_err_batch_not_open
BES_err_image_not_found
BES_err_companion_not_found
BES_err_no_resources
BES_err_already_in_transaction
CSM_object_busy
BES_err_invalid_companion_id
BES_err_image_companion_invalid_parameter

*/

7.61. BES_update_index_total

error_typ

BES_update_index_total(ims_sess_id, batch_cap_p, index_id, value)
ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)
BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)
long index_id; /* IN: index id of index */
FP_number value; /* IN: expected batch total for (pg211.20)
specified index */

/* BES_update_index_total_total provides the client with the ability to update the batch total for the specified index.

ERRORS:

BES_err_batch_not_open
BES_err_ixdir_not_found

*/

7.62. BES_verify_image

error_typ

+++ BES - Batch Entry Services +++

```
BES_verify_image(ims_sess_id, batch_cap_p, image_id, ver_status, image_desc_p)
  ASE_session_number_typ  ims_sess_id;    /* IN:  session number */      (pg483.24)
  BES_batch_cap_typ       *batch_cap_p;   /* IN:  batch capabilities */  (pg28.3)
  ASE_doc_id_typ          image_id;        /* IN:  image id of image verified */ (pg483.37)
  BES_verify_stat_typ     ver_status;      /* IN:  verification status to put on (pg26.22)
                                         image */
  BES_image_desc_typ      *image_desc_p;  /* OUT: new image description */  (pg32.41)
```

/* BES_verify_image updates the image verification status in the image descriptor record associated with the specified image. This routine just updates a status variable, no inspection of the image is actually done in this routine.

No transaction need be in progress on the image when this routine is called. This update takes effect immediately (no BES_close_image call need be done).

If the verify_images flag the batch header is TRUE, then all images in the batch must have a verification status of BES_GOOD or else a commit attempt will be rejected with an error.

ERRORS:

```
    BES_err_batch_not_open
    BES_err_image_not_found
*/
```

7.63. BES_write_image

```
error_typ
BES_write_image(ims_sess_id, batch_cap_p, offset, num_bytes, data_p)
  ASE_session_number_typ  ims_sess_id;    /* IN:  session number */      (pg483.24)
  BES_batch_cap_typ       *batch_cap_p;   /* IN:  batch capability */    (pg28.3)
  unsigned long           offset;         /* IN:  offset from start of image,
                                         in bytes, to start write at */
  long                    num_bytes;      /* IN:  number of bytes to write */
  any_ptr                 data_p;         /* IN:  buffer to write */      (pg491.24)
```

/* BES_write_image attempts to write num_bytes bytes to an image on which there is a current transaction. BES_write_image starts writing at the specified offset.

BES_write_image can be called only on BES_create_image and BES_update_image transactions. An error will be returned if BES_write_image is attempted on a BES_open_image transaction.

+++ BES - Batch Entry Services +++

ERRORS:

BES_err_batch_not_open
BES_err_no_transaction
BES_err_invalid_batch_cap

*/

7.64. BES_write_image_companion

error_typ

BES_write_image_companion (ims_sess_id, batch_cap_p, offset, num_bytes,
data_p)

ASE_session_number_typ ims_sess_id; /* IN: session number */ (pg483.24)

BES_batch_cap_typ *batch_cap_p; /* IN: batch capability */ (pg28.3)

unsigned long offset; /* IN: offset from start of companion,
in bytes, to start write at */

unsigned long num_bytes; /* IN: number of bytes to write */

any_ptr data_p; /* IN: buffer to write */ (pg491.24)

/* BES_write_image_companion attempts to write num_bytes bytes to an companion
object on which there is a current transaction. BES_write_image_companion
starts writing at the specified offset.

BES_write_image_companion can be called only on BES_create_image_companion
and BES_update_image_companion transactions. An error will be returned
if BES_write_image_companion is attempted on a BES_open_image_companion
transaction.

ERRORS:

BES_err_batch_not_open
BES_err_no_transaction
BES_err_invalid_batch_cap
BES_err_image_companion_invalid_parameter

*/

8. CKS DECLARATIONS

```
/******
```

CKS abstract interface

```
*****/
```

```
#ifndef CKS_defs
#define CKS_defs
#ifndef AS_externals_h
#include <AS_externals.h>
#endif
#endif
```

9. CKS SUBROUTINES

9.1. CKS_compute_csum

```
long
CKS_compute_csum (ptr, numbytes, partial_csum)
    long*      ptr;          /* IN: memory buffer */
    long       numbytes;    /* IN: length of data to checksum */
    long       partial_csum; /* IN: checksum computed so far (0 if
                             first piece of data to be
                             checksummed) */
```

/* Computes a checksum on the data indicated. If all data of an object to be checksummed is in one memory buffer, use:

```
"checksum = CKS_compute_csum (ptr, numbytes, 0);"
```

If buffer is in multiple pieces, pass in the checksum of the nth piece when computing the checksum of piece n+1. For example, if data is in two buffers, use:

```
"csum1 = CKS_compute_csum (ptr1, numbytes1, 0);"
"csum2 = CKS_compute_csum (ptr2, numbytes2, csum1);"
```

where "csum2" is the checksum of both ptr1 and ptr2.

+++ CKS - Checksum Services +++

The checksum computed is in a format native to the local machine, and should be serialized across the network as a long word. If the local machine is not bigender format, then this checksum should be long swapped before being stored on media which must be in bigender format.

Restrictions:

When checksumming images, "ptr" must be on a 4 byte boundary, and if `CKS_compute_csum` is called multiple times to compute one checksum, all calls prior to the last one must have a length which is a multiple of 4.

For applications other than checksumming images, "ptr" does not need to be on a 4 byte boundary, but "ptr modulo 4" must be the same each time `CKS_compute_csum` is called. */

10. CSM DECLARATIONS

/*

This is the interface description for the cache services manager abstract (CSM). This abstract supplies transparently remotable calls used to create, open, close, delete, read, write, enumerate, copy, move and retrieve or update attributes for object contained in a cache.

To use the service, a client logs onto a cache and is returned a session handle which he uses in all subsequent calls to the cache services abstract. If the client creates or opens an object, he is returned an object handle which must be passed in when calls are made to close, read or write to an object.

The following interface definitions describe the data types, data structures and interface definitions used in the cache services abstract.

*/

/*

File: CSM.defs. Cache services manager data type
declaration include file

*/

/*\$I <SEC.defs>*/

#ifndef CSM_defs /* ensure we only use this header once */
#define CSM_defs

#ifndef ErrEncode_h
#include <ErrEncode.h>
#endif

#ifndef AS_externals_h
#include <AS_externals.h>
#endif

#ifndef SEC_defs
#include <SEC.defs>
#endif

/* Misc defines */

#define CSM_MAX_TRAN_CACHES 8
#ifdef SYS_V

+++ CSM - Cache Service Manager +++

```
#define CSM_MAX_PARTITIONS      255 /* partition(8bit field):0 not used */
#else
#define CSM_MAX_PARTITIONS      64
#endif
#define CSM_ALL_SSNS            0
#define CSM_ALL_DOCS            0
#define CSM_ALL_PAGES           ASE_INVALID_PAGE_NUMBER - 1
#define CSM_CLIENT_ATTR_ELMNTS  8
#define CSM_INVALID_CACHE_ID    -1
#define CSM_ALL_CACHES          CSM_INVALID_CACHE_ID
#define CSM_TEMP_ID_MIN         4160000000
#define CSM_TEMP_ID_MAX         4294000000
#define CSM_SECT_SIZE           1024
#define CSM_MASK                 (CSM_SECT_SIZE-1)
#define CSM_ROUND(x)            (((unsigned int)x + CSM_MASK) & ~CSM_MASK)
```

/* CSM object access modes. */

```
typedef unsigned short          CSM_object_access_mode;
#define CSM_OBJECT_READ         SEC_WANT_READ
#define CSM_OBJECT_WRITE        SEC_WANT_WRITE
#define CSM_READ_NOUPDATE       0x0008
```

/* Object age duration constants.

The system configuration files defines a prefetch duration, migrate duration, and a refresh duration. These durations are the number of seconds an object will remain in the cache before they are eligible to be deleted by the cache daemon process. The values below allow a caller to create an object which has one of the three predefined durations, or which will never be deleted from the cache by the daemon (CSM_LOCKED_DURATION). Note that CSM_CURRENT_DURATION is only used by a modify call, and it indicates to not change the existing duration on the object.

```
*/
#define CSM_LOCKED_DURATION      0 /* Object is locked in the cache*/
#define CSM_PREFETCH_DURATION    1 /* For objects prefetch to cache*/
#define CSM_MIGRATE_DURATION     2 /* For objects migrated to cache*/
#define CSM_REFRESH_DURATION     3 /* For objects already read */
#define CSM_CURRENT_DURATION     4 /* Don't change current duration*/
```

/* The handle to be used for list logons */

```
typedef char*                   CSM_cache_list_typ;
```

```
typedef struct CSM_bucket_struct {
    unsigned        lower;
```

+++ CSM - Cache Service Manager +++

```
    unsigned        upper;
    unsigned        objects;
    unsigned        sectors;
} CSM_bucket_typ;

/* Cache access modes. */

typedef unsigned short                CSM_cache_access_mode;
#define CSM_CACHE_READ                0x0001
#define CSM_CACHE_WRITE               0x0002
#define CSM_CACHE_DELETE              0x0004
#define CSM_CACHE_READWRITE          (CSM_CACHE_READ | CSM_CACHE_WRITE)
#define CSM_CACHE_READDELETE         (CSM_CACHE_READ | CSM_CACHE_DELETE)
#define CSM_CACHE_WRITEDELETE        (CSM_CACHE_WRITE | CSM_CACHE_DELETE)
#define CSM_CACHE_READWRITEDELETE    (CSM_CACHE_READ | CSM_CACHE_WRITE |
                                        CSM_CACHE_DELETE)

/* Define cache services type declarations. */

typedef enum {CSM_NOT_FULL, CSM_NEAR_FULL, CSM_FULL} CSM_cache_fullness_typ;
typedef enum {CSMS_TRACE, CSML_TRACE} CSM_trace_typ;

typedef short                CSM_cache_id_typ;
typedef unsigned            CSM_object_handle_typ;
typedef unsigned            CSM_session_handle_typ;
typedef char                CSM_cache_name_typ[ASE_MAX_OBJECT_LEN + 1];
typedef unsigned            CSM_io_handle_typ;

/* CSM cache attributes data structure. */

typedef struct CSM_cache_attr_struct {
    CSM_cache_name_typ    name;                                (pg93.25)
    CSM_cache_id_typ     cache_id;                            (pg93.22)
    unsigned              min_sector_size;                    /* min cache size in sects */
    unsigned              max_sector_size;                    /* max cache size in sects */
    unsigned              sectors_avail;                       /* # sectors available */
    unsigned              sectors_in_use;                     /* # sectors in use */
    unsigned              locked_sectors;                     /* # locked sectors */
    unsigned              unlocked_sectors;                   /* # unlocked-inuse or avail*/
    unsigned              num_objects;                        /* current # objects */
    unsigned              locked_objects;                     /* # locked objects */
    unsigned              unlocked_objects;                   /* # unlocked objects */
    bool                  refcnts;                            /* TRUE => refcnts enabled */ (pg491.22)
    bool                  ageable;                            /* TRUE => objects can age */ (pg491.22)
    SEC_access_restrictions security;                         /* access for this cache */ (pg385.2)
    CSM_cache_fullness_typ fullness;                          /* percentage of fullness */ (pg93.19)
} CSM_cache_attr_typ;
```

+++ CSM - Cache Service Manager +++

```

typedef struct CSM_cache_config_struct {
    CSM_cache_id_typ      cache_id;           (pg93.22)
    ASE_ssn_typ           ssn;               (pg483.47)
    long                  min_cache_size;
    long                  max_cache_size;
    bool                  refcnts;           (pg491.22)
    bool                  ageable;          (pg491.22)
} CSM_cache_config_typ;

/* CSM object description data structure. */

typedef struct CSM_object_desc_struct {
    ASE_ssn_typ           ssn;      /* system serial number */ (pg483.47)
    ASE_doc_id_typ        id;       /* document id */ (pg483.37)
    ASE_page_number_typ   page;     /* document page */ (pg483.41)
} CSM_object_desc_typ;

/* CSM object attributes data structure. */

typedef struct CSM_object_attr_struct {
    unsigned              max_length;      /* maximum length of object */
    unsigned              cur_length;      /* current length of object */
    ASE_time_typ          created;         /* time object was created */ (pg489.31)
    ASE_time_typ          last_read;       /* time last opened for read */ (pg489.31)

    /* This field is a small positive integer on the input side
       (i.e. create_object and modify_object_attributes; see object
       durations such as CSM_LOCKED_DURATION, CSM_REFRESH_DURATION, etc.),
       and is a date-time on the output side (i.e. open_object and
       get_object_attributes). */

    ASE_time_typ          duration;        (pg489.31)
    SEC_access_restrictions security;     (pg385.2)
    unsigned short        client_attr[CSM_CLIENT_ATTR_ELMNTS];
} CSM_object_attr_typ;

/* Disk Address structure. */

typedef struct CSM_disk_addr_struct {
#ifdef BIGENDER
    unsigned              partition : 8;
    unsigned              sector    : 24;
#else
    unsigned              sector    : 24;
    unsigned              partition : 8;
#endif
} CSM_disk_addr_typ;

```

+++ CSM - Cache Service Manager +++

```
typedef struct CSM_object_breakup_struct {
    unsigned long sector_offset;          /* Sector offset */
                                           /* in the carrier cache object */
    CSM_object_attr_typ attr;             (pg94.35)
    CSM_object_desc_typ object;          (pg94.16)
    long csum;                            /* Checksum value if required */
    bool csum_exists;                     /* Checksum exists for this object? */ (pg491.22)
} CSM_object_breakup_typ;

/* Courier error number definitions.    */

#define CSM_XIN_PROTOCOL      0  /* Defined in this Courier protocol */
#define CSM_XNOT_IN_PROTOCOL  1  /* Errors not defined there */

#define CSM_ZOTHER_ERROR     12

#define csm_err(x)          err_encode(err_CSM,CSM_XNOT_IN_PROTOCOL,(x))
#define csm_pro_err(x)     err_encode(err_CSM,CSM_XIN_PROTOCOL,(x))

/*$D err_CSM */

#define CSM_bad_version          csm_err(0)
/*$M Bad abstract link version when calling CSM. */

#define CSM_already_exists      csm_pro_err(1)
/*$M CSM: Attempt to create a CSM object which already exists */

#define CSM_invalid_object_handle csm_pro_err(3)
/*$M An invalid object handle has been encountered. */

#define CSM_invalid_session_handle csm_pro_err(4)
/*$M CSM given invalid session handle; session probably timed out. */

#define CSM_IO_error            csm_pro_err(5)
/*$M CSM_read, CSM_write: An IO error was encountered. */

#define CSM_no_permission       csm_pro_err(7)
/*$M Security attributes prevent access to this object. */

#define CSM_no_resources        csm_pro_err(8)
/*$M Magnetic disk cache is full.
Wait and retry the operation. Image Services will normally free up cache space
by writing document to optical disk, MSAR, etc. If this problem persists, check
the elog for error messages and check the Storage Library Control window for RSVPs.
Changing cache setting via fn_edit may also help with this problem. */

#define CSM_no_such_cache       csm_pro_err(9)
/*$M Unknown cache_id. */
```

+++ CSM - Cache Service Manager +++

```
#define CSM_no_such_object          csm_pro_err(10)
/*$M The specified object does not exist in the current cache. */

#define CSM_object_busy            csm_pro_err(11)
/*$M The object is busy (inuse by others). Access not allowed at this time.*/

#define CSM_other_error            csm_pro_err(CSM_ZOTHER_ERROR)
/*$M CSM error: Catchall for other errors encountered inside CSM. */

#define CSM_out_of_range           csm_pro_err(13)
/*$M Attempt to read/write beyond end of object. */

#define CSM_invalid_name_type      csm_err(16)
/*$M Invalid value for 'name_type' argument of logon. */

#define CSM_nil_data_pointer        csm_err(17)
/*$M A nil data pointer was passed. */

#define CSM_nil_bytes_trans_pointer csm_err(18)
/*$M A nil data transfer count pointer was passed. */

#define CSM_not_a_legal_cache      csm_err(20)
/*$M Attempt to use a partition which is not a cache services partition. */

#define CSM_illegal_disk_partition csm_err(21)
/*$M An illegal disk partition has been encountered in cache services. */

#define CSM_invalid_mode           csm_err(22)
/*$M Attempt to open an object or cache with an invalid mode. */

#define CSM_invalid_object_desc     csm_err(23)
/*$M An invalid object descriptor was passed in for the desired operation. */

#define CSM_tools_unavailable       csm_err(24)
/*$M You may not use this tool now, another client is currently using CSM. */

#define CSM_mult_open              csm_err(25)
/*$M Attempt to open same object twice with write access */

#define CSM_invalid_close          csm_err(26)
/*$M Attempt to close object when external MKF transaction was aborted.
Client called CSM_begin_transaction, CSM_open_object, then transaction
aborted, then client called CSM_close_object(update=TRUE). This sequence
is illegal. */

#define CSM_not_available          csm_err(27)
/*$M The CSM tools are being run, clients to CSM are locked out. */
```

+++ CSM - Cache Service Manager +++

```
#define CSM_no_such_known_cache          csm_err(29)
/*$M A cache service name was looked up when that name is not on this server. */

#define CSM_fatal_err                   csm_err(30)
/*$M A fatal, unrecoverable CSM error has been detected. */

#define CSM_invalid_cache_handle        csm_err(31)
/*$M An invalid cache handle has been encountered. */

#define CSM_invalid_object_size         csm_err(32)
/*$M Probably an attempt to shrink an object below it's current size. */

#define CSM_invalid_bulk_data_source    csm_err(33)
/*$M An invalid bulk data source has been specified. */

#define CSM_invalid_bulk_data_sink      csm_err(34)
/*$M An invalid bulk data sink has been specified. */

#define CSM_invalid_age_duration        csm_err(35)
/*$M The age duration specified is unknown to CSM. */

#define CSM_invalid_maxlength           csm_err(36)
/*$M Invalid 'max_length' field in object attribute during create object */

#define CSM_invalid_relop               csm_err(37)
/*$M Invalid relational operation used to find objects in cache */

#define CSM_invalid_wildcard            csm_err(38)
/*$M Invalid wildcards used in find objects.
If the ssn is CSM_ALL_SSNS, object id must be CSM_ALL_DOCS.
If the object id is CSM_ALL_DOCS, the page must be CSM_ALL_PAGES. */

#define CSM_nil_cursor                  csm_err(39)
/*$M Nil cursor detected (CSM program error). */

#define CSM_no_ageable_docs             csm_pro_err(40)
/*$M Cannot put ageable document into this cache.
An attempt was made to put an ageable document into a cache, but
this cache has been configured to not allow ageable documents. */

#define CSM_refcnt_overflow             csm_err(41)
/*$M The reference count on an object has exceeded the maximum of 65534. */

#define CSM_refcnt_incremented          csm_pro_err(42)
/*$M Reference count incremented due to attempt to create duplicate */

#define CSM_invalid_sas_handle          csm_err(43)
```

+++ CSM - Cache Service Manager +++

```
/*$M Cannot change sessions on one connection when objects are open. */

#define CSM_logon_when_connect_open          csm_err(44)
/*$M Attempt to logon when connection open */

#define CSM_undefined_procedure              csm_err(45)
/*$M Attempt to call an undefined CSM procedure */

#define CSM_no_temp_ids                      csm_err(46)
/*$M No more temporary object ids. */

#define CSM_bulk_read_err                    csm_err(47)
/*$M CSM program error: Bytes read doesn't match #bytes transferred. */

#define CSM_err_internal_rpc_error           csm_err(48)
/*$M Internal rpc error occurred in CSM. */

#define CSM_err_not_debugging                csm_err(49)
/*$M Debugging operation requested from non-debugging CSM. */

#define CSM_no_refcnts                       csm_pro_err(50)
/*$M Reference counts not enabled on this cache */

#define CSM_err_cannot_find_named_service    csm_err(51)
/*$M Cannot find the named CSM service in the Clearinghouse. */

#define CSM_logon_access_mode                csm_err(52)
/*$M 'Mode' on cache logon does not allow this operation */

#define CSM_open_access_mode                 csm_err(53)
/*$M 'Mode' on object open does not allow this operation */

#define CSM_not_implemented                  csm_err(54)
/*$M This function not implemented. */

#define CSM_temp_busy                        csm_err(55)
/*$M Object temporarily busy.
Note: A client of CSM should never see this error. */

#define CSM_invalid_object                   csm_err(56)
/*$M Invalid object record in csm_used_space table.
The record in the csm_used_space table is inconsistent. */

#define CSM_invalid_io_handle                csm_err(57)
/*$M IO handle is invalid.
The io handle is invalid if an attempt is made to do asynchronous IO
on a remote cache, if memory runs out, or if a null handle is passed
in for any other reason. */
```


+++ CSM - Cache Service Manager +++

```
#define CSM_invalid_checksum          csm_err(58)
/*$M Invalid checksum detected by Cache Services */

#define CSM_ilgl_csum_update          csm_err(59)
/*$M This update not allowed on a object with a checksum.
You cannot write to or modify the length of an object with a checksum.
This action combined with a system crash could result in an object
in the cache being left with an invalid checksum. This is a system
programming error. */

#define CSM_config_err                csm_err(60)
/*$M Invalid cache configuration detected */

#define CSM_exim_foreground_only     csm_err(61)
/*$M CSM_exim tool only allowed to run in foreground, when using tape media. */

#define CSM_tool_update_info         csm_err(62)
/*$M CSM_tool: update completed successfully (INFORMATION ONLY) */

#define CSM_trace_info                csm_err(63)
/*$M Internal trace output (INFORMATION ONLY) */

#define CSM_no_local_tape             csm_err(64)
/*$M No valid local tape devices found */

#define CSM_overflow_breakup_object   csm_err(65)
/*$M One of the breakup sector offset accesses beyond the carrier boundry */

#define CSM_overlap_breakup_object    csm_err(66)
/*$M The breakup objects overlap */

#define CSM_attr_bad_breakup_object   csm_err(67)
/*$M The breakup attr field cur_length > max_length */

#define CSM_no_resources_in_target_cache CSM_no_resources
#define CSM_object_exists_in_target_cache CSM_already_exists

#define CSM_MAGIC                     0x50434D00
#define CSM_MAGIC1                    0x50434E00

#endif /* CSM_defs */
```

11. CSM SUBROUTINES

11.1. CSM_breakup_object

error_typ

```
CSM_breakup_object (handle, oh_p, num_breakup_objects, breakup_object_p,
                    allow_duplicates)
ASE_session_number_typ  handle;          /* IN:  cache handle */          (pg483.24)
CSM_object_handle_typ*  oh_p;           /* IN/OUT: object handle for   (pg93.23)
                                         the carrier object */
unsigned long num_breakup_objects;       /* IN: number of
                                         breakup_object_p */
CSM_object_breakup_typ  *breakup_object_p; /* IN: an array of objects to (pg95.8)
                                         be created/extracted from
                                         the carrier object */
bool allow_duplicates;                  /* IN: allow duplicate        (pg491.22)
                                         breakup_objects. If an
                                         object already exist with
                                         the same object ID and
                                         allow_duplicates is set to
                                         true then skip that object.
                                         */
```

/* The CSM_breakup_object subroutine will take carrier object and break it into a specified number (using the num_breakup_objects parameter) of new objects. It will delete carrier object and free any unused space. In addition it will update the CSM statistics to reflect these changes. This will all be done in a single transaction.

Parameters:

handle is a session handle returned from CSM_logon().

oh_p must be a valid object handle returned from CSM_create_object(), or other similar entry points.

num_breakup_objects specifies the number of cache objects that this cache object will be broken up into.

breakup_object_p is a pointer to an array of "num_breakup_objects" CSM_object_breakup_typ data structures, each of which specifies characteristics for a new cache object, e.g. where the object starts in the current cache object, how long it is, the cache object id and attributes for the new cache object. The data structure is defined as:

```
typedef struct CSM_object_breakup_struct {
    unsigned long sector_offset;          * Sector starting offset *
                                         * in the carrier cache object *
    CSM_object_attr_typ attr;           * As per CSM_create_object() *
```

+++ CSM - Cache Service Manager +++

```
    CSM_object_desc_typ object; * As per CSM_create_object() *
    bool csum_exists;          * Checksum exists for this object? *
    long csum;                 * Checksum value if required *
} CSM_object_breakup_typ;
```

allow_duplicates, if TRUE, indicates that the existence of a cache object specified by a CSM_object_breakup_typ data structure will not result in failure of this entry point, instead, this entry point will free the space in the carrier object associated with the duplicate cache object (the existing object is not deleted to so as not to have to deal with the case where that object is in use).

An example of sequence of calls leading up to the use of CSM_breakup_object:

```
obj_desc.id = ASE_INVALID_DOC_ID; * Use a temporary object *

* Create a large "carrier" cache object *
CSM_create_object(handle, &obj_attr, &obj_desc, &oh);

* Write multiple pages of data to the cache object in one I/O *
CSM_write_object(handle, oh, ...);

* Breakup the cache object into individual per page cache objects *
CSM_breakup_object(handle, &oh, ...);
```

CSM_breakup_object() fragments the current cache object into multiple per page cache objects by inserting multiple rows into the csm_used_space table and deleting the row for the current object, all within a single MKF transaction. Thus from the perspective of the "carrier" cache object, a successful return will have deleted the carrier object and nulled the *oh_p parameter.

Note: The breakup_object_p array does not need to be sorted. The carrier object does not need to be packed. Any unused space will be freed. The following consistency checks will be done on the breakup_object array:

- 1) Check to see that the breakup_objects don't overflow the carrier object
- 2) Check to make sure that the breakup_objects cur_length does not exceed the max_length
- 3) Check for overlapping breakup_objects

If any error is encountered, the carrier object will remain intact with no new object created since all the updates are done in a single transaction. In addition, the cache statistics will not be updated.

+++ CSM - Cache Service Manager +++

ERRORS: CSM_object_busy
CSM_no_such_object
CSM_overflow_breakup_object
CSM_attr_bad_breakup_object
CSM_overlap_breakup_object
CSM_invalid_object_handle
CSM_already_exists

*/

11.2. CSM_chg_refcnt

error_typ

CSM_chg_refcnt (handle, object_p, addit, cnt, new_cnt_p)

ASE_session_number_typ	handle;	/* IN: cache handle */	(pg483.24)
CSM_object_desc_typ*	object_p;	/* IN: cache object */	(pg94.16)
bool	addit;	/* IN: TRUE=>cnt added to object reference count, FALSE=>cnt is stored as the reference count */	(pg491.22)
long	cnt;	/* IN: Count added or moved into reference count of object. "cnt" may be positive, negative, or zero, but the resultant reference count of the object can't be less than 1 or greater than 65535 */	
long*	new_cnt_p;	/* OUT: Value of reference count in object after being set. */	

/* Sets/gets/increments/decrements the reference of an object,
provided the object exists and reference counts are enabled
on the cache.

ERRORS: CSM_no_such_object
CSM_no_refcnts

If both the object does not exist, and the cache does not allow
reference counts, a "CSM_no_such_object" error is returned. */

11.3. CSM_close_connection

error_typ

CSM_close_connection(handle)

ASE_session_number_typ	handle;	/* IN: cache handle */	(pg483.24)
------------------------	---------	------------------------	------------

+++ CSM - Cache Service Manager +++

/* Closes a network connection to a cache service */

11.4. CSM_close_csum_object

error_typ

```
CSM_close_csum_object(handle, oh_p, update, csum_exists, csum)
  ASE_session_number_typ  handle;      /* IN:  cache handle */      (pg483.24)
  CSM_object_handle_typ*  oh_p;        /* IN:  object handle */      (pg93.23)
  bool                    update;      /* IN:  if want to update db rec */ (pg491.22)
  bool                    csum_exists; /* IN:  if checksum exists */    (pg491.22)
  long                    csum;        /* IN:  checksum on object */
```

/* Closes the object, sets the checksum if csum_exists=TRUE, and zeroes the handle in *oh_p. Just like CSM_close_object except that it allows setting a checksum on the object.

Specifying "update"=FALSE will prevent the attributes (not data) of the object from being updated, which are "cur_length", "last_read", "duration", "security", and "client_attr". Therefore calls made to CSM_write_object or CSM_modify_object_attributes made while this object was open will not change the values of these attributes.

ERRORS: none other than network errors or system malfunctions. */

11.5. CSM_close_delete_object

error_typ

```
CSM_close_delete_object (handle, oh_p)
  ASE_session_number_typ  handle; /* IN:  cache handle */      (pg483.24)
  CSM_object_handle_typ  *oh_p; /* IN/OUT: object handle (zeroed out (pg93.23)
                                upon return */
```

/* Closes and deletes the object, and zeroes the handle in *oh_p. This routine may only be called if the object was opened via the CSM_create_object routine.

ERRORS: CSM_open_access_mode

*/

11.6. CSM_close_object

error_typ

```
CSM_close_object(handle, oh_p, update)
  ASE_session_number_typ  handle; /* IN:  cache handle */      (pg483.24)
```

+++ CSM - Cache Service Manager +++

```
CSM_object_handle_typ*    oh_p;    /* IN/OUT: object handle (zeroed out      (pg93.23)
                               upon return) */
bool                      update; /* IN:  update database record */      (pg491.22)
```

/* Closes the object and zeroes the handle in *oh_p (open object handles are never zero).

Specifying "update"=FALSE will prevent the attributes (not data) of the object from being updated, which are "cur_length", "last_read", "duration", "security", and "client_attr". Therefore calls made to CSM_write_object or CSM_modify_object_attributes made while this object was open will not change the values of these attributes.

CAUTION: Under a no-checksum environment, a CSM_close_object with no update may cause a modified object to not be backed up under the following scenario: CSM_open_object with write access, CSM_write_object, and CSM_close_object with no update. It is the caller's responsibility to be able to recover if the updates made to the object should disappear.

ERRORS: none other than network errors or system malfunctions.

*/

11.7. CSM_copy_object

error_typ

```
CSM_copy_object(handle, target_handle, target_service_p, object_p, new_object_p)
  ASE_session_number_typ  handle;    /* IN:  cache handle */      (pg483.24)
  ASE_session_number_typ  target_handle; /* IN:  destination cache      (pg483.24)
                               handle or NULL */
  ASE_service_name_typ*   target_service_p; /* IN:  destination cache name (pg486.28)
                               (must be specified) */
  CSM_object_desc_typ*    object_p;    /* IN:  object to copy */    (pg94.16)
  CSM_object_desc_typ*    new_object_p; /* IN/OUT: new object name */ (pg94.16)
```

/* Copies an object. The destination object will be created with duration of CSM_LOCKED_DURATION.

Setting new_object_p->id = ASE_INVALID_DOC_ID will cause the service to generate a temporary object descriptor for the target object. This temp object descriptor will be returned to the client.

If *object_p refers to a temporary object descriptor, *new_object_p must not be the same as *object_p.

If the new object already exists and reference counts are enabled, the reference count in the destination object will be incremented, and

+++ CSM - Cache Service Manager +++

no error will be returned.

ERRORS: CSM_no_permission
CSM_already_exists
CSM_object_busy
CSM_invalid_checksum
CSM_invalid_object_desc

*/

11.8. CSM_create_object

error_typ

```
CSM_create_object(handle, attr_p, object_p, oh_p)
  ASE_session_number_typ handle;      /* IN: cache handle */           (pg483.24)
  CSM_object_attr_typ* attr_p;        /* IN: attributes of object being
                                     created, but only:
                                     "attr_p->max_length",
                                     "attr_p->duration",
                                     "attr_p->security", and
                                     "attr_p->client_attr" are
                                     used from this structure.
                                     All other attributes in
                                     this structure are ignored
                                     because appropriate defaults
                                     are used. */
  CSM_object_desc_typ* object_p;      /* IN/OUT: object ssn/id/page. If
                                     object_p->id equals
                                     ASE_INVALID_DOC_ID, then a
                                     unique temporary object id
                                     will be assigned and returned
                                     in this structure, with
                                     object_p->page always = 1.
                                     Note that if a temporary
                                     object is deleted, the
                                     temporary object id may be
                                     reused. */           (pg94.16)
  CSM_object_handle_typ* oh_p;        /* OUT: The handle used to access this
                                     object on subsequent calls. */   (pg93.23)

/* Creates an object of the specified size, and leaves this object
open in CSM_OBJECT_WRITE mode (exclusive access). This object may be
written using the handle in *oh_p.
```

After the object is created and prior to it being updated on close, the object is marked as being incomplete. If the system gets rebooted, all incomplete objects are removed from the cache, therefore the application does not need special logic to detect partially written objects.

+++ CSM - Cache Service Manager +++

The object is visible prior to the close and can be found with `CSM_get_object_attributes` or `CSM_find_objects`, and will cause a `CSM_already_exists` or `CSM_refcnt_incremented` error on a `CSM_create_object` call which attempts to create a duplicate. `CSM_open_object` calls done between the create and the close will block until a timeout has expired or the close is completed.

If the cache supports reference counts, each object will be created initially with a reference count of 1. Subsequent calls to this routine when the object already exists will increment the reference count by 1 and a `CSM_refcnt_incremented` error will be returned.

When calling `CSM_create_object()` through the WAL library, it uses one DBP buffer. When it is used in combination with `DOC_batch_create`, `CSM_create_object`, `CSM_open_object`, `CSM_open_csum_object`, `BES_create_image`, `BES_open_image` or `BES_update_image`, there could be need to increase the number of DBP buffers on the IMS server. There is the potential for deadlocking the IMS if multiple applications are running at the same time and each program calls more than one of the entry points above and there are few DBP buffers on the IMS server.

ERRORS: `CSM_refcnt_incremented`
`CSM_already_exists`
`CSM_no_resources`
`CSM_no_permission`
`CSM_logon_access_mode`
`CSM_no_ageable_docs`
`CSM_invalid_object_desc`
`CSM_invalid_maxlength`

*/

Operations on cache objects

-----*/

11.9. `CSM_delete_object`

`error_typ`

`CSM_delete_object(handle, object_p)`

<code>ASE_session_number_typ</code>	<code>handle;</code>	<code>/* IN: cache handle */</code>	(pg483.24)
<code>CSM_object_desc_typ</code>	<code>*object_p;</code>	<code>/* IN: object description. wildcards allowed */</code>	(pg94.16)

+++ CSM - Cache Service Manager +++

/* Performs a delete operation on the object(s) indicated. This action will delete an object from a non-reference count cache, and will decrement the reference count by 1 for a reference count cache. The object will only be physically removed from a reference count cache if the reference count is 1 prior to this call.

Caller must have logged on to the cache with DELETE access.

*object_p may use the following wildcards:

```
object_p->page == CSM_ALL_PAGES
--or--
object_p->page == CSM_ALL_PAGES and object_p->id == CSM_ALL_DOCS
--or--
object_p->page == CSM_ALL_PAGES and object_p->id == CSM_ALL_DOCS and
object_p->ssn == CSM_ALL_SSNS
```

NOTE: If wildcarded, the initial value of the object_p->page field may be altered by this entry point.

ERRORS: CSM_logon_access_mode
CSM_invalid_wildcard
CSM_no_permission
CSM_no_such_object (only returned if wildcards not used)

*/

11.10. CSM_find_objects

error_typ

CSM_find_objects (handle, id_pattern_p, last_object_p, rel_op, max_objects, objects_p, num_returned_p, done_p)

```
ASE_session_number_typ handle; /* IN: cache handle */ (pg483.24)
CSM_object_desc_typ* id_pattern_p; /* IN: object id (wildcards ok) */ (pg94.16)
CSM_object_desc_typ* last_object_p; /* IN: start point in wildcard */ (pg94.16)
ASE_relational_op_typ rel_op; /* IN: EQL, GEQ, or GTR */ (pg484.34)
unsigned max_objects; /* IN: max # objects to return */
CSM_object_desc_typ* objects_p; /* OUT: array of objects returned*/ (pg94.16)
unsigned* num_returned_p; /* OUT: #objects returned */
bool* done_p; /* OUT: if all objects for given (pg491.22)
id_pattern_p returned */
```

/* Returns a list of objects found in the cache.

The values in *id_pattern_p can be:

To match one particular object:
ssn=<some ssn>, id=<some id>, page=<some pagenumber>

+++ CSM - Cache Service Manager +++

To match all pages for a document:

 ssn=<some ssn>, id=<some id>, page=CSM_ALL_PAGES

To match all objects for a particular system (based on ssn):

 ssn=<some ssn>, id=CSM_ALL_DOCS, page=CSM_ALL_PAGES

To match all objects in the specified cache:

 ssn=CSM_ALL_SSNS, id=CSM_ALL_DOCS, page=CSM_ALL_PAGES

If `done_p` is returned FALSE, the query can be continued by setting `rel_op` to GTR and setting `last_object_p` to the last object descriptor returned by the previous call to `find_objects`.

The size of the data area pointed to by the `objects_p` pointer must be at least as big as the `CSM_object_desc_typ` structure times the number of object to be returned. (i.e `objects_p` data size \geq `sizeof(CSM_object_desc_typ) * max_objects`).

ERRORS: CSM_no_such object (only returned if wildcards used)

*/

11.11. CSM_get_cache_attributes

error_typ

CSM_get_cache_attributes(handle, attr_p)

 ASE_session_number_typ handle; /* IN: Handle for this cache */ (pg483.24)

 CSM_cache_attr_typ *attr_p; /* OUT: Cache attributes returned */ (pg93.47)

/* Returns attributes about the given cache.

ERRORS: none other than network errors or system malfunctions.

*/

11.12. CSM_get_obj_attr_2

error_typ

CSM_get_obj_attr_2 (handle, object_p, attr_p, lastwrite_p)

 ASE_session_number_typ handle; /* IN: cache handle */ (pg483.24)

 CSM_object_desc_typ* object_p; /* IN: object descriptor */ (pg94.16)

 CSM_object_attr_typ* attr_p; /* OUT: regular obj attributes */ (pg94.35)

 ASE_time_typ* lastwrite_p; /*OUT:time created/last updated */ (pg489.31)

/* This entry is a superset of `CSM_get_object_attributes`.

In addition to returning the regular object attributes in `attr_p`, it also returns the last write time, which is the time the object is created or the last update time, whichever is later.

The last update is taken from the `lastupdate` column in `csm_used_space`

+++ CSM - Cache Service Manager +++

table in transient db.

Lastupdate is treated as a CSM internal object attribute.

Unlike created or last_read, it cannot be modified by CSM client, and it can only be inquired via this entry point.

ERRORS: CSM_no_such_object

*/

11.13. CSM_get_object_attributes

error_typ

CSM_get_object_attributes(handle, object_p, attr_p)

ASE_session_number_typ handle; /* IN: cache handle */ (pg483.24)

CSM_object_desc_typ* object_p; /* IN: object descriptor */ (pg94.16)

CSM_object_attr_typ* attr_p; /* OUT: object attributes */ (pg94.35)

/* Returns the objects attributes. Gets attributes even if object is open. Will return attributes which reflect last modify call, even if CSM_close_object has not been called to make modifications permanent.

ERRORS: CSM_no_such_object

*/

11.14. CSM_logoff

error_typ

CSM_logoff(handle)

ASE_session_number_typ handle; /* IN: Handle for cache service */ (pg483.24)

/* Logs off of the cache service. Closes any objects which are currently open on this handle.

ERRORS: none other than network errors or system malfunctions.

*/

11.15. CSM_logon

error_typ

CSM_logon(service_name_p, leave_open, mode, timeout_p, handle_p)

ASE_service_name_typ* service_name_p; /* IN: name of cache */ (pg486.28)

bool leave_open; /* IN: if leave connection open */ (pg491.22)

CSM_cache_access_mode mode; /* IN: open mode - (pg93.8)

CSM_CACHE_READ

+++ CSM - Cache Service Manager +++

CSM_CACHE_WRITE
CSM_CACHE_DELETE
CSM_CACHE_READWRITE
CSM_CACHE_READDELETE
CSM_CACHE_READWRITEDELETE */

unsigned* timeout_p; /* OUT: timeout (obsolete) */
ASE_session_number_typ* handle_p; /* OUT: handle returned */ (pg483.24)

/* Logs on to a cache service. The handle returned can be used for subsequent calls to this service.

ERRORS: CSM_no_such_cache (Cache does not exist)
 CSM_err_cannot_find_named_service (Not a cache or configuration error)

*/

11.16. CSM_lookup_cache

error_typ

CSM_lookup_cache(ssn, cache, service_name_p)
ASE_ssn_typ ssn; /* IN: ssn of cache */ (pg483.47)
CSM_cache_id_typ cache; /* IN: cache id of cache */ (pg93.22)
ASE_service_name_typ *service_name_p; /* OUT: name of cache */ (pg486.28)

/* Each cache is uniquely identified by a cache name, and also a ssn/cache id pair. This routine returns the cache name for a given ssn and cache id.

This routine will only translate a cache id/ssn pair to a cache name if a prior CSM_logon to the same domain has been done.

ERRORS: CSM_no_such_known_cache */

11.17. CSM_modify_object_attributes

error_typ

CSM_modify_object_attributes(handle, oh, attr_p)
ASE_session_number_typ handle; /* IN: cache handle */ (pg483.24)
CSM_object_handle_typ oh; /* IN: object handle */ (pg93.23)
CSM_object_attr_typ* attr_p; /* IN: new attributes (max_length can't be changed) */ (pg94.35)

/* Modifies the objects attributes with the new attributes specified. The new attributes are saved in memory and will be returned CSM_get_object_attributes while this object is open, but the new attributes will only be saved in the database when CSM_close_object is called with the update flag = TRUE.

+++ CSM - Cache Service Manager +++

This routine cannot be used to change the `max_length` field (see `CSM_resize_object`). Attempts to change `max_length` will be ignored.

ERRORS: `CSM_open_access_mode`
`CSM_no_ageable_docs`
`CSM_ilgl_csum_update`

*/

11.18. `CSM_move_object`

`error_typ`

`CSM_move_object(handle, target_handle, target_service_p, object_p,`
`new_object_p)`

<code>ASE_session_number_typ</code>	<code>handle;</code>	<code>/* IN: cache handle */</code>	<code>(pg483.24)</code>
<code>ASE_session_number_typ</code>	<code>target_handle;</code>	<code>/* IN: destination cache</code> <code> handle or NULL */</code>	<code>(pg483.24)</code>
<code>ASE_service_name_typ*</code>	<code>target_service_p;</code>	<code>/* IN: destination cache name</code> <code> (must be specified) */</code>	<code>(pg486.28)</code>
<code>CSM_object_desc_typ*</code>	<code>object_p;</code>	<code>/* IN: object to copy */</code>	<code>(pg94.16)</code>
<code>CSM_object_desc_typ*</code>	<code>new_object_p;</code>	<code>/* OUT: new object name */</code>	<code>(pg94.16)</code>

/* Moves an object. The destination object will be created with duration of `CSM_LOCKED_DURATION`. The source object will be deleted (or the reference count will be decremented if it was previously greater than 1).

Setting `new_object_p->id = ASE_INVALID_DOC_ID` will cause the service to generate a temporary object descriptor for the target object. This temp object descriptor will be returned to the client.

If `*object_p` refers to a temporary object descriptor, `*new_object_p` must not be the same as `*object_p`.

If the new object already exists and reference counts are enabled, the reference count in the destination object will be incremented, and no error will be returned.

ERRORS: `CSM_no_permission`
`CSM_already_exists`
`CSM_object_busy`
`CSM_invalid_checksum`
`CSM_invalid_object_desc`

*/

+++ CSM - Cache Service Manager +++

11.19. CSM_open_connection

```
error_typ
CSM_open_connection(handle)
    ASE_session_number_typ handle; /* IN: cache handle */ (pg483.24)

/* Opens a network connection for use with CSM. Connections
should not be left open for long periods (e.g. minutes).
Opening connections and then issuing a series of CSM calls
can be used to improve performance.

ERRORS: network errors
*/
```

11.20. CSM_open_csum_object

```
error_typ
CSM_open_csum_object (handle, object_p, mode, oh_p, attr_p,
                    csum_exists_p, csum_p)
    ASE_session_number_typ handle; /* IN: cache handle */ (pg483.24)
    CSM_object_desc_typ* object_p; /* IN: cache object */ (pg94.16)
    CSM_object_access_mode mode; /* IN: open mode - (pg92.19)
                                CSM_OBJECT_READ (read only)
                                CSM_OBJECT_WRITE (read & write)
                                CSM_OBJECT_READNOUPDATE */
    CSM_object_handle_typ* oh_p; /* OUT: handle used for object */ (pg93.23)
    CSM_object_attr_typ* attr_p; /* OUT: object attributes */ (pg94.35)
    bool* csum_exists_p; /* OUT: if checksum exists */ (pg491.22)
    long* csum_p; /* OUT: checksum, if exists */

/* Opens the object for subsequent reads or writes. Just like
CSM_open_object, except that it also returns the value of the object
checksum, if any.
```

The checksum on an object is computed by the client of cache services.
Cache services will not inspect or set this checksum.

The "mode" flag has the following effects:

```
CSM_OBJECT_READ: Opens the object in a shared mode. Others may
also have the object open for read at the same
time. The object may only be read if opened in
this mode.
CSM_OBJECT_WRITE: Opens the object in exclusive mode for either
reads or writes. No other entity may have the
object open when this mode is used.
CSM_OBJECT_READNOUPDATE: Same as CSM_OBJECT_READ, except that it won't
```

+++ CSM - Cache Service Manager +++

update the object to record the "last_read" time.
This option is a performance optimization
over the CSM_OBJECT_READ mode.

If two users try to open the same object and one of the two users specifies CSM_OBJECT_WRITE mode, the first opener of the object will succeed, and the second opener will block until either the first opener closes the object, or 30 seconds has elapsed. If the 30 second timeout elapses a CSM_object_busy error is returned and the open request is refused.

When calling CSM_open_csum_object() through the WAL library, it uses one DBP buffer. When it is used in combination with DOC_batch_create, CSM_create_object, CSM_open_object, CSM_open_csum_object, BES_create_image, BES_open_image or BES_update_image, there could be need to increase the number of DBP buffers on the IMS server. There is the potential for deadlocking the IMS if multiple applications are running at the same time and each program calls more than one of the entry points above and there are few DBP buffers on the IMS server.

ERRORS: CSM_object_busy
CSM_no_permission
CSM_no_such_object
CSM_invalid_mode

*/

11.21. CSM_open_object

error_typ

```
CSM_open_object(handle, object_p, mode, oh_p, attr_p)
  ASE_session_number_typ  handle;      /* IN:  cache handle */           (pg483.24)
  CSM_object_desc_typ*    object_p;    /* IN:  object to open */       (pg94.16)
  CSM_object_access_mode  mode;        /* IN:  open mode -
                                         CSM_OBJECT_READ  (read only)
                                         CSM_OBJECT_WRITE (read & write)
                                         CSM_OBJECT_READNOUPDATE */
  CSM_object_handle_typ*  oh_p;        /* OUT: handle used to access object */ (pg93.23)
  CSM_object_attr_typ*    attr_p;      /* OUT: attributes of object */    (pg94.35)
```

/* Opens the object for subsequent reads or writes.

The "mode" flag has the following effects:

CSM_OBJECT_READ: Opens the object in a shared mode. Others may also have the object open for read at the same

+++ CSM - Cache Service Manager +++

time. The object may only be read if opened in this mode.

CSM_OBJECT_WRITE: Opens the object in exclusive mode for either reads or writes. No other entity may have the object open when this mode is used.

CSM_OBJECT_READNOUPDATE: Same as CSM_OBJECT_READ, except that it won't update the object to record the "last_read" time. This option is a performance optimization over the CSM_OBJECT_READ mode.

If two users try to open the same object and one of the two users specifies CSM_OBJECT_WRITE mode, the first opener of the object will succeed, and the second opener will block until either the first opener closes the object, or 30 seconds has elapsed. If the 30 second timeout elapses a CSM_object_busy error is returned and the open request is refused.

When calling CSM_open_object() through the WAL library, it uses one DBP buffer. When it is used in combination with DOC_batch_create, CSM_create_object, CSM_open_object, CSM_open_csum_object, BES_create_image, BES_open_image or BES_update_image, there could be need to increase the number of DBP buffers on the IMS server. There is the potential for deadlocking the IMS if multiple applications are running at the same time and each program calls more than one of the entry points above and there are few DBP buffers on the IMS server.

ERRORS: CSM_object_busy
CSM_no_permission
CSM_no_such_object
CSM_invalid_mode

*/

11.22. CSM_read_object

error_typ

```
CSM_read_object(handle, oh, offset, read_len, data_p, buf_size, bytes_rd_p)
  ASE_session_number_typ handle; /* IN: cache handle */ (pg483.24)
  CSM_object_handle_typ oh; /* IN: object handle */ (pg93.23)
  unsigned offset; /* IN: byte offset to start read at */
  unsigned read_len; /* IN: #read_len to read */
  char* data_p; /* IN: buffer pointer */
  unsigned buf_size; /* IN: size of buffer pointed to by
  data_p */
  unsigned* bytes_rd_p; /* OUT: #bytes read */
```


+++ CSM - Cache Service Manager +++

/* Reads the number of bytes requested from the specified offset in the object. If the read goes past the end of the object but starts within the object, *bytes_rd_p will equal the object size minus the starting offset (less than "read_len"), and no error will be returned. If the read starts past the end of the object, a CSM_out_of_range error will be returned.

For best performance, start reads on a 1024 byte boundary, and make buf_size a multiple of 1024 also. Note that "buf_size" should always be at least "read_len" rounded up to a multiple of 1024.

ERRORS: CSM_out_of_range
CSM_IO_error

*/

11.23. CSM_rename_object

error_typ

CSM_rename_object(handle, old_object_p, new_object_p)

ASE_session_number_typ handle; /* IN: cache handle */ (pg483.24)
CSM_object_desc_typ* old_object_p; /* IN: object to rename */ (pg94.16)
CSM_object_desc_typ* new_object_p; /* IN/OUT: new object name */ (pg94.16)

/* Renames the object from the old name to the new name.

Setting new_object_p->id = ASE_INVALID_DOC_ID will cause the service to generate a temporary object descriptor for the target object. This temp object descriptor will be returned to the client in *new_object_p.

If the new object already exists and reference counts are enabled, the reference count in the destination object will be incremented, and no error will be returned.

If the old object has a reference count of greater than 1, the old object is copied to the new object and the reference count of the old object is decremented.

ERRORS: CSM_no_such_object
CSM_logon_access_mode
CSM_no_permission
CSM_already_exists

*/

11.24. CSM_resize_object

error_typ

+++ CSM - Cache Service Manager +++

```
CSM_resize_object(handle, object_p, new_size)
  ASE_session_number_typ    handle;      /* IN: session handle */      (pg483.24)
  CSM_object_desc_typ*      object_p;    /* IN: object descriptor */  (pg94.16)
  unsigned                   new_size;   /* IN: new object size (bytes) */

/* Resizes the object. Object may be made larger or smaller, but may
not be made smaller than the "cur_length" attribute.

ERRORS: CSM_logon_access_mode
        CSM_invalid_object_size
        CSM_no_resources
        CSM_object_busy
        CSM_no_such_object
*/
```

11.25. CSM_write_object

```
error_typ
CSM_write_object(handle, oh, offset, bytes, data_p, buf_size, bytes_wrt_p)
  ASE_session_number_typ  handle;      /* IN: cache handle */      (pg483.24)
  CSM_object_handle_typ   oh;          /* IN: object handle */     (pg93.23)
  unsigned                 offset;     /* IN: byte offset in object
                                     to start write at */
  unsigned                 bytes;      /* IN: #bytes to write */
  char*                    data_p;     /* IN: pointer to buffer */
  unsigned                 buf_size;   /* IN: size of buffer pointed to
                                     by "data_p" */
  unsigned*                bytes_wrt_p; /* OUT: #bytes written (=bytes) */

/* Writes an object at the offset specified for the length specified.
Write goes directly to magnetic disk with no buffering, therefore
data will be present after a system reboot if the write completes,
even if the CSM_close_object is yet not done.
```

For higher performance, writes should start on 1024 byte boundaries, and the buffer size ("buf_size") should be a multiple of 1024 bytes.

Writing to an object which already has a checksum is not allowed. If you need to update an object, create a new object, delete the old object, and rename the new object to the old object name.

Writing to an object which has been previously created may result in only part of the object being written if a system crash occurs. This condition should be avoided by doing a rename (see above) if partial writes would cause application problems.

ERRORS: CSM_open_access_mode

+++ CSM - Cache Service Manager +++

CSM_logon_access_mode
CSM_ilgl_csum_update
CSM_out_of_range
CSM_IO_error

*/

12. DOC DECLARATIONS

/* This is the implementation for Document Services (DOC).

Document Services provides for the management of documents, in particular their storage on and migration from optical disk. Documents are dealt with as whole entities or as collections of pages; Document Services does not recognize the format of a page.

Clients need to also use Cache Services in order to use many services of Document Services.

Document Services includes functions for the committal and deletion of documents, their migration between caches and optical disks, and the retrieval of information on documents' locations on optical disk, on optical disk surfaces, and on optical disk families. It also provides the ability to annotate pages of documents with text data.

Document Services maintains an internal database called the locator database which contains data on each document, optical disk surface, optical disk family, and cluster that has been created on or imported onto the system. The locator database can be queried for much of this data; in general it cannot be updated.

A typical document goes through several stages in its lifetime. The first three stages below typically happen only once in the life of a document, during its entry into the system. The final stage happens only once and defines the end of the document's lifetime.

1. A series of IDs are allocated by a call to `DOC_allocate_ids` to be used as image IDs for the pages of the document.
2. The pages are created in a cache using Cache Services.
3. The document is committed using `DOC_commit_document`, indicating that it should be migrated to optical disk.
4. The document is migrated to a cache using `DOC_migrate_from_optical_disk`, where it may be retrieved using Cache Services. This may happen many times in the life of the document.
5. Eventually it may be deleted with `DOC_delete_documents`.

Note: Any service name which has a zero length object field or is passed in as a nil pointer (if the interface specifies a pointer) implies that the default service name will be used for that service. This is not true for the service name passed to `DOC_logon`.

*/
#ifndef DOC_defs

+++ DOC - Document Services +++

```
#define DOC_defs

#include <AS_externals.h>
#include <CSM.defs>
#include <INX.defs>
#include <SEC.defs>

#define DOC_MAX_DELETIONS      100      /* Max # of deletions in 1 call */
#define DOC_MAX_ID_ALLOCATION    5000     /* Max. # of IDs allocated in 1 call*/
#define DOC_MAX_PREFETCHES     1000     /* Max. # of prefetches in 1 call */
#define DOC_MAX_PAGES_PER_BATCH 10000   /* Max. # of pages in a doc batch */
#define DOC_MAX_ASYNC_MIGRATES  1000    /* Max. # of migrates in 1 call */
#define DOC_MAX_DOC_SURF_LOCS   1000    /* Max. # of doc_surf_loc's in 1 call */

#define DOC_PSEUDO_REQUEST_ID   2468

/* The inbox priority of a surface, which controls which surface gets
   ejected to make room in a full osar. */

#define DOC_DEFAULT_INBOX_PRIORITY 128
#define DOC_MAX_INBOX_PRIORITY     255
typedef unsigned char              DOC_inbox_priority_typ; /* range = 0..255*/

/* The description of where a surface is */

#define DOC_IN_DRIVE              1      /* this side of surface is loaded by
                                           an optical drive*/
#define DOC_IN_OSAR                2      /* in osar but this side not loaded by
                                           an optical drive. */
#define DOC_NOT_IN_OSAR           3      /* not in an osar */
#define DOC_NOT_ON_OPT_DISK       4      /* not on an optical disk (write
                                           request pending or committed with
                                           no_migrate option) */

typedef unsigned short             DOC_surf_loc_typ;

/* The description of the most easily accessed surface a document is on, and
   where that surface is. */

typedef struct {
    ASE_surface_id_typ  surf_id;          /* Surface document would most      (pg486.47)
                                           likely be retrieved from. */
    DOC_surf_loc_typ    surf_loc;         /* Location of surface */           (pg119.34)
    bool                intervention_req; /* If the operator intervention is   (pg491.22)
                                           required to get the surface
                                           because the surface is out of
                                           box, or is in a disabled slot,
                                           drive, or osar. */
    bool                in_cache;         /* If the pages requested of the     (pg491.22)
                                           surface are in cache. */
} DOC_surf_desc_typ;
```

+++ DOC - Document Services +++

```
error_typ          err;          document are in cache. */
                                  /* Error getting this info. If (pg493.26)
                                  non-zero, all info above is
                                  not valid. */

} DOC_doc_surf_loc_typ;

/* The description of the location, etc. of a document on optical disk. */

typedef struct
{
ASE_doc_id_typ      doc_id;          (pg483.37)
  /* Its ID */
ASE_surface_id_typ  primary_surface_id; (pg486.47)
  /* The ID of the surface on which the family copy resides;
  set to ASE_INVALID_SURFACE_ID if the primary copy has
  not been written to optical disk */
ASE_surface_offset_typ primary_offset; (pg487.2)
  /* The offset on the surface where the family copy resides;
  set to ASE_INVALID_SURFACE_OFFSET if the primary copy has
  not been written to optical disk */
ASE_document_status_typ primary_status; (pg487.6)
  /* The status of the family copy */
ASE_surface_id_typ  secondary_surface_id; (pg486.47)
  /* The ID of the surface on which the transaction copy resides;
  set to ASE_INVALID_SURFACE_ID if the secondary copy has
  not been written to optical disk */
ASE_surface_offset_typ secondary_offset; (pg487.2)
  /* The offset on the surface where the transaction copy resides;
  set to ASE_INVALID_SURFACE_OFFSET if the secondary copy has
  not been written to optical disk */
ASE_document_status_typ secondary_status; (pg487.6)
  /* The status of the transaction copy */
ASE_page_number_typ  pages;          (pg483.41)
  /* The number of pages in the document */
ASE_ssn_typ          original_ssn;   (pg483.47)
  /* The SSN of the system on which this document was created;
  set to ASE_LOCAL_SSN if it was created on the local system */
ASE_doc_id_typ      original_doc_id; (pg483.37)
  /* The document ID by which this document is known on the system where
  where it was created; set to ASE_INVALID_DOC_ID if it was created
  on the local system */
SEC_access_restrictions security;    (pg385.2)
  /* The access restrictions on the document */
ASE_service_name_typ  cache;         (pg486.28)
  /* The name of the cache in which the document resides if it has not
  been migrated to optical disk, or null if it has. */
} DOC_doc_location_desc_typ;
```

+++ DOC - Document Services +++

```
typedef struct
{
ASE_doc_id_typ      doc_id;                      (pg483.37)
  /* Its ID */
ASE_surface_id_typ  primary_surface_id;          (pg486.47)
  /* The ID of the surface on which the family copy resides;
   set to ASE_INVALID_SURFACE_ID if the primary copy has
   not been written to optical disk */
ASE_surface_offset_typ  primary_offset;          (pg487.2)
  /* The offset on the surface where the family copy resides;
   set to ASE_INVALID_SURFACE_OFFSET if the primary copy has
   not been written to optical disk */
ASE_document_status_typ  primary_status;          (pg487.6)
  /* The status of the family copy */
ASE_surface_id_typ      secondary_surface_id;    (pg486.47)
  /* The ID of the surface on which the transaction copy resides;
   set to ASE_INVALID_SURFACE_ID if the secondary copy has
   not been written to optical disk */
ASE_surface_offset_typ  secondary_offset;        (pg487.2)
  /* The offset on the surface where the transaction copy resides;
   set to ASE_INVALID_SURFACE_OFFSET if the secondary copy has
   not been written to optical disk */
ASE_document_status_typ  secondary_status;        (pg487.6)
  /* The status of the transaction copy */
ASE_page_number_typ     pages;                   (pg483.41)
  /* The number of pages in the document */
ASE_ssn_typ             original_ssn;            (pg483.47)
  /* The SSN of the system on which this document was created;
   set to ASE_LOCAL_SSN if it was created on the local system */
ASE_doc_id_typ          original_doc_id;         (pg483.37)
  /* The document ID by which this document is known on the system where
   where it was created; set to ASE_INVALID_DOC_ID if it was created
   on the local system */
SEC_access_restrictions security;                (pg385.2)
  /* The access restrictions on the document */
ASE_service_name_typ    cache;                   (pg486.28)
  /* The name of the cache in which the document resides if it has not
   been migrated to optical disk, or null if it has. */
FN_uint32_t             sds_id;                  /* SDS unit number */ (pg491.9)
} DOC_doc_location_w_sds_desc_typ;
```

/* The description of an optical disk surface. */

```
typedef struct
{
ASE_surface_id_typ  surface_id;                  (pg486.47)
  /* Its ID */
```

+++ DOC - Document Services +++

```

INX_fam_id_typ    family_id;                                (pg243.25)
    /* The ID of the family to which this surface belongs;
       family ID number 1 is the transaction logging family */
bool              surface_unavailable;                    (pg491.22)
    /* TRUE if the surface should not be used for reads or writes */
bool              write_protect;                          (pg491.22)
    /* TRUE if the surface cannot be used for writes */
unsigned short    sides;
    /* The number of sides on the cartridge containing the surface */
ASE_time_typ     label_date;                              (pg489.31)
    /* The date and time at which the surface was entered in the system */
ASE_time_typ     full_date;                              (pg489.31)
    /* The date and time at which the surface filled up */
ASE_disk_typ     disk_type;                              (pg487.34)
    /* The type of this disk */
unsigned long     surface_size;
    /* The number of 1024-byte sectors on the surface */
ASE_ssn_typ      original_ssn;                          (pg483.47)
    /* The SSN of the system on which the surface was originally created;
       set to ASE_LOCAL_SSN if local */
ASE_surface_id_typ original_surface_id;                  (pg486.47)
    /* The ID of the surface as it was known on its original system;
       set to ASE_INVALID_SURFACE_ID if local */
unsigned long     active_docs;
    /* The number of currently accessible documents on the surface */
unsigned long     deleted_docs;
    /* The number of deleted documents on the surface */
unsigned long     clusters;
    /* The number of clusters on the surface */
unsigned long     sectors;
    /* The number of sectors consumed on the surface */
unsigned long     pages;
    /* The total number of pages in all the active and deleted documents
       on the surface */
ASE_server_id_typ server_id;                              (pg484.8)
    /* The ID of the osar server which currently has this surface */
} DOC_surface_desc_typ;

/* The description of optical disk family information for a server.
   Used only in the DOC_family_desc_type (see below). */

typedef struct
{
ASE_server_id_typ server_id;                              (pg484.8)
    /* The ID of the osar server */
unsigned short     desired_current_surfaces;
    /* The number of surfaces that can be current (i.e. writable) at
       any one time for the family */

```


+++ DOC - Document Services +++

```
unsigned short    actual_current_surfaces;
    /* The number of surfaces that are current at the moment; this will
       deviate from desired_current_surfaces only if the latter has
       changed recently */
ASE_surface_id_typ current_surfaces[ASE_MAX_CURR_SURFS];           (pg486.47)
    /* The IDs of the current surfaces; only the first
       actual_current_surfaces IDs are valid */
ASE_surface_id_typ previous_surfaces[ASE_MAX_CURR_SURFS];         (pg486.47)
    /* A sequence of IDs for the previous surfaces; this only has
       valid values if interleaving (see below) is TRUE (then only the
       first actual_current_surfaces IDs are valid); it is used when
       interleaving to determine which cartridge to return to */
char              preferred_osars[ASE_MAX_CURR_SURFS];
    /* The OSAR IDs of OSARs into which new surfaces will be placed;
       only the first desired_current_surfaces IDs are valid;
       OSAR Services will step through the sequence when placing new
       cartridges */
unsigned short    num_future_surfaces;
    /* The surfaces in the "future_surfaces" array */
ASE_surface_id_typ future_surfaces[ASE_MAX_FUTURE_SURFS];         (pg486.47)
    /* The surfaces which will be written to after the current_surfaces
       are full, but before a new surface id will be allocated. Surface
       ids are put in this field when a surface is enabled. */
} DOC_family_server_desc_typ;
```

```
/* The description of an optical disk family.
   A family is a stream of optical disk surfaces, used to divide different
   classes across a group of disks. */
```

```
typedef struct
{
    INX_fam_name_typ    family_name;           (pg243.24)
    /* The name of the family */
    INX_fam_id_typ     family_id;             (pg243.25)
    /* The (per-system) unique ID of the family */
    bool               is_primary;            (pg491.22)
    /* TRUE if family is primary; FALSE if is transaction log */
    INX_fam_id_typ     tran_families [ASE_MAX_TRAN_FAMILIES];     (pg243.25)
    /* Transaction log family ids if is_primary = TRUE; else undefined.
       Value of ASE_INVALID_FAMILY_ID indicates no transaction log,
       otherwise it must be a valid family id of a transaction log
       family. */
    unsigned short     interleave_cnt;
    /* Number of surfaces to interleave. One implies write surfaces
       A B A B; two implies A A B B; three implies A A A B B B; etc.
       As an example, if the interleave count is one, surfaces would
       be written in order 3000, 3001, 3002, 3003, 3004, 3005, etc, but
       if the interleave count is 3, surfaces would be written in order
```

+++ DOC - Document Services +++

```
    3000, 3002, 3004, 3001, 3003, 3005, etc. */
ASE_disk_type_typ      disk_type;                (pg487.34)
/* The type of disk that is used for all disks in this family. */
unsigned short        num_servers;
/* The number of server descriptions in server_descs (below) */
DOC_family_server_desc_typ  server_descs[ASE_MAX_OSAR_SERVERS]; (pg123.24)
/* Information related to the servers which currently commit documents
   to this surface */
} DOC_family_desc_typ;

/* The information for a server used in creating a new optical disk family. */

typedef struct {
ASE_server_id_typ    server_id;                (pg484.8)
unsigned short      desired_current_surfaces;
bool                preferred_osars_valid;      (pg491.22)
char                preferred_osars[ASE_MAX_CURR_SURFS];
} DOC_family_create_server_typ;

typedef ASE_fam_id_typ  DOC_famid_ary [ASE_MAX_TRAN_FAMILIES]; (pg487.25)

/* The "type" of an index value.
This is a variable-length structure, dependent on the length of the data.
An instance of this type always begins on word (2-byte) boundaries.
The index_name is ASE_MAX_IXNAME+1 bytes (plus a pad byte to make it even).
The type is 2 bytes.
The string length, if present, is 2 bytes.
The data is padded to a 2 or 4 byte boundary, depending on the platform
this software is run on. Use the entry point "INX_index_data_size" to
determine the length of the "data" field and its padding.
All data is "left justified", with pad put on the end of the data field.
Strings do not have terminating nulls. */

typedef struct
{
INX_index_name_typ  index_name;                (pg235.31)
/* the name of the index */
char                _pad[3]; /* align 'data' below on long word */
/* not used, just to pad to even boundary */
INX_value_type_typ  type;                    (pg235.48)
/* the type of the data item */
char                data[2];
/* the value which is type-dependent, can be longer or shorter */
} DOC_index_value_typ;

/* The description of information used to commit a document. */

typedef struct
```

+++ DOC - Document Services +++

```

{
ASE_doc_id_typ          doc_id;                                (pg483.37)
    /* The document ID of the document to be committed */
ASE_service_name_typ    cache;                                (pg486.28)
    /* The name of the cache from which to get the pages of the doc */
ASE_ssn_typ             ssn;                                  (pg483.47)
    /* The ssn that all the images have in their object descriptors */
unsigned short          images_num;
    /* The number of images pointed to by images (below) */
ASE_image_id_typ        *images;                             (pg483.33)
    /* A list of image IDs that are to make up the pages of the document;
       all must be located in the above named cache, all must have
       the ssn indicated above, and all must have a page number of
       ASE_INVALID_PAGE_NUMBER */
INX_fam_id_typ          family_id;                            (pg243.25)
    /* The family to which the document should be committed */
SEC_access_restrictions security;                             (pg385.2)
    /* The access restrictions to be imposed on the document */
INX_dcl_name_typ        doc_class;                            (pg239.47)
    /* The name of the document class to which the document belongs */
INX_doc_type_typ        doc_type;                             (pg234.11)
    /* The type of the document (e.g. image, text, form, mixed) */
INX_cluster_key_typ     cluster;                              (pg243.39)
    /* The cluster key for the document; use INX_INVALID_CLUSTER_KEY
       if clustering is not desired */
bool                    tolerate_dups;                       (pg491.22)
    /* If this field is TRUE and the doc_id in this structure is a
       duplicate of an already committed document, then the
       DOC_commit_doc procedure will do nothing and return successfully.
       This should be used only when re-committing one or more documents
       when it is not known if the documents committed successfully
       before. */
unsigned short          indexes_len;
    /* The (exclusive) length in bytes of all the indexes that are pointed
       to by indexes (below) */
unsigned short          indexes_num;
    /* The number of indexes pointed to by indexes (below) */
DOC_index_value_typ     *indexes;                             (pg124.44)
    /* A list of indexes by which the document is going to be indexed.
       This list should include user indexes only--no system indexes.
       The user indexes are the ones contained in the "indexes" array
       in the INX_dcl_index_desc_typ structure, and also have the
       "is_system" field in the INX_index_desc_typ set to FALSE.
       Note that the calls which pass INX_doc_index_rec_typ records (such
       as INX_create_DIR and INX_find_DIRS) have both system indexes and
       user indexes in the INX_doc_index_rec_typ structure, and therefore
       contain more indexes than those passed in the DOC_committal_desc_typ
       structure. */

```

+++ DOC - Document Services +++

```
} DOC_committal_desc_typ;
```

```
typedef enum {  
    DOC_not_cursurf,  
    DOC_sideA_cursurf,  
    DOC_sideB_cursurf  
} DOC_cursurf_typ;
```

```
/* The DOC_migrate_order_typ controls the order in which pages are migrated  
from optical disk to cache. Ascending means a 5 page document will be  
migrated page 1 first and page 5 last. Descending means page 5 is  
migrated first and page 1 last. The presence of EXACT on the keyword  
indicates that only the pages requested will be migrated. Absence of  
exact will cause Document Services to also get the next (ascending or  
descending) pages of the document, but at a slightly lower priority than  
pages explicitly requested. The client also does not wait for the  
additional pages for a non-exact migrate request (migrate status  
returned as done as soon as explicitly requested pages are migrated). */
```

```
typedef unsigned short DOC_migrate_order_typ;  
#define DOC_ASCENDING          1  /* Ascending order, other pages are likely  
to be migrated. */  
#define DOC_DESCENDING        2  /* Descending order, other pages are  
likely to be migrated. */  
#define DOC_EXACT_ASCENDING    3  /* Ascending order, other pages are not  
likely to be migrated. */  
#define DOC_EXACT_DESCENDING  4  /* Descending order, other pages are not  
likely to be migrated. */
```

```
typedef unsigned short DOC_priority_typ; /* priority of a retrieval */  
#define DOC_PRIO_LOW          0  
#define DOC_PRIO_MID          1  
#define DOC_PRIO_HIGH         2
```

```
typedef unsigned short DOC_queue_num_typ;  
#define DOC_NO_QUEUE_NUM      0xffff  
#define DOC_MAX_QUEUE_NAME_LEN 11
```

```
/* The "notify data" record is passed into DOC_async_migrate_from_od, and  
when the migrate completes, the notify data as well as completion status  
(success or error) is returned via DOC_read_queue. This data can be  
anything the user wants, and is helpful in determining the request which  
just finished. Examples of information which can be put in the notify data  
are doc_id, ssn, first page, and last page. */
```

```
typedef struct {  
    unsigned long      userid0;  
    unsigned long      userid1;
```

+++ DOC - Document Services +++

```
unsigned short      userid2;
unsigned short      userid3;
} DOC_notify_data_typ;
```

/* The "async comp" record is returned to indicate completion status of each migrate initiated via DOC_async_migrate_from_od. The notify_data is the same as that which was passed in, and the request id is the same as the request id in the "DOC_async_rcv_doc_typ". The error is zero if the migrate was successful, and non-zero otherwise. */

```
typedef struct {
    DOC_notify_data_typ    notify_data;                (pg127.3)
    ASE_request_id_typ     request_id;                (pg486.15)
    error_typ              error;                    (pg493.26)
} DOC_async_comp_typ;
```

/* The "async xmit doc" record is used to submit a migrate request */

```
typedef struct {
    ASE_doc_id_typ         doc_id;      /* doc id of doc to retrieve      */ (pg483.37)
    ASE_ssn_typ            ssn;         /* ssn of doc to retrieve          */ (pg483.47)
    ASE_page_number_typ    first_page; /* first page to retrieve          */ (pg483.41)
    ASE_page_number_typ    last_page;  /* last page to retrieve           */ (pg483.41)
    ASE_doc_id_typ         dest_doc_id; /* rename to this id when retrieved */ (pg483.37)
    ASE_ssn_typ            dest_ssn;   /* rename to this ssn when retrieved */ (pg483.47)
    DOC_notify_data_typ    notify_data; /* data returned on completion     */ (pg127.3)
} DOC_async_xmit_doc_typ;
```

/* The async rcv record is returned for each migrate initiated. This record contains status about the document migrated, as well as information necessary to cancel the migrate, if desired. */

```
typedef struct {
    ASE_page_number_typ    pages_available; /* pages already in cache          */ (pg483.41)
    ASE_page_number_typ    pages_in_doc;   /* #pages in document              */ (pg483.41)
    ASE_service_name_typ   actual_cache;   /* cache doc put into              */ (pg486.28)
    ASE_ssn_typ            local_ssn;      /* local ssn of document           */ (pg483.47)
    ASE_doc_id_typ         local_doc_id;   /* document id of document         */ (pg483.37)
    ASE_request_id_typ     request_id;     /* request id of this doc          */ (pg486.15)
    ASE_server_id_typ      server_id;     /* osar server id of this doc     */ (pg484.8)
    ASE_migrate_status_typ migrate_status; /* if in cache, in slot, etc      */ (pg487.12)
    error_typ              migrate_error;  /* if error on migrate            */ (pg493.26)
} DOC_async_rcv_doc_typ;
```

```
typedef struct /* The ID of an annotation. */
{
    unsigned long    high_order;
    unsigned long    low_order;
```

+++ DOC - Document Services +++

```
    } DOC_annotation_id_typ;

typedef byte *DOC_annotation_typ; /* The type of an annotation. */ (pg491.23)
#define DOC_MAX_ANNOTATION_LEN 800 /* The maximum length of annotation data. */

#define DOC_insert_action          1
#define DOC_export_action         2
#define DOC_delete_action         3
#define DOC_update_action         4
typedef FN_int16_t DOC_ce_action_t; (pg491.6)

#define DOC_MAX_CE_RETURN_REC     500

/* The description of an annotation. */

typedef struct
{
    ASE_doc_id_typ      doc_id; (pg483.37)
    /* The ID of the document which is annotated */
    ASE_page_number_typ page; (pg483.41)
    /* The number of the page which is annotated */
    DOC_annotation_id_typ annot_id; (pg128.1)
    /* The ID of the annotation */
    ASE_time_typ        created; (pg489.31)
    /* The date and time that the annotation was created */
    ASE_time_typ        modified; (pg489.31)
    /* The date and time that the annotation was last modified */
    SEC_access_restrictions security; (pg385.2)
    /* The access restrictions on the annotation */
    unsigned short      annot_len;
    /* The length of the annotation data pointed to by data */
    byte                 annot[DOC_MAX_ANNOTATION_LEN]; (pg491.23)
    /* The annotation data */
} DOC_annotation_desc_typ;

/* Document attribute modification defines. */

#define DOC_SECURITY_ATTRIBUTE      0 /* New document security */
#define DOC_PRIMARY_STATUS_ATTRIBUTE 1 /* New status of primary copy */
#define DOC_SECONDARY_STATUS_ATTRIBUTE 2 /* New status of secondary copy */

/* The description of a document attribute value. */

typedef struct
{
    unsigned short attr_type;
    union
    {
```

+++ DOC - Document Services +++

```
    SEC_access_restrictions security; /* Only for security */          (pg385.2)
    ASE_document_status_typ status; /* For primary & secondary status */(pg487.6)
    } attr_union;
} DOC_doc_attribute_value_typ;

typedef struct {
    unsigned word;
    unsigned sect;
} DOC_page_offset_typ;

/* In the surf_ids and offsets arrays, element 0 is the primary surface,
and element 1 is the transaction log. */

typedef struct {
    ASE_surface_id_typ surf_ids[2];          (pg486.47)
    DOC_page_offset_typ offsets[2];        (pg129.9)
} DOC_page_odloc_typ;

typedef long DOC_iteration_typ;

typedef struct {
    DOC_iteration_typ iteration; /* iteration number */          (pg129.19)
    long pages_written; /* # pages written so far */
    ASE_fam_id_typ cur_fam; /* cur family being written */      (pg487.25)
    ASE_surface_id_typ last_surf; /* last surf written */      (pg486.47)
} DOC_page_hdr_typ;

typedef struct {
    DOC_page_hdr_typ hdr; /* header info */          (pg129.26)
    DOC_page_odloc_typ pgs[1]; /* variable length array */      (pg129.17)
} DOC_page_odloc_struct_typ;

typedef struct {
    unsigned long page_size;
    unsigned long page_offset;
    unsigned long page_or_doc_num; /* page# if <= 1000, doc_id if > 1000 */
} DOC_page_cacheloc_typ;

/* Overhead per page in a fast committal batch */

#define DOC_BATCH_PAGE_OVERHEAD \
    (sizeof(DOC_page_odloc_typ)+sizeof(DOC_page_cacheloc_typ))

/* Max batch name length for DOC batches */

#define DOC_MAX_BATCHNAME_LEN 22

/* Batch name prefixes. Batch services created batches should be
```

+++ DOC - Document Services +++

prefixed with "F_B_", and COLD batches should be prefixed with
"F_C_", using the prefix defines below. */

```
#define DOC_SYS_BATCHNAME_PFX      "F_"
#define DOC_BS_BATCHNAME_PFX      "B_"
#define DOC_COLD_BATCHNAME_PFX    "C_"
```

```
typedef struct {
    unsigned long    cacheloc_offset;
    unsigned long    odloc_offset;
    unsigned short   num_pages;
    bool             has_checksums;
} DOC_clientattr_typ;                                     (pg491.22)
```

```
typedef struct {
    INX_value_type_typ  index_type;    /* index type, INX_VT... */
    short               max_ascii_len; /* if ascii, max string length */
    short               name_len;     /* strlen of external name */
} DOC_inx_info_typ;                                     (pg235.48)
```

/* Document committal options. */

```
typedef unsigned short          DOC_cmt_opt_typ;
#define DOC_COMMIT_OPTION_MIGRATE_DELAY 1
```

```
typedef struct {
    DOC_cmt_opt_typ option_type;    /* Type of option in union structure
                                     below */
    union {
        long         migrate_delay; /* If not specified, document is migrated
                                     to optical disk when committed. If
                                     specified, document is migrated to
                                     optical disk after "migrate_delay"
                                     seconds has elapsed. If value of
                                     migrate_delay is -1, then document is
                                     not migrated to optical disk (unless
                                     DOC_migrate_to_optical_disk is called).
                                     */
    } opt;
} DOC_commit_option_typ;                                     (pg130.23)
```

/* The family commit record is used to specify a remote family on another
system a document will be committed to. For each document committed to
a given local family, it will also be committed to family "rmt_fam_name" on
the remote system specified by "rmt_domain". */

```
typedef struct DOC_family_commit_typ {
    struct DOC_family_commit_typ* next_p;
}                                     (pg131.3)
```


+++ DOC - Document Services +++

```
ASE_domain_name_typ      rmt_domain; /* Name of remote domain */      (pg486.33)
ASE_fam_name_typ         rmt_fam_name; /* Name of remote family */  (pg487.24)
} DOC_family_commit_typ;
```

```
/* Maximum number of DOC_family_commit_typ records allowed for a single
family */
```

```
#define DOC_MAX_COMMIT_RECS      20
```

```
#define DOC_EXPORT_LOG_LEVEL 1
```

```
/* Error function defines: */
```

```
#define DOC_XIN_PROTOCOL      0 /* Defined in this Courier protocol */
```

```
#define DOC_XNOT_IN_PROTOCOL  1 /* Errors not defined there */
```

```
/* Error number defines: */
```

```
/* These error numbers must have the exact same numbers as those
transmitted by the Courier protocol used by this module.
```

```
*/
```

```
#define DOC_zother_error      1
#define DOC_zdocument_not_found 2
#define DOC_zinvalid_family_name 3
#define DOC_zno_permission    4
#define DOC_zinvalid_cache    5
#define DOC_zdocument_already_migrated 6
#define DOC_zinvalid_network_address 7
#define DOC_zinvalid_request_id 8
#define DOC_zinvalid_osar_id   9
#define DOC_zinvalid_image_id  10
#define DOC_zinvalid_security  11
#define DOC_zduplicate_doc_id  12
#define DOC_zduplicate_family   13
#define DOC_zinvalid_user_id    14
#define DOC_zinvalid_session_handle 15
#define DOC_zno_matches         16
#define DOC_ztoo_many_ids_requested 17
#define DOC_zpage_out_of_range  18
#define DOC_znot_in_osar        19
#define DOC_zannotation_too_large 20
#define DOC_zannotation_not_found 21
#define DOC_zno_capability      22
#define DOC_zannotation_busy    23
#define DOC_zinvalid_number_of_osars 24
#define DOC_zinvalid_number_of_surfaces 25
#define DOC_zannotation_not_busy 26
#define DOC_zno_resource        27
#define DOC_zinvalid_family_id  28
#define DOC_zcache_not_local    29
```

+++ DOC - Document Services +++

```
#define DOC_zstatus_not_changeable      30
#define DOC_zbad_cache_to_use          31
#define DOC_ztoo_many_prefetches      32
#define DOC_zservice_not_local        33
#define DOC_znot_impl_pdb             34
#define DOC_zannotation_already_exists 60
#define DOC_zannotation_id_invalid     61
#define DOC_zmagic_rm_id_invalid      62
#define DOC_zparameter_rm_invalid     63
#define DOC_zCFS_annot_doc_not_found   64
#define DOC_zCFS_wrong_annot_log_level 65
#define DOC_zCFS_annot_DN_not_found    66
#define DOC_zCFS_annot_action_failed   67
```

```
/* These error numbers are not able to be transmitted as Courier error numbers
   (i.e. they are not in the protocol specification).
```

```
*/
#define DOC_zbad_version                1
#define DOC_zrpc_error                 2
#define DOC_znot_debugging             3
#define DOC_zconnection_already_open   4
#define DOC_zconnection_not_open       5
#define DOC_zid_wrap_around            6
#define DOC_zbad_attribute_type        7
#define DOC_zbad_document_status       8
#define DOC_zbad_index_value_type      9
#define DOC_zcannot_find_named_service 10
#define DOC_zinvalid_relational_operator 11
#define DOC_zbad_session_type          12
#define DOC_zinvalid_session_number    13
#define DOC_ztoo_many_servers          14
#define DOC_zserver_not_found          15
```

```
/* Constructed error three-tuples: */
#define DOC_err_bad_version \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zbad_version)
/*$M Bad abstract link version when calling DOC. */
#define DOC_err_internal_rpc_error \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zrpc_error)
/*$M Internal rpc error occurred in DOC. */
#define DOC_err_not_debugging \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_znot_debugging)
/*$M Debugging operation requested from non-debugging DOC. */
#define DOC_err_connection_already_open \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zconnection_already_open)
/*$M Connection already open when attempting to open connection with DOC. */
#define DOC_err_connection_not_open \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zconnection_not_open)
```

+++ DOC - Document Services +++

```
/*$M Connection not open when attempting to close connection with DOC. */
#define DOC_err_id_wrap_around \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zid_wrap_around)
/*$M Image IDs have wrapped around in DOC - serious system problem. */
#define DOC_err_bad_attribute_type \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zbad_attribute_type)
/*$M Bad attribute type given to DOC. */
#define DOC_err_bad_document_status \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zbad_document_status)
/*$M Bad document status given to DOC. */
#define DOC_err_bad_index_value_type \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zbad_index_value_type)
/*$M Bad index value type given to DOC. */
#define DOC_err_cannot_find_named_service \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zcannot_find_named_service)
/*$M Cannot find the named DOC service in the Clearinghouse. */
#define DOC_err_invalid_relational_operator \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zinvalid_relational_operator)
/*$M Relational operator invalid for given DOC service. */
#define DOC_err_bad_session_type \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zbad_session_type)
/*$M Bad session type (illegal call to PDB or archive DB?) */
#define DOC_err_invalid_session_number \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zinvalid_session_number)
/*$M Invalid session number for a DOC session - not currently logged in. */
#define DOC_err_too_many_servers \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_ztoo_many_servers)
/*$M An invalid number of servers was given to DOC. */
#define DOC_err_server_not_found \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, DOC_zserver_not_found)
/*$M DOC encountered an unknown server ID - internal software error. */
#define DOC_err_badinterleave \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 35)
/*$M Bad value for interleave count (must be 1, 2, 3, ..., or 8) */
#define DOC_err_interleaveone \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 36)
/*$M The interleave count must be 2 or less when #current surfaces > 1 */
#define DOC_err_tranlog_not_defined \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 37)
/*$M Transaction log family not defined */
#define DOC_err_primary_cant_be_tranlog \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 38)
/*$M A primary family cannot be a transaction log of another family */
#define DOC_err_invalid_osar_disk_type \
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 40)
/*$M The Storage Library disk type is invalid.
The Storage Library disk type is not the correct value for the type of
drives on the Storage Library to be used. */
```

+++ DOC - Document Services +++

```
#define DOC_err_cant_change_is_primary \  
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 41)  
/*$M Can't change the primary/transaction log status of a family */  
#define DOC_err_cant_change_disk_type \  
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 42)  
/*$M Can't change media type on an existing family. */  
#define DOC_err_bad_entry_point \  
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 43)  
/*$M Invalid call to entry point in DOC.  
Invalid entry point number, or invalid parameter (system error).  
A program has not been coded correctly, or has not been compiled with  
the correct version of the DOC abstract */  
#define DOC_err_archive_in_progress \  
    err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 45)  
/*$M Update not allowed--archive in progress */  
#define DOC_err_not_legal \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 46)  
/*$M Requested function is not a legal operation */  
#define DOC_err_batch_already_open \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 47)  
/*$M Batch already open.  
Two batches cannot be opened at the same time on the same DOC logon */  
#define DOC_err_bad_write \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 48)  
/*$M Bad write to cache object  
Unexpected number of bytes written */  
#define DOC_err_no_doc_hdr \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 49)  
/*$M Document header not in last document.  
An attempt has been made to close the batch when the last document is  
not complete (the document header is missing). Either the document  
header must be written, or the batch must be aborted. */  
#define DOC_err_invalid_page_num \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 50)  
/*$M Invalid page number */  
#define DOC_err_invalid_doc_id \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 51)  
/*$M Invalid doc id */  
#define DOC_err_too_many_pages \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 52)  
/*$M Too many pages */  
#define DOC_err_batch_not_open \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 53)  
/*$M Batch not open */  
#define DOC_err_no_more_space \  
    err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 54)  
/*$M No more space in batch */  
#define DOC_err_invalid_index_type \  

```

+++ DOC - Document Services +++

```
err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 55)
/*$M Invalid index type */
#define DOC_err_invalid_num_pages \
err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 56)
/*$M Invalid number of pages */
#define DOC_err_req_cancelled \
err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 57)
/*$M Request cancelled */
#define DOC_err_cant_lock \
err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 58)
/*$M Cannot lock object into an ageable cache */
#define DOC_err_csum_inconsistent \
err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 59)
/*$M If one page has a checksum, all pages must have checksums.
During document committal, if one page has a checksum, all other pages
in the document must also have checksums. */
#define DOC_err_bad_handle \
err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 60)
/*$M Bad session handle passed to DOC */
#define DOC_err_dup_in_fast_batch \
err_encode (err_DOC, DOC_XNOT_IN_PROTOCOL, 61)
/*$M Duplicate write request detected during fast batch committal.
This condition occurs when someone uses CSM_tool to initialize the cache,
but doesn't also delete the write requests. To correct this problem, you
need to delete all write request records for which no object exists in the
cache. Note that WRT_clean can be used to delete all write requests. */
#define DOC_err_interleave_single_side \
err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 62)
/*$M Interleave count must be 1 for single-sided disks. */
#define DOC_err_zero_length_page \
err_encode(err_DOC, DOC_XNOT_IN_PROTOCOL, 63)
/*$M Page length of zero used to write a page. */
#define DOC_err_other_error \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zother_error)
/*$M Error other than standard DOC error reported in protocol - bug in DOC. */
#define DOC_err_document_not_found \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zdocument_not_found)
/*$M Document not found by DOC. */
#define DOC_err_invalid_family_name \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_family_name)
/*$M Invalid family name given to DOC. */
#define DOC_err_no_permission \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zno_permission)
/*$M No permission to perform specified DOC function. */
#define DOC_err_invalid_cache \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_cache)
/*$M Invalid cache name given to DOC. */
#define DOC_err_document_already_migrated \
```

+++ DOC - Document Services +++

```
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zdocument_already_migrated)
/*$M Document already migrated when migration requested of a document by DOC. */
#define DOC_err_invalid_network_address \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_network_address)
/*$M Invalid network address given to DOC. */
#define DOC_err_invalid_request_id \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_request_id)
/*$M Invalid request ID given to DOC. */
#define DOC_err_invalid_osar_id \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_osar_id)
/*$M Invalid Storage Library ID given to DOC. */
#define DOC_err_invalid_image_id \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_image_id)
/*$M Invalid image ID given to DOC. */
#define DOC_err_invalid_security \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_security)
/*$M Invalid security given to DOC. */
#define DOC_err_duplicate_doc_id \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zduplicate_doc_id)
/*$M Duplicate document ID supplied to DOC when committing a document. */
#define DOC_err_duplicate_family \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zduplicate_family)
/*$M Duplicate family name supplied to DOC when creating a family. */
#define DOC_err_invalid_user_id \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_user_id)
/*$M Invalid user ID supplied for logging on to DOC. */
#define DOC_err_invalid_session_handle \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_session_handle)
/*$M DOC given invalid session handle; session probably timed out. */
#define DOC_err_no_matches \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zno_matches)
/*$M No matches found when attempting to find information from DOC. */
#define DOC_err_too_many_ids_requested \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_ztoo_many_ids_requested)
/*$M Too many image IDs requested to be allocated from DOC. */
#define DOC_err_page_out_of_range \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zpage_out_of_range)
/*$M Page number out of the range of pages in a document given to DOC. */
#define DOC_err_annotation_too_large \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zannotation_too_large)
/*$M Too big of an annotation given to DOC. */
#define DOC_err_annotation_not_found \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zannotation_not_found)
/*$M Annotation not found by DOC. */
#define DOC_err_no_capability \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zno_capability)
/*$M No capability for updating the given item through DOC. */
#define DOC_err_annotation_busy \
```

+++ DOC - Document Services +++

```
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zannotation_busy)
/*$M DOC annotation is busy being updated by another client. */
#define DOC_err_invalid_number_of_osars \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_number_of_osars)
/*$M Invalid number of Storage Libraries given to DOC. */
#define DOC_err_invalid_number_of_surfaces \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_number_of_surfaces)
/*$M Invalid number of surfaces given to DOC. */
#define DOC_err_annotation_not_busy \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zannotation_not_busy)
/*$M Annotation not busy when override was requested of DOC. */
#define DOC_err_no_resource \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zno_resource)
/*$M No resources available to log on to DOC. */
#define DOC_err_invalid_family_id \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zinvalid_family_id)
/*$M Invalid family ID given to DOC. */
#define DOC_err_not_in_osar \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_znot_in_osar)
/*$M Synchronous migration from storage media was requested when
operator intervention is required. */
#define DOC_err_status_not_changeable \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zstatus_not_changeable)
/*$M The status of this document is not currently changeable. */
#define DOC_err_cache_not_local \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zcache_not_local)
/*$M Cache must be local to system with DOC service for committal
without migration. */
#define DOC_err_bad_cache_to_use \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zbad_cache_to_use)
/*$M Specified cache is invalid to use with the given DOC function. */
#define DOC_err_too_many_prefetches \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_ztoo_many_prefetches)
/*$M Number of prefetches specified to DOC is beyond the limit. */
#define DOC_err_service_not_local \
err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zservice_not_local)
/*$M Attempt to logon to a DOC service with a service not from that server. */
#define DOC_err_timeout \
err_encode(err_DOC, DOC_XIN_PROTOCOL, 35)
/*$M Timed out waiting for operation to finish. */
#define DOC_err_too_many_queues \
err_encode(err_DOC, DOC_XIN_PROTOCOL, 36)
/*$M Too many DOC queues allocated--cannot allocate additional queues. */
#define DOC_err_too_many_migrates \
err_encode(err_DOC, DOC_XIN_PROTOCOL, 37)
/*$M Too many documents to migrate in single procedure call */
#define DOC_err_cancel_detected \
err_encode(err_DOC, DOC_XIN_PROTOCOL, 38)
```

+++ DOC - Document Services +++

```
/*$M Cancel key detected during synchronous migrate--request cancelled. */
#define DOC_err_qname_too_long \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 39)
/*$M Queue name too long */
#define DOC_err_no_more_queues \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 40)
/*$M Too many queues allocated; no more queues left */
#define DOC_err_queue_not_allocated \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 41)
/*$M Queue not allocated; read not allowed. */
#define DOC_err_queue_inuse \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 42)
/*$M Queue in use; cannot free queue */
#define DOC_err_cant_startup_daemon \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 43)
/*$M Can't startup daemon process */
#define DOC_err_invalid_option \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 44)
/*$M Invalid option */
#define DOC_err_invalid_duration \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 45)
/*$M Duration may not be less than -1 */
#define DOC_err_not_impl_pdb_or_archive \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 46)
/*$M Feature not implemented in archive or PDB document service */
#define DOC_err_too_many_commit_families \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 47)
/*$M Too many entries in family commit list */
#define DOC_err_not_compatible_ssn \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 48)
/*$M SSN of document not compatible with target system.
The system has been configured incorrectly. */
#define DOC_err_not_on_tranlog \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 49)
/*$M Operation requested may not be done on a transaction log family.
Operation is only permitted on a primary family. */
#define DOC_err_circular_commit \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 50)
/*$M Commit list for family <x> can't contain family <x> */
#define DOC_err_no_osar \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 51)
/*$M No Storage Library configured.
Migrate to storage media cannot be done because no Storage Library is
configured on the Storage Library Server being used. */
#define DOC_err_commit_not_remote \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 52)
/*$M Family commit list must specify a remote domain.
When setting up a family so that a document committed to one family will
```


+++ DOC - Document Services +++

```
automatically go to another family, the second family must be a different
domain from the local family (or else a duplicates error can occur). */
#define DOC_err_invalid_surf_id \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 53)
/*$M Invalid surface id. */
#define DOC_err_pgm_err \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 54)
/*$M Internal program error in DOC Service. */
#define DOC_err_no_osar_migrate_off \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 55)
/*$M Migrate is turned off due to the invalid Storage Library Server
id provided. */
#define DOC_err_trace_info \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 56)
/*$M Internal trace output (INFORMATION ONLY). */
#define DOC_err_intervention_required \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 57)
/*$M Operator intervention is required. A device may be disabled.
The migration of the document cannot complete until the device is enabled. */
#define DOC_err_cant_decrease_preflib_curwrtsurfsnum \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, 58)
/*$M Neither can the number of current write surfaces specified to the
family's preferred library be decreased, nor can the preferred
library be removed from a family. */
#define DOC_err_need_disable_surf \
    err_encode(err_DOC, DOC_XIN_PROTOCOL,59)
/*$M Number of current active surfaces will take effect after the current
surfaces become full, or after manually disabling them. */
#define DOC_err_annotation_already_exists \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zannotation_already_exists)
/*$M The requested annotation id already exists */
#define DOC_err_annotation_id_invalid \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zannotation_id_invalid)
/*$M DOC annotation id is invalid. Zero(0) is not a valid annotation id. */

#define DOC_err_magic_rm_id_invalid \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zmagic_rm_id_invalid)
/*$M DOC record management magic id is invalid. */

#define DOC_err_parameter_rm_invalid \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zparameter_rm_invalid)
/*$M DOC record management function parameter is invalid (null or out of range) */

#define DOC_err_CFS_annot_doc_not_found \
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zCFS_annot_doc_not_found)
/*$M CFS-IS Import Agent cannot find the associated annotated document.
This error happens when exporting annotations. Make sure document owns
the annotations is already cataloged on CE. */
```

+++ DOC - Document Services +++

```
#define DOC_err_CFS_wrong_annot_log_level \  
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zCFS_wrong_annot_log_level)  
/*$M Unrecognized CFS-IS Annotation log level is detected.  
Make sure the installed binaries are built and distributed by the compatible  
CFS-IS release version. */
```

```
#define DOC_err_CFS_annot_DN_not_found \  
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zCFS_annot_DN_not_found)  
/*$M CFS-IS Import Agent cannot find the Distinguished Name (DN) for IS.  
LDAP DN is not mapped to specific IS security principal and the associated  
MKF annot_log entry will be deleted from the queue. Refer to the CE  
error log for more details. Then correct the DN mapping error and re-export  
the annotations. */
```

```
#define DOC_err_CFS_annot_action_failed \  
    err_encode(err_DOC, DOC_XIN_PROTOCOL, DOC_zCFS_annot_action_failed)  
/*$M CFS-IS Import Agent encounters errors while processing annotations.  
Refer to the CE error log for more details. Annotations will continue to  
stay in the queue until the problem is resolved. */
```

```
#define DOC_err_not_impl_pdb          DOC_err_not_impl_pdb_or_archive  
#define DOC_err_not_impl_archive     DOC_err_not_impl_pdb_or_archive
```

```
#endif
```

13. DOC SUBROUTINES

13.1. DOC_alloc_queue

```
error_typ  
DOC_alloc_queue (queue_name, queue_num_p)  
    char*          queue_name; /* IN: Ascii string which names the queue  
                               Maxlen=DOC_MAX_QUEUE_NAME_LEN */  
    DOC_queue_num_typ* queue_num_p; /* OUT: queue number assigned */ (pg126.35)  
  
/* Allocates a queue to be used in conjunction with DOC_async_migrate_from_od.  
This queue is only valid on the local station. All callers who specify  
the same name will get the same queue number (provided that noone calls  
DOC_free_queue with this name). A maximum of 32 different queues are  
available.  
  
ERRORS:  
    DOC_err_no_more_queues  
*/
```

+++ DOC - Document Services +++

13.2. DOC_allocate_ids

error_typ

DOC_allocate_ids(number, number_of_ids, first_id_p)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)

unsigned number_of_ids; /* IN: number of image ids requested */

ASE_image_id_typ* first_id_p; /* OUT: First id allocated. */ (pg483.33)

/* Allocates a contiguous range of (per-system) unique image IDs.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_too_many_ids_requested

DOC_err_no_ids_requested

*/

13.3. DOC_async_migrate_from_od

error_typ

DOC_async_migrate_from_od (number, ascending, cache_p, priority, q_num, duration, num_docs, doc_xmit_ary, doc_rcv_ary)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)

bool ascending; /* IN: if ascending order */ (pg491.22)

ASE_service_name_typ* cache_p; /* IN: cache to put docs into, null strings implies put into default cache(s). */ (pg486.28)

DOC_priority_typ priority; /* IN: DOC_PRIO_LOW, DOC_PRIO_MID, or DOC_PRIO_HIGH */ (pg126.30)

DOC_queue_num_typ q_num; /* IN: DOC queue to return status to. This queue should have been allocated with DOC_alloc_queue, or a value of DOC_NO_QUEUE_NUM will skip returning status. */ (pg126.35)

ASE_time_typ duration; /* IN: The length of time to keep the object in the cache. (pg489.31)

Values are:

CSM_LOCKED_DURATION

CSM_PREFETCH_DURATION

CSM_MIGRATE_DURATION

CSM_REFRESH_DURATION

If cache is non-ageable, object will be locked in

+++ DOC - Document Services +++

```
cache and the duration is
ignored. */
unsigned short      num_docs;      /* IN: #docs to migrate, max is
DOC_MAX_ASYNC_MIGRATES */
DOC_async_xmit_doc_typ doc_xmit_ary[]; /* IN: documents to migrate */ (pg127.27)
DOC_async_rcv_doc_typ doc_rcv_ary[]; /* OUT: info about documents
to migrate. */ (pg127.43)
```

/* Migrates a committed document from optical disk to cache. Notifies the client asynchronously by returning status into a DOC queue.

The client should call DOC read queue to find out which migrates have finished. Note that a record will be readable via DOC read queue even if doc_rcv_ary[i].migrate_status == ASE_ALL_MIGRATED or doc_rcv_ary[i].migrate_error != 0. However, if the result of DOC_async migrate from od is an error (non-zero), whether or not a status will be put into the queue is not known, and the application must be able to handle either case (re-issue of command if desired and ignore duplicate status, if present).

No migrate is outstanding upon completion of this routine if:

```
doc_rcv_ary[i].migrate_error != 0 (failure)
-or-
doc_rcv_ary[i].migrate_error == 0 and
doc_rcv_ary[i].migrate_status == ASE_ALL_MIGRATED
(successful completion)
```

If no migrate is outstanding, a request id of DOC_PSEUDO_REQUEST_ID will be returned in doc_rcv_ary[i].request_id, and will also be placed into the queue read by DOC_read_queue.

The parameters in doc_xmit_ary are "doc_id", "ssn", "first_page", "last_page", "dest_doc_id", "dest_ssn", and "notify_data". For each array element, the document specified by "doc_id" and "ssn" is retrieved into the cache, and is renamed to "dest_doc_id" and "dest_ssn" (page number is unchanged). Note that "dest_doc_id" and "dest_ssn" are normally the same as "doc_id" and "ssn", and are only different if the client has a remote cache where objects names must conform to non-standard conventions. The "notify_data" is passed in as tag information, and exactly this information will be returned to the client when the migrate is complete (see description of DOC_read_queue).

If "last_page" in the doc_xmit_ary is either zero or a number higher than the number of pages in the document, "last_page" will be set to the number of pages in the document. If "first_page" is zero, then "first_page" is set to 1. If "first_page" is greater than "last_page" and "last_page" is non-zero, "first_page" and "last_page" are swapped to give

+++ DOC - Document Services +++

an ascending page range. If "first_page" is greater than the number of pages in the document, an error "DOC_err_page_out_of_range" will be returned.

ERRORS:

DOC_err_queue_not_allocated

*/

13.4. DOC_batch_abort

error_typ

DOC_batch_abort (number)

ASE_session_number_typ number;

(pg483.24)

/* Aborts the operations on the batch, deallocates the cache space for the batch.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_batch_not_open

*/

13.5. DOC_batch_close

error_typ

DOC_batch_close (number)

ASE_session_number_typ number; /* IN session number */

(pg483.24)

/* Closes the batch and queues it for processing. Processing will consist of inserting the records into the docs and doctaba tables, and also writing the documents to optical disk.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_batch_not_open

DOC_err_no_doc_hdr

DOC_err_invalid_page_num

*/

13.6. DOC_batch_create

error_typ

DOC_batch_create (number, fam_name, fam_id, batch_size, batch_name)

+++ DOC - Document Services +++

```
ASE_session_number_typ  number;      /* IN  DOC logon handle          */ (pg483.24)
ASE_fam_name_typ        fam_name;     /* IN  Family to write batch to   */ (pg487.24)
ASE_fam_id_typ          fam_id;       /* IN  Family to write batch to if (pg487.25)
                                     fam_name is a null string ("")*/
unsigned                batch_size; /* IN  #bytes in batch when complete*/
char*                   batch_name; /* IN  batch name,
                                     maxlen = DOC_MAX_BATCHNAME_LEN.
                                     The batch name is used to
                                     identify the creator of the
                                     batch. FileNet system software
                                     will create batches with a
                                     prefix of "F_" (see
                                     DOC_SYS_BATCHNAME_PFX), and all
                                     other clients will create
                                     batches with leading characters
                                     other than "F_".
                                     This name does not need to be
                                     unique, but should identify the
                                     creator of the batch. */
```

/* Creates a batch in the cache and leaves batch open. Batch must be closed by calling DOC_batch_abort or DOC_batch_close. Batch will automatically be closed and deleted if you logoff without calling DOC_batch_close, or if the system is rebooted before calling DOC_batch_close.

If duplicate documents are committed using fast batch committal, no errors are reported, and no problems occur. However, a document with the same doc id should have the same page data, and also the same indexes, because it cannot be guaranteed when or if the new duplicate data will replace the old data. Note that duplicates are not reported as an error because this prevents problems from occurring when an application "backs up" too far in order to recover from a system boot or software restart.

Clustering is not supported by DOC_batch_create. All documents are written to the current write surface of the family specified by "fam_name" or "fam_id".

Only 1 batch may be open at a time per DOC_logon handle. If you want more batches open, you must log on to DOC multiple times.

The maximum number of pages in a batch is DOC_MAX_PAGES_PER_BATCH (=10000). The document header counts as a page when computing the maximum. Therefore you can put 5000 1 page documents in a single batch, or 5 documents with 999 pages each, etc.

The batch_size passed in to this routine must be greater than or equal

+++ DOC - Document Services +++

to the number of bytes which the batch will occupy when complete.
The size of a batch with fixed sized documents can be computed
with the formula:

```
doc_hdr_size = DOC_batch_get_csum_doc_hdr_size (...);
document_size = doc_hdr_size;
for (i = 1; i <= num_pages; i++)
    document_size += ROUNDUP (page_size[i], 4);
batch_size = ROUNDUP (document_size, CSM_SECT_SIZE) * num_documents;
batch_size += num_documents * (num_pages+1) * DOC_BATCH_PAGE_OVERHEAD;
```

In the above formula:

"ROUNDUP(x,y)" rounds x up to be the next multiple of y.
"num_pages" is the number of data pages in a document (not including
the document header)
"num_documents" is the number of documents in the batch
"page_size[i]" is the size of the "i"th page of a document, in bytes

The result of the above computation, "batch_size", can be passed in to
DOC_batch_create to reserve the proper amount of space for the batch.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

```
DOC_err_batch_already_open
CSM_no_resources (cache is full, try again later)
```

*/

13.7. DOC_batch_get_csum_doc_hdr_size

error_typ

```
DOC_batch_get_csum_doc_hdr_size (class_name_len, num_pages, num_indexes,
    inx_ary, sec_rd_name_len, sec_wrt_name_len,
    sec_exe_name_len, pages_are_cksmd, doc_hdr_size_p)
short      class_name_len;      /* IN:  strlen of class name      */
short      num_pages;           /* IN:  #pages in doc            */
short      num_indexes;         /* IN:  #indexes(non system only)*/
DOC_inx_info_typ  inx_ary[];    /* IN:  index information        */ (pg130.19)
short      sec_rd_name_len;     /* IN:  strlen of security read  */
short      sec_wrt_name_len;    /* IN:  strlen of security write */
short      sec_exe_name_len;    /* IN:  strlen of security      */
bool       pages_are_cksmd;     /* IN:  if pages are checksummed */ (pg491.22)
long*     doc_hdr_size_p;      /* OUT: size of doc hdr in bytes */
```

+++ DOC - Document Services +++

/* Given the size and types of indexes in a document header along with other information, returns the size (in bytes) of the document header. This size is to be used when making calculations of the size of a batch when calling create batch.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_invalid_index_type

*/

13.8. DOC_batch_get_doc_hdr_size

error_typ

DOC_batch_get_doc_hdr_size (class_name_len, num_pages, num_indexes, inx_ary,
sec_rd_name_len, sec_wrt_name_len,
sec_exe_name_len, doc_hdr_size_p)

short	class_name_len;	/* IN: strlen of class name	*/
short	num_pages;	/* IN: #pages in doc	*/
short	num_indexes;	/* IN: #indexes(non system only)*/	
DOC_inx_info_typ	inx_ary[];	/* IN: index information	*/ (pg130.19)
short	sec_rd_name_len;	/* IN: strlen of security read name	*/
short	sec_wrt_name_len;	/* IN: strlen of security write name	*/
short	sec_exe_name_len;	/* IN: strlen of security execute/append name	*/
long*	doc_hdr_size_p;	/* OUT: size of doc hdr in bytes	*/

/* Given the size and types of indexes in a document header along with other information, returns the size (in bytes) of the document header. This size is to be used when making calculations of the size of a batch when calling create batch.

Note: If you are putting checksums on pages, use DOC_batch_get_csum_doc_hdr_size instead on this routine.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_invalid_index_type

*/

13.9. DOC_batch_set_checksum

+++ DOC - Document Services +++

error_typ

DOC_batch_set_checksum (number, page_num, csum)

ASE_session_number_typ number; /* IN session number */ (pg483.24)
ASE_page_number_typ page_num; /* IN page number to set checksum on */ (pg483.41)
long csum; /* IN checksum of current page */

/* Called prior to calling DOC batch write doc_hdr to set the checksum on pages 1..n of a document, but after calling DOC batch write page for that page. If any page of the batch has a checksum, then all pages of the batch must have checksums.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_csum_inconsistent
DOC_err_invalid_page
DOC_err_batch_not_open

*/

13.10. DOC_batch_write_doc_hdr

error_typ

DOC_batch_write_doc_hdr (number, doc_id, rd_name, wrt_name,
exe_name, doc_class, doc_type,
indexes_num, indexes_p, bytes_rem_p)

ASE_session_number_typ number; /* IN: doc handle */ (pg483.24)
ASE_doc_id_typ doc_id; /* IN: doc id of document */ (pg483.37)
SEC_name_typ rd_name; /* IN: security read name */ (pg371.25)
SEC_name_typ wrt_name; /* IN: security write name */ (pg371.25)
SEC_name_typ exe_name; /* IN: security exe/app name */ (pg371.25)
INX_dcl_name_typ doc_class; /* IN: document class */ (pg239.47)
INX_doc_type_typ doc_type; /* IN: document type */ (pg234.11)
unsigned short indexes_num; /* IN: number of user indexes */
DOC_index_value_typ* indexes_p; /* IN: pointer to user indexes */ (pg124.44)
long* bytes_rem_p; /* OUT: bytes remaining in batch */

/* Writes out header (page 0) for a document of the batch. Must be written after all data pages. Subsequent data pages written after the header will belong in a new document.

It's the responsibility of the caller to insure that the indexes are correct and complete for the document class of this batch. No errors will be flagged for missing or invalid indexes when this batch is processed by DOC, therefore no database related errors will prevent the batch from being successfully processed (i.e. if you change the document class while batches are being processed, no documents will fail to be inserted into the databases, nor will they fail to

+++ DOC - Document Services +++

be written to optical disk). Note that the indexes passed in are the user indexes only; no system indexes need be included.

The document class specified in this call is not used to determine the osar server and optical disk surface the document will be written to. Instead the family specified on the DOC_batch_create call is used, and the family or cluster of the document class is ignored by fast batch committal.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_batch_not_open
DOC_err_too_many_pages
DOC_err_no_more_space

*/

13.11. DOC_batch_write_page

error_typ

DOC_batch_write_page (number, page_num, buf_p, num_bytes, bytes_rem_p)

ASE_session_number_typ number; /* IN: doc handle */ (pg483.24)
ASE_page_number_typ page_num; /* IN: page number of this page */ (pg483.41)
char* buf_p; /* IN: buffer pointer */
long num_bytes; /* IN: #bytes in this call */
long* bytes_rem_p; /* OUT: #bytes space remaining in */
/* batch after write complete */

/* Writes out data for a page of the batch. Used for any page except page zero (the doc_hdr). May be called multiple times to write pages which are too large for a single buffer. However, a single page must be written with consecutive calls to this routine--you may not interleave writes to one page with writes to another.

It's recommended that pages are written in ascending page number order, because this method will be more efficient for retrievals. Writing page 1 last, however, is reasonably efficient for large documents which have page one as a directory.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_batch_not_open
DOC_err_invalid_page_num
DOC_err_no_more_space
DOC_err_too_many_pages

+++ DOC - Document Services +++

DOC_err_zero_length_page

*/

13.12. DOC_cancel_async_migrate

error_typ

DOC_cancel_async_migrate (number, request_id, server_id, local_doc_id)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_request_id_typ	request_id;	/* IN: request id */	(pg486.15)
ASE_server_id_typ	server_id;	/* IN: osar server id */	(pg484.8)
ASE_doc_id_typ	local_doc_id;	/* IN: document id */	(pg483.37)

/* Cancels a migrate request initiated with DOC_async_migrate_from_od. The request_id, server_id, and local_doc_id must come from the doc_rcv_ary on the DOC_async_migrate_from_od call.

ERRORS:

DOC_err_invalid_request_id

*/

13.13. DOC_cancel_migrate

error_typ

DOC_cancel_migrate(number, request_id)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_request_id_typ	request_id;	/* IN: request id */	(pg486.15)

/* Cancels the migration of a document. Because of the asynchronous nature of a migration request, the cancellation cannot be guaranteed, but no harm will come in this case. The request ID is no longer valid after this call.

ERRORS:

DOC_err_invalid_request_id

*/

13.14. DOC_cancel_migrate_from_surf

error_typ

DOC_cancel_migrate_from_surf(number, surf_id, both_sides)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_surface_id_typ	surf_id;	/* IN: surface id */	(pg486.47)
bool	both_sides;	/* IN: cancel both sides of the disk */	(pg491.22)

/* Cancels the migration of documents from the specified surface.

+++ DOC - Document Services +++

Because of the asynchronous nature of a migration request, the cancellation cannot be guaranteed, but no harm will come in this case. The request IDs for retrievals from this surface are no longer valid after this call.

ERRORS:

DOC_err_invalid_surface_id

*/

13.15. DOC_close_connection

error_typ

DOC_close_connection(number)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)

/* Closes a connection which was used with DOC services.

ERRORS:

DOC_err_connection_not_open

*/

13.16. DOC_commit_document

error_typ

DOC_commit_document(number, commit_desc_p, migrate)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)

DOC_committal_desc_typ* commit_desc_p; /* IN: description of document */ (pg126.1)

bool migrate; /* IN: TRUE=>migrate to optical disk, FALSE=>just leave in cache. */ (pg491.22)

/* Commits a document, which causes it to be made available for retrieval. If migrate is TRUE, the document is also queued to be written to optical disk.

Prior to this call being made, element "i" in the "images" array is identified in the cache by: ssn=commit_desc_p->ssn, doc_id=commit_desc_p->images[i], page=ASE_INVALID_PAGE_NUMBER.

After this call is made, element "i" is identified by:

ssn=local_ssn, doc_id=commit_desc_p->images[0], page=i+1.

Note that the ssn is changed to the local system ssn, the doc_id is changed to the document id of the first image in the images array, and the page number is change to array index plus one (first page is 1).

If the document is to be indexed, then Index Services should be called after (not before) this call.

+++ DOC - Document Services +++

See also `DOC_commit_option_doc`.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

`DOC_err_invalid_family_id`
`DOC_err_invalid_cache`
`DOC_err_invalid_image_id`
`DOC_err_invalid_security`
`DOC_err_duplicate_doc_id`
`DOC_err_cache_not_local`
`DOC_err_bad_index_value_type`

*/

13.17. `DOC_commit_option_doc`

`error_typ`

`DOC_commit_option_doc` (`number`, `commit_desc_p`, `num_options`, `options_ary`)
`ASE_session_number_typ` `number`; /* IN: session number */ (pg483.24)
`DOC_committal_desc_typ*` `commit_desc_p`; /* IN: description of document */ (pg126.1)
`unsigned short` `num_options`; /* IN: #options in array below */
`DOC_commit_option_typ` `options_ary[]`; /* IN: array of commit options */ (pg130.40)

/* Commits a document, which causes it to be made available for retrieval.

Prior to this call being made, element "i" in the "images" array is identified in the cache by: `ssn=commit_desc_p->ssn`, `doc_id=commit_desc_p->images[i]`, `page=ASE_INVALID_PAGE_NUMBER`. After this call is made, element "i" is identified by: `ssn=local_ssn`, `doc_id=commit_desc_p->images[0]`, `page=i+1`. Note that the `ssn` is changed to the local system `ssn`, the `doc_id` is changed to the document id of the first image in the images array, and the page number is change to array index plus one (first page is 1).

If the document is to be indexed, then Index Services should be called after (not before) this call.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

`DOC_err_invalid_family_id`
`DOC_err_invalid_cache`
`DOC_err_invalid_image_id`
`DOC_err_invalid_security`
`DOC_err_duplicate_doc_id`
`DOC_err_cache_not_local`
`DOC_err_bad_index_value_type`

+++ DOC - Document Services +++

*/

13.18. DOC_create_annotation

error_typ

DOC_create_annotation(number, doc_id, page, security_p, annot_len, annot_p,
 annot_id_p)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_doc_id_typ	doc_id;	/* IN: doc id of doc to annotate*/	(pg483.37)
ASE_page_number_typ	page;	/* IN: page to annotate */	(pg483.41)
SEC_access_restrictions*	security_p;	/* IN: who can access annotation*/	(pg385.2)
unsigned short	annot_len;	/* IN: length of annotation data*/	
DOC_annotation_typ	annot_p;	/* IN: annotation data */	(pg128.3)
DOC_annotation_id_typ*	annot_id_p;	/* OUT: annotation id assigned to this annotation */	(pg128.1)

/* Annotations are text strings that are logically part of the document,
but are kept physically separate to provide for their independent
creation, update, etc.. Annotations are actually associated with a
document page to facilitate their easy retrieval when a page is displayed.
There can be any number of annotations to a document page.

The security of an annotation is independent of the document that it
is annotating. The caller must have execute-append permission on a
document in order to annotate it.

Annotations are treated as an uninterpreted string of (binary) bytes
by Document Services. The format of an annotation string is specified
elsewhere.

This subroutine creates an annotation to a page of a document.
Execute-append permission on the document is required in order to
annotate it.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_document_not_found
DOC_err_page_out_of_range
DOC_err_annotation_too_large
DOC_err_no_permission
DOC_err_invalid_security

*/

13.19. DOC_create_annotation_id

+++ DOC - Document Services +++

error_typ

DOC_create_annotation_id (number, doc_id, page, security_p, annot_len, annot_p,
annot_id_p)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_doc_id_typ	doc_id;	/* IN: doc id of doc to annotate*/	(pg483.37)
ASE_page_number_typ	page;	/* IN: page to annotate */	(pg483.41)
SEC_access_restrictions	*security_p; /* IN: who can access annotation*/		(pg385.2)
unsigned short	annot_len;	/* IN: length of annotation data*/	
DOC_annotation_typ	annot_p;	/* IN: annotation data */	(pg128.3)
DOC_annotation_id_typ	*annot_id_p; /* IN/OUT: annotation id & SSN */		(pg128.1)

/* WARNING: This entry point should be used with great care and normal entry point (DOC_create_annotation) should be used instead.

This entry point is similiar to DOC_create_annotation() entry point.

The difference is that DOC_create_annotation_id() takes a annot_id_p as input parameter. It allows a client to specify the annot_id. The DOC_annotation_id_typ includes high_order and low_order field. The high_order field is used to indicate the SSN. The low_order field is used to indicate the annotation sequence number.

The annot_id_p as INPUT parameter:

The high_order field or SSN field is ignored or unused. This field is an output field only.

The low_order field is used to indicate the annotation sequence number.

The annot_id_p as OUTPUT parameter:

The high_order field or SSN field is output parameter. The local SSN is always returned.

The low_order field will remain the same as it's input value.

Another difference between these entry points is that the DOC_create_annotation_id() entry point may now return the following new error tuples:

DOC_err_annotation_already_exists - This would indicate that annotation sequence number is already being used.

DOC_err_annotation_id_invalid - This would indicate an invalid annotation ID (zero(0) is passed in.

Annotations are text strings that are logically part of the document, but are kept physically separate to provide for their independent creation, update, etc.. Annotations are actually associated with a document page to facilitate their easy retrieval when a page is displayed.

There can be any number of annotations to a document page. security of an annotation is independent of the document that it

+++ DOC - Document Services +++

is annotating. The caller must have execute-append permission on a document in order to annotate it.

Annotations are treated as an uninterpreted string of (binary) bytes by Document Services. The format of an annotation string is specified elsewhere.

This subroutine creates an annotation to a page of a document. Execute-append permission on the document is required in order to annotate it.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_document_not_found
DOC_err_page_out_of_range
DOC_err_annotation_too_large
DOC_err_no_permission
DOC_err_invalid_security
DOC_err_annotation_already_exists
DOC_err_annotation_id_invalid

*/

13.20. DOC_delete_annotation

error_typ

DOC_delete_annotation(number, doc_id, page, annot_id)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_doc_id_typ	doc_id;	/* IN: doc id */	(pg483.37)
ASE_page_number_typ	page;	/* IN: page */	(pg483.41)
DOC_annotation_id_typ	annot_id;	/* IN: annotation id */	(pg128.1)

/* Deletes an annotation.
Both write access and execute-append access to a document are required in order to delete an annotation.
The deletion will also fail if another client has the annotation locked.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_document_not_found
DOC_err_annotation_not_found
DOC_err_annotation_busy
DOC_err_no_permission

*/

+++ DOC - Document Services +++

13.21. DOC_delete_documents

```
error_typ
DOC_delete_documents(number, doc_ids_num, doc_ids)
  ASE_session_number_typ  number;          /* IN:  session number */          (pg483.24)
  unsigned                doc_ids_num;     /* IN:  number of docs to delete */
  ASE_doc_id_typ*        doc_ids;         /* IN:  array of doc_ids */       (pg483.37)
```

/* Deletes a list of documents.
The documents are deleted from the locator database and the default cache if the document was committed with migrate=FALSE (does not delete document from default cache if migrate=TRUE--lets document age out of cache). All annotations for the documents are deleted as well.

If any document in the list is not deletable, then no docs are deleted.

If the list contains documents which don't exist and no errors prevent deletion of any existing document, a DOC_err_document_not_found error is returned and all existing documents in the list are deleted.

If the documents are in the Index Database, then Index Services should be called to delete the docs from the Index Database before (not after) calling this subroutine.

Both write access and execute-append access to a document and its annotations are required in order to delete a document.

This subroutine is not supported for archive databases (but is supported for PDBS).

ERRORS:
 DOC_err_document_not_found
 DOC_err_no_permission

*/

13.22. DOC_find_by_family_id

```
error_typ
DOC_find_by_family_id(number, family_id, rel_op, max_matches,
  actual_matches_p, family_descs_p, last_match_p)
  ASE_session_number_typ  number;          /* IN:  session number */          (pg483.24)
  INX_fam_id_typ         family_id;       /* IN:  family id */              (pg243.25)
  ASE_relational_op_typ  rel_op;          /* IN:  GTR, GEQ, EQL */          (pg484.34)
  unsigned               max_matches;     /* IN:  max #family recs to
  return */
  unsigned short*       actual_matches_p; /* OUT: #family recs returned*/
  DOC_family_desc_typ*  family_descs_p;  /* OUT: array of family recs */ (pg124.9)
```

+++ DOC - Document Services +++

```
bool*          last_match_p;          /* OUT: TRUE=>no more recs for (pg491.22)
                                         query */
/* Finds all family descriptions matching a query specification by family ID.

This subroutine is not supported for archive databases or PDBs.

ERRORS:
  DOC_err_no_matches
  DOC_err_invalid_relational_operator
*/
```

13.23. DOC_find_by_family_name

```
error_typ
DOC_find_by_family_name(number, family_name, rel_op, max_matches,
                        actual_matches_p, family_descs_p, last_match_p)
ASE_session_number_typ  number;          /* IN: session number */      (pg483.24)
INX_fam_name_typ        family_name;     /* IN: family name */      (pg243.24)
ASE_relational_op_typ   rel_op;          /* IN: GTR, GEQ, EQL */    (pg484.34)
unsigned                max_matches;     /* IN: max #family recs to
                                         return. */
unsigned short*        actual_matches_p; /* OUT: #family recs returned*/
DOC_family_desc_typ*   family_descs_p;   /* OUT: array of family recs */ (pg124.9)
bool*                  last_match_p;     /* OUT: TRUE=>no more records (pg491.22)
                                         for query. */

/* Finds all family descriptions matching a query spec by family name.

This subroutine is not supported for archive databases or PDBs.

ERRORS:
  DOC_err_no_matches
  DOC_err_invalid_relational_operator
*/
```

13.24. DOC_find_documents

```
error_typ
DOC_find_documents(number, ssn, doc_id, rel_op, max_matches, actual_matches_p,
                  doc_loc_descs_p, last_match_p)
ASE_session_number_typ  number;          /* IN: session number      */ (pg483.24)
ASE_ssn_typ             ssn;            /* IN: ssn of start doc    */ (pg483.47)
ASE_doc_id_typ          doc_id;         /* IN: doc_id of start doc */ (pg483.37)
ASE_relational_op_typ   rel_op;         /* IN: EQL, GTR, or GEQ   */ (pg484.34)
unsigned                max_matches;     /* IN: max number of docs
                                         to return (note: the
```


+++ DOC - Document Services +++

ERRORS:

DOC_err_no_matches (Returns this if no matches found or if
user has no permission to access document)
DOC_err_invalid_relational_operator

*/

13.26. DOC_find_surfaces

error_typ

DOC_find_surfaces(number, ssn, surf_id, rel_op, max_matches, actual_matches_p,
surface_descs_p, last_match_p)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_ssn_typ	ssn;	/* IN: starting ssn */	(pg483.47)
ASE_surface_id_typ	surf_id;	/* IN: starting surface id */	(pg486.47)
ASE_relational_op_typ	rel_op;	/* IN: GTR, GEQ or EQL */	(pg484.34)
unsigned	max_matches;	/* IN: max #surfs to return */	
unsigned short*	actual_matches_p;	/* OUT: #surfaces returned */	
DOC_surface_desc_typ*	surface_descs_p;	/* OUT: surface descriptions */	(pg122.37)
bool*	last_match_p;	/* OUT: if last doc for query returned */	(pg491.22)

/* Finds all surface descriptions matching a query specification.
The comparison is by SSN and surface ID.

This subroutine is not supported for PDBs or archive databases.

ERRORS:

DOC_err_no_matches (Returns this if no matches found)
DOC_err_invalid_relational_operator

*/

13.27. DOC_free_queue

error_typ

DOC_free_queue (queue_num)

DOC_queue_num_typ queue_num; /* Number of queue returned. */ (pg126.35)

/* Returns a queue allocated by DOC_alloc_queue. Note that if someone
reallocates this queue with a call to DOC_alloc_queue, the queue will
probably be assigned a new number.

ERRORS:

DOC_err_queue_inuse
DOC_err_queue_not_allocated

*/

+++ DOC - Document Services +++

13.28. DOC_get_and_lock_annotation

error_typ

```
DOC_get_and_lock_annotation(number, doc_id, page, annot_id, override,
                             security_p, annot_len_p, annot_p, cap)
ASE_session_number_typ  number;          /* IN: session number */      (pg483.24)
ASE_doc_id_typ          doc_id;          /* IN: document id */      (pg483.37)
ASE_page_number_typ    page;            /* IN: page */              (pg483.41)
DOC_annotation_id_typ  annot_id;        /* IN: annotation id */    (pg128.1)
bool                   override;        /* IN: TRUE=>override someone (pg491.22)
                             else's lock */
SEC_access_restrictions *security_p;    /* OUT: security of annotation */ (pg385.2)
unsigned short         *annot_len_p;    /* OUT: length of annot data */
DOC_annotation_typ     annot_p;         /* OUT: annotation data. */    (pg128.3)
ASE_capability_typ     cap;             /* OUT: capability used on (pg484.19)
                             subsequent update. */
```

/* Gets and locks an annotation to a page of a document.
The locking only prevents other clients from locking the annotation
at the same time (except when they override the previous lock).
This subroutine should be used prior to modifying an annotation.
The annotation ID, if not already known, should be obtained with a call
to DOC_get_annotations.
Since the annotation may have changed since the last time it was examined,
the annotation itself is returned from this call.
The procedure DOC_update_annotation should be called after calling
this subroutine even if the annotation is not changed.
Read and write access to the document is required in order to get and lock
the annotation.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

```
DOC_err_document_not_found
DOC_err_annotation_not_found
DOC_err_annotation_busy - if no override and already locked
DOC_err_annotation_not_busy - if override and not locked; can be
                             recovered from by setting override false
                             and retrying
DOC_err_no_permission
```

*/

13.29. DOC_get_annotations

error_typ

```
DOC_get_annotations(number, doc_id, page, data, iter_p, cleanup_p)
ASE_session_number_typ  number;          /* IN: session number */      (pg483.24)
```

+++ DOC - Document Services +++

```
ASE_doc_id_typ      doc_id;      /* IN: document id */      (pg483.37)
ASE_page_number_typ page;        /* IN: page */            (pg483.41)
any_ptr             data;        /* IN: passed in to *iter_p and
                                *cleanup_p */          (pg491.24)
error_typ           (*iter_p)(); /* IN: callback routine which
                                is given annotation data*/ (pg493.26)
error_typ           (*cleanup_p)(); /* IN: callback routine for
                                cleanup purposes. */      (pg493.26)
```

/* Gets all the annotations to a page of a document. Read access to the document is required in order to get any annotations for it. If an annotation is currently locked, then the old value for that annotation is returned. If no annotations exist for the specified document and page, this routine returns success (zero) and does not call the iter_p routine to pass annotation data to the caller.

iter_p - a function to call for each annotation found; this function has the following description:

```
error_typ
iter(data, desc_p)
any_ptr             data;
DOC_annotation_desc_typ *desc_p;
```

The parameter descriptions are:

data - the data passed to DOC_get_annotations

annot_desc_p - a pointer to the annotation descriptor;

this data must be copied, it can be stepped on

Reasonable functions for *iter_p to perform are: allocate memory, copy the data, fill in some data structures, and link a structure into a list.

cleanup_p - a function to call to cleanup from all calls to *iter_p (in this invocation of DOC_get_annotations); this function has the following description:

```
error_typ
cleanup(data)
any_ptr             data;
```

The parameter descriptions are:

data - the data passed to DOC_get_annotations

Reasonable functions for *cleanup_p to perform are:

deallocate all memory allocated by *iter_p and modify some data structures.

cleanup_p is called once to cleanup after an error for all calls to *iter_p in this invocation of DOC_get_annotations. cleanup_p will only be called if DOC_get_annotations gets an error AND *iter_p has been called at least once during the current get annotations call.

Note that both functions must be fast and non-buggy (therefore small) because they can be called in a critical section. In particular, they

+++ DOC - Document Services +++

is significant, so `check_cache` should be passed in as `FALSE` when this information is not needed. `Check_cache = FALSE` will cause the `"location_ary[x].in_cache"` field to not be set.

If `"last_page"` in the `page_range_ary` is either zero or a number higher than the number of pages in the document, `"last_page"` will be set to the number of pages in the document. If `"first_page"` is zero, then `"first_page"` is set to 1. If `"first_page"` is greater than `"last_page"` and `"last_page"` is non-zero, `"first_page"` and `"last_page"` are swapped to give an ascending page range. If `"first_page"` is greater than the number of pages in the document, an error `"DOC_err_page_out_of_range"` will be returned.

Not supported for PDBs (supported for archive databases).

ERRORS:

`DOC_err_page_out_of_range`

*/

13.32. `DOC_get_ssn_of_docs`

`error_typ`

`DOC_get_ssn_of_docs(number, ssn_p)`

`ASE_session_number_typ` `number;` /* IN: session number */ (pg483.24)

`ASE_ssn_typ*` `ssn_p;` /* OUT: ssn associated with this document service */ (pg483.47)

/* Returns the ssn of the documents in the document service.

ERRORS:

none other than network errors.

*/

13.33. `DOC_is_annotated`

`error_typ`

`DOC_is_annotated(number, doc_id, annotated_p)`

`ASE_session_number_typ` `number;` /* IN: session number */ (pg483.24)

`ASE_doc_id_typ` `doc_id;` /* IN: document id */ (pg483.37)

`bool` `*annotated_p;` /* OUT: TRUE=>doc is annotated */ (pg491.22)

/* Returns an indication as to whether or not a given document is annotated.

Not supported for PDBs (is supported for archive databases).

ERRORS:

`DOC_err_document_not_found`

+++ DOC - Document Services +++

DOC_err_no_permission

*/

13.34. DOC_is_migrated

error_typ

DOC_is_migrated(number, request_id, timeout_val, migrated_p)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)

ASE_request_id_typ request_id; /* IN: request id */ (pg486.15)

unsigned timeout_val; /* IN: time to wait for migrate
to complete, in seconds. */

bool* migrated_p; /* OUT: TRUE=>migrate complete, (pg491.22)
FALSE otherwise. */

/* Checks the migration status of a document.

If timeout_val (expressed in seconds) is non-zero, then the caller will be blocked until the ready is complete (or an error occurred), or the timeout expired, whichever comes first. If timeout_val is zero, this will not block the caller and will just return the current state of the ready.

This subroutine should be called repetitively (possibly checking for an operator requested abort between calls) until *migrated_p is TRUE, or until DOC_cancel_migrate is called. The time interval between DOC_is_migrated calls should not exceed 5 seconds. If the application does not call DOC_is_migrated for an active migrate for over 5 seconds, the osar server may discard the migrate completion status in which case the DOC_is_migrated will never return TRUE.

This subroutine is never used for PDBs, because all PDB documents will be in the cache.

ERRORS:

DOC_err_invalid_request_id

*/

13.35. DOC_logoff

error_typ

DOC_logoff (number)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)

/* Logs off from DOC services. This must be done after all usage of a session.

ERRORS:

None other than network errors.

+++ DOC - Document Services +++

*/

13.36. DOC_logon

error_typ

DOC_logon (service_name_p, name_type, leave_open, number_p, timeout_p)

ASE_service_name_typ* service_name_p; /* IN: name of service */ (pg486.28)

ASE_nch_name_type_typ name_type; /* IN: ASE_IMS_NAME or ASE_SVC_NAME */ (pg483.28)

bool leave_open; /* IN: leave connection open */ (pg491.22)

ASE_session_number_typ* number_p; /* OUT: session number */ (pg483.24)

unsigned* timeout_p; /* OUT: obsolete--may be NULL */

/* Logs on to DOC services.

This must be done by each process before doing any work with this service.

Note: The session number returned by this subroutine is usable only by one process at a time, which means that abstracts and multi-process applications must be careful.

ERRORS:

DOC_err_cannot_find_named_service

*/

13.37. DOC_migrate_from_optical_disk

error_typ

DOC_migrate_from_optical_disk(number, page_range_p, cache_p, order, notify, status_p, pages_available_p, request_id_p, pages_in_doc_p, actual_cache_p)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)

ASE_page_range_typ* page_range_p; /* IN: pages of document to migrate. First_page and last_page equal to zero migrates entire document */ (pg484.42)

ASE_service_name_typ* cache_p; /* IN: the cache name of the cache that the doc will reside in; if this name is null, then a default cache will be used. */ (pg486.28)

DOC_migrate_order_typ order; /* IN: DOC_EXACT_ASCENDING, DOC_EXACT_DESCENDING, DOC_ASCENDING, or DOC_DESCENDING */ (pg126.20)

ASE_notify_option_typ notify; /* IN: ASE_NOTIFY_SYNCHRONOUS, (pg486.10)

+++ DOC - Document Services +++

```

ASE_migrate_status_typ* status_p;          /* OUT: ASE_NOTIFY_ASYNCHRONOUS
                                           or ASE_NOTIFY_NONE. */
                                           (pg487.12)
                                           ASE_ALL_MIGRATED,
                                           ASE_IN_DRIVE,
                                           ASE_IN_SLOT or
                                           ASE_INTERVENTION_REQ */
ASE_page_number_typ*    pages_available_p; /* OUT: #pages already in
                                           cache before this
                                           migrate issued. */
                                           (pg483.41)
ASE_request_id_typ*     request_id_p;      /* OUT: Request id used to wait
                                           for a completion of an
                                           asynchronous migrate.
                                           Zero if no waiting
                                           allowed due to
                                           *status_p being equal
                                           ASE_ALL_MIGRATED, or
                                           error. */
                                           (pg486.15)
ASE_page_number_typ*    pages_in_doc_p;    /* OUT: #pages in document */
                                           (pg483.41)
ASE_service_name_typ*  actual_cache_p;    /* OUT: cache document put
                                           into. Same as *cache_p
                                           if cache explicitly
                                           named. */
                                           (pg486.28)
```

/* Migrates a committed document from optical disk to a cache.
After the migrate is complete, the caller can call cache services
to read the migrated pages.

If the cache the document is migrated into is a retrieval (ageable)
cache, then the pages are not guaranteed to remain in the cache,
so the client may need to call this subroutine again. If the
cache is not a retrieval cache, then the pages will stay in the cache
until explicitly deleted.

Asynchronous migrates (notify=ASE_NOTIFY_ASYNCHRONOUS) must be either
cancelled or waited on until complete. The *request_id_p is non-zero
whenever a migrate must be waited on or cancelled, and failure to
wait/cancel this request can result in a running out of workstation
memory (memory is used to save the migrate status) and also poor server
performance.

When doing asynchronous migrates the application program must not
wait more than 5 seconds between DOC_is_migrated calls, or else the
migrate status may be dropped by the server and the migrate may
never complete. The intended use of asynchronous migrate is to allow
the application program to periodically check for an operator requested
abort between calls to DOC_is_migrated.

If "last_page" in the page range is either zero or a number higher

+++ DOC - Document Services +++

than the number of pages in the document, "last_page" will be set to the number of pages in the document. If "first_page" is zero, then "first_page" is set to 1. If "first_page" is greater than "last_page" and "last_page" is non-zero, "first_page" and "last_page" are swapped to give an ascending page range. If "first_page" is greater than the number of pages in the document, an error "DOC_err_page_out_of_range" will be returned.

ERRORS:

DOC_err_document_not_found
DOC_err_invalid_cache
DOC_err_page_out_of_range
DOC_err_invalid_network_address

*/

13.38. DOC_migrate_to_optical_disk

error_typ

DOC_migrate_to_optical_disk(number, doc_id, cache_p)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)
ASE_doc_id_typ doc_id; /* IN: doc id of doc to migrate */ (pg483.37)
ASE_service_name_typ* cache_p; /* IN: obsolete (pass in NULL) */ (pg486.28)

/* Migrates a committed document from a cache to optical disk.
The entire document must be in the specified cache.
The migration is done asynchronously to the call.
The document must not be previously migrated, and will
be written to the family and/or cluster dictated by the
DOC_committal_desc provided when the document was committed.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_document_not_found
DOC_err_invalid_cache
DOC_err_document_already_migrated

*/

13.39. DOC_migrate_w_ssn_from_od

error_typ

DOC_migrate_w_ssn_from_od(number, page_range_p, cache_p, order,
notify, status_p, pages_available_p, request_id_p,
pages_in_doc_p, actual_cache_p, local_ssn_p, local_doc_id_p)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)
ASE_page_range_w_ssn_typ* page_range_p; /* IN: pages of document to (pg485.2)

+++ DOC - Document Services +++

```

migrate.  First_page
and last_page equal to
zero migrates
entire document */
ASE_service_name_typ*  cache_p;      /* IN: the cache name of the (pg486.28)
cache that the doc
will reside in;
if this name is null,
then a default cache
will be used. */
DOC_migrate_order_typ  order;      /* IN: DOC_EXACT_ASCENDING, (pg126.20)
DOC_EXACT_DESCENDING,
DOC_ASCENDING,
or DOC_DESCENDING */
ASE_notify_option_typ  notify;    /* IN: ASE_NOTIFY_SYNCHRONOUS, (pg486.10)
ASE_NOTIFY_ASYNCHRONOUS
or ASE_NOTIFY_NONE. */
ASE_migrate_status_typ*  status_p;  /* OUT: ASE_ALL_MIGRATED, (pg487.12)
ASE_IN_DRIVE,
ASE_IN_SLOT or
ASE_INTERVENTION_REQ */
ASE_page_number_typ*    pages_available_p; /* OUT: #pages already in (pg483.41)
cache before this
migrate issued. */
ASE_request_id_typ*     request_id_p; /* OUT: Request id used to wait (pg486.15)
for a completion of an
asynchronous migrate.
Zero if no waiting
allowed due to
*status_p being equal
ASE_ALL_MIGRATED, or
error. */
ASE_page_number_typ*    pages_in_doc_p; /* OUT: #pages in document */ (pg483.41)
ASE_service_name_typ*   actual_cache_p; /* OUT: cache document put (pg486.28)
into. Same as *cache_p
if cache explicitly
named. */
ASE_ssn_typ*            local_ssn_p;   /* OUT: local ssn of document */ (pg483.47)
ASE_doc_id_typ*         local_doc_id_p; /* OUT: local document id */ (pg483.37)

```

/* Migrates a committed document from optical disk to a cache.
After the migrate is complete, the caller can call cache services
to read the migrated pages. When calling cache services, the document
can be found with (ssn, doc_id) equal to (*local_ssn_p, *local_doc_id_p).
Note that the local ssn/doc id may be different from the ssn/doc_id
passed in, so the local ssn/doc id should always be used for subsequent
cache services calls.

+++ DOC - Document Services +++

If the cache the document is migrated into is a retrieval (ageable) cache, then the pages are not guaranteed to remain in the cache, so the client may need to call this subroutine again. If the cache is not a retrieval cache, then the pages will stay in the cache until explicitly deleted.

Asynchronous migrates (notify=ASE_NOTIFY_ASYNCHRONOUS) must be either cancelled or waited on until complete. The *request_id_p is non-zero whenever a migrate must be waited on or cancelled, and failure to wait/cancel this request can result in a running out of workstation memory (memory is used to save the migrate status) and also poor server performance.

When doing asynchronous migrates the application program must not wait more than 5 seconds between DOC_is_migrated calls, or else the migrate status may be dropped by the server and the migrate may never complete. The intended use of asynchronous migrate is to allow the application program to periodically check for an operator requested abort between calls to DOC_is_migrated.

If "last_page" in the page range is either zero or a number higher than the number of pages in the document, "last_page" will be set to the number of pages in the document. If "first_page" is zero, then "first_page" is set to 1. If "first_page" is greater than "last_page" and "last_page" is non-zero, "first_page" and "last_page" are swapped to give an ascending page range. If "first_page" is greater than the number of pages in the document, an error "DOC_err_page_out_of_range" will be returned.

ERRORS:

DOC_err_document_not_found
DOC_err_invalid_cache
DOC_err_page_out_of_range
DOC_err_invalid_network_address

*/

13.40. DOC_modify_document_attribute

error_typ

DOC_modify_document_attribute(number, doc_id, attr_p)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)
ASE_doc_id_typ doc_id; /* IN: document to modify */ (pg483.37)
DOC_doc_attribute_value_typ *attr_p; /* IN: attribute to modify */ (pg129.4)

/* Modifies the attributes of an existing document.
Write access is required on the document to modify any of its attributes.
Only some attributes of the document are modifiable, the other attributes

+++ DOC - Document Services +++

are read-only to clients.

This does not affect any current users of the document.

There is no checking to ensure that another user has changed any of these attributes between the time that the client read the attribute and the time that the client modified the attribute.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_invalid_security
DOC_err_no_permission
DOC_err_bad_attribute_type
DOC_err_bad_document_status
DOC_err_status_not_changeable

*/

13.41. DOC_open_connection

error_typ

DOC_open_connection(number)

ASE_session_number_typ number; /* IN: Session number */ (pg483.24)

/* Opens a connection for use with DOC services.
Connections should not be left open for long periods (e.g. minutes).

ERRORS:

DOC_err_connection_already_open (you may not open a connection
which is already open).

*/

13.42. DOC_prefetch_from_optical_disk

error_typ

DOC_prefetch_from_optical_disk(number, page_ranges_num, page_ranges_p,
cache_p, order, data, iter_p, cleanup_p)

ASE_session_number_typ number; /* IN: session number */ (pg483.24)

unsigned page_ranges_num; /* IN: #elements in page_ranges_p
array. */

ASE_page_range_typ *page_ranges_p; /* IN: docs/pages array */ (pg484.42)

ASE_service_name_typ *cache_p; /* IN: cache to put docs into, or
null strings for default
cache. */ (pg486.28)

DOC_migrate_order_typ order; /* IN: DOC_EXACT_ASCENDING,
DOC_EXACT_DESCENDING,
DOC_ASCENDING, or
DOC_DESCENDING */ (pg126.20)

+++ DOC - Document Services +++

```
any_ptr          data;          /* IN:  parameter passed through to (pg491.24)
                                "iter_p" and "cleanup_p" */
error_typ        (*iter_p)();    /* IN:  callback procedure */      (pg493.26)
error_typ        (*cleanup_p)(); /* IN:  callback procedure */      (pg493.26)
```

/* Prefetches documents from optical disk.

This is used if the client anticipates requiring the documents in the near future. The prefetch is done at a lower priority than the migration initiated by `DOC_migrate_from_optical_disk`.

Prefetching a document is strictly optional and does not replace the need for calling `DOC_migrate_from_optical_disk`.

There is no guarantee that the prefetch will ever be done; on a busy system prefetches are unlikely to be completed.

One prefetch call with a large list is better than several prefetch calls with small lists, as it will enable the optical disk selection algorithm to better minimize the total amount of disk swapping that will go on.

The maximum number of documents to prefetch is 1000.

If an error occurred on prefetching an individual document, then a prefetch will still be attempted on the other documents.

If a cache is not specified in the input parameters, the default will be the cache associated with the storage device the document resides on.

"iter_p" and "cleanup_p" are procedures passed in to this subroutine for the purpose of returning the cache each document is prefetched into, because there may be more than one default cache used when `*cache_p` is passed in as null strings. Both `iter_p` and `cleanup_p` may be input as null if you don't need the name of the destination cache.

`iter_p` is called once for each page range input. It's description follows:

```
error_typ
iter (data, err, pages_in_doc, actual_cache_p)
    any_ptr          data;
    error_typ        err;
    unsigned         pages_in_doc;
    ASE_service_name_typ *actual_cache_p;
```

*iter_p's parameters are:

data - the data passed to `DOC_prefetch_from_optical_disk`
err - any error found while prefetching this particular page range
pages_in_doc - the actual number of pages in the document
(useful if a request is made to prefetch all pages in the document)
actual_cache_p - pointer to the name of the actual cache for the corresponding page range (the calls to

+++ DOC - Document Services +++

*iter_p are made in the order of the ranges passed to DOC_prefetch_from_optical_disk); this cache name must be used to find the pages if they were migrated to a default cache; this data must be copied, it can be stepped on

cleanup_p is called once to cleanup after an error for all calls to *iter_p in this invocation of DOC_prefetch_from_optical_disk. cleanup_p will only be called if DOC_prefetch_from_optical_disk gets an error AND *iter_p has been called at least once during the current prefetch call.

cleanup_p has the following description:

```
error_typ
cleanup (data)
    any_ptr          data;
```

*cleanup_p's parameters are:

data - the data passed to DOC_prefetch_from_optical_disk

If "last_page" in the page range is either zero or a number higher than the number of pages in the document, "last_page" will be set to the number of pages in the document. If "first_page" is zero, then "first_page" is set to 1. If "first_page" is greater than "last_page" and "last_page" is non-zero, "first_page" and "last_page" are swapped to give an ascending page range. If "first_page" is greater than the number of pages in the document, an error "DOC_err_page_out_of_range" will be returned.

ERRORS:

```
DOC_err_document_not_found
DOC_err_invalid_cache
DOC_err_page_out_of_range
DOC_err_bad_cache_to_use
DOC_err_too_many_prefetches
```

*/

13.43. DOC_read_queue

```
error_typ
DOC_read_queue (queue_num, timeout_val, data_p)
    DOC_queue_num_typ  queue_num;      /* IN: queue to read from */      (pg126.35)
    long               timeout_val;     /* IN: timeout in 1/100ths seconds */
    DOC_async_comp_typ* data_p;         /* OUT: data about migrate which has (pg127.15)
```

+++ DOC - Document Services +++

finished. */

/* Gets information about the most recently completed migrate request initiated by DOC_async_migrate_from_od. The order in which migrates will complete is not necessarily the same as the order in which they were issued.

ERRORS:

DOC_err_timeout
DOC_err_queue_not_allocated

*/

13.44. DOC_set_surface_inbox_priority

error_typ

DOC_set_surface_inbox_priority (number, surface_id, surf_prio)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_surface_id_typ	surface_id;	/* IN: surface id */	(pg486.47)
DOC_inbox_priority_typ	surf_prio;	/* IN: surface priority */	(pg119.22)

/* This routine controls how long the surface will remain in an osar when an osar is full. The maximum value is DOC_MAX_INBOX_PRIORITY (255), the minimum is 0, and the default for disks which have not been modified with this subroutine is DOC_DEFAULT_INBOX_PRIORITY (128).

The algorithm works as follows:

When a new disk is inserted into a full osar, a different disk must be ejected to make room in the slot rack. The disk ejected is chosen by finding the set of disks with the lowest value for the inbox priority, and then taking the disk from this group which was least recently used.

This subroutine is not supported for archive databases or PDBs.

*/

13.45. DOC_update_annotation

error_typ

DOC_update_annotation(number, cap, security_p, annot_len, annot_p)

ASE_session_number_typ	number;	/* IN: session number */	(pg483.24)
ASE_capability_typ	cap;	/* IN: capability from getnlock */	(pg484.19)
SEC_access_restrictions	*security_p;	/* IN: new security info */	(pg385.2)
unsigned short	annot_len;	/* IN: length of annot (below) */	
DOC_annotation_typ	annot_p;	/* IN: annotation data */	(pg128.3)

+++ DOC - Document Services +++

/* Updates an annotation with new data (replaces an existing annotation).
Write access to the document is required to update an annotation.

This subroutine is not supported for archive databases or PDBs.

ERRORS:

DOC_err_annotation_too_large
DOC_err_document_not_found
DOC_err_annotation_not_found
DOC_err_no_capability
DOC_err_no_permission
DOC_err_invalid_security

*/

14. DTI DECLARATIONS

=====

DTI - Date Time Interface

=====

This interface allows an application to convert dates and times from one format to another.

The DTI interface is no longer the preferred method to convert dates and times to string formats, however, since this function is now provided by the native operating system environment on all supported platforms. The DTI routines to convert between strings and longwords, and strings and structures are provided for backward compatibility only. You should **not** use them in any new development.

On POSIX compliant systems the `strftime` and `strptime` functions should be used to convert between longword times and strings; while `localtime`, `gmtime`, and `mktime` should be used to convert between longword times and the `tm` time structure. These functions use the POSIX LOCALE to obtain the correct formats for converting times to strings.

The native conversion routines on some OS's support conversions between times and UNICODE strings. DTI does not now, and never will support UNICODE. The masks supported by DTI are a subset of the masks provided by POSIX to format date and time strings.

One function of DTI which you will have to continue using, however, is the conversion between the FileNet longword date type, and the POSIX longword time type. The FileNet date type is proprietary, and there is no other way to convert it to a standard time type (see details of FileNet date type below.)

On non-POSIX compliant systems, DTI retrieves the default date/time strings such as the names of the month, date/time mask, etc. from the system referenced by the "sess_num" argument on each call. The "sess_num" parameter must be an IMS handle returned by the IMS_logon procedure.

On POSIX compliant systems, DTI gets the default date/time strings from the native OS's environment. When the native environment is configured to use a date/time mask which DTI cannot support, DTI will use a hardcoded, internal default, and an error will be logged in this case. Note that the "sess_num" argument is not used on POSIX compliant systems, so therefore `sess_num` may

+++ DTI - Date Time Interface +++

be passed in as NULL.

The formats for dates and times can be any of the following:

- longword time
- longword date
- string
- structure

A longword time (`ASE_time_typ`) is the number of seconds from midnight on a certain date. A longword time can be negative or positive to indicate times before or after this date. Note that this longword specifies both a date and a time of day on that date.

A longword date (`ASE_date_typ`) is the number of days since a certain date. The longword can be negative or positive to indicate days before or after this date. The granularity of a date is to a day, therefore a time of day cannot be computed from a longword date. The longword date, however, has greater range than a longword time and can specify points farther into the future or past.

Both longword dates and times are defined to be Greenwich Mean Time (GMT) relative. DTI does not support a local time version of either longword dates or times.

A string is an ascii string in various formats which indicate either the date or date and time. The string can be in either GMT or local time, and will have a mask (`DTI_mask_typ`) associated with it to indicate the string format. For example, a mask can be "yy/mm/dd hh:tt:ss" for date and time, or just "yy/mm/dd" for less granularity.

A date/time structure (`DTI_tm_struct_typ`) is a C record which has the various components of a date and time represented as integers. This structure can be in either GMT or local time.

DECLARATIONS:

```
#ifndef DTI_defs
#define DTI_defs
```

```
#ifndef FileNet_h
#include <FileNet.h>
#endif
#ifndef AS_externals_h
#include <AS_externals.h>
#endif
```

```
/* Type which indicates whether the time zone is local or Greenwich Mean
```

+++ DTI - Date Time Interface +++

```
Time (GMT) */

typedef char DTI_zone_typ;
#define DTI_LOCAL          1
#define DTI_GMT            2

/* DTI_datetime_typ stands for either a date number or a time number. Each
DTI_datetime_typ will also have a DTI_dtkind_typ associated with it,
and the DTI_datetime_typ should be replaced with an ASE_date_typ or
ASE_time_typ (see AS_externals.h) depending on whether the
DTI_dtkind_typ is DTI_DATEONLY or DTI_DATETIME. Note that a ASE_date_typ
is the number of days since a certain date, and ASE_time_typ is the number
of seconds since a midnight on a certain date. Both ASE_date_typ and
ASE_time_typ are GMT times. */

typedef long DTI_datetime_typ;

/* Type which indicates whether a DTI_datetime_typ is a date or a time */

typedef char DTI_dtkind_typ;
#define DTI_DATE          1 /* date (day granularity) */
#define DTI_DATEONLY     DTI_DATE /* date (day granularity) */
#define DTI_TIME          2 /* date and time (second granularity) */
#define DTI_DATETIME     DTI_TIME /* date and time (second granularity) */
#define DTI_TIMEONLY     3 /* time only (time mask only)
This should only be used in calls to
DTI_mask_value() to obtain a time-only
mask. The calling application must
then combine this time-only mask with
the date mask to obtain a complete
date/time mask */

/* Type which is used for a string date or time format. To get the maximum
length of a date or time string, call "DTI_mask_strlen(...)", and this will
return the maximum length of a date or time string which conforms to a
given mask. */

typedef char*          DTI_dtstring_ptr_typ;

/* The mask value indicates the format of a date or date/time string.
The mask consists of components, and a separator character. For
example, in "yy/mm/dd", the separator character is a "/", and the
components are "yy" for a two digit year number, "mm" for a two digit
month number, and "dd" for a two digit day number.

The separator can be just about any non-alphanumeric character. The
list of valid separators follows:
```

+++ DTI - Date Time Interface +++

,./?:;\'`"(<>[]{-_+=|*%^\$#@!

The original components can be any of the following:

dd - day of month (01..31)
ddd - three digit day of year (1..365)
mm - month (01..12)
mon - abbreviated month name
month - unabbreviated month name
yy - two digit year
yyyy - four digit year
w - day of week (0..6, Unix: Sunday=0 NT: Monday=0)
day - day name (abbreviated)
dayname - day name (unabbreviated)
hh - hour of day (01..12 if am not specified, o/w 00..23)
tt - minute of hour (00..59)
ss - second of minute (00..59)
am - "am" or "pm"

The above list was modified in 3.5.0 to include the following masks:

d day (no leading zero for single digit)
m month (no leading zero for single digit)
t minute (no leading zero for single digit)
s second (no leading zero for single digit)
h hour 12hr format (no leading zero for single digit)
H hour 24hr format (no leading zero for single digit)
HH hour 24hr format

If lower case 'h' or 'hh' is used, 24 hr format is assumed when the am/pm mask is absent. 'H' or 'HH' overrides the am/pm mask (forces 24 hour time). Note that the am/pm mask for FileNET is 'am' but for NT it is 'tt'. Examples:

hh:tt:ss am -> 01:02:03 PM
hh:tt:ss -> 13:02:03
HH:tt:ss am -> 13:02:03 PM
HH:tt:ss -> 13:02:03

The following mask conversions occur on NT.

These masks can be entered in the Regional Settings panel. They are converted to FileNET standard masks for use in DTI. When displayed through a GUI, they are converted back to NT style for consistency with Regional Settings.

The conversions are performed in NLT.c nlt_NT2FN_mask().

+++ DTI - Date Time Interface +++

Left column: the mask as input in Regional Settings.
the mask as displayed in Image Services UI windows.

Right column: the converted mask used in DTI.

dddd	<->	dayname	day name (full)
M	<->	m	month (no leading zero)
MM	<->	mm	month
MMM	<->	mon	month name (abbreviated)
MMMM	<->	month	month name (full)
m	<->	t	minutes (no leading zero)
mm	<->	tt	minutes
tt	<->	am	am/pm

The following conversions are not reversible because they are not unique.

Left column: the mask as input in Regional Settings.

Middle column: the converted mask used in DTI.

Right column: the mask as displayed in Image Services UI windows.

y	->	YY	->	YY	2 digit year
YYY	->	YYYY	->	YYYY	4 digit year
t	->	am	->	tt	am/pm

Note: The 'ddd' mask is used on NT as an abbreviated day name but on Image Services it is used as the Julian Day (day of year 1-366). It is not converted to its Image Services equivalent 'day'.

The following entry points perform FileNET <=> native conversions:

```
DTI_native2FN_mask(native_p, FNmask_p)
DTM_native2FN_mask(native_p, FNmask_p)
DTI_FN2native_mask(FNmask_p, native_p)
DTM_FN2native_mask(FNmask_p, native_p)
```

When doing a date/time number to/from string conversion, components may be specified in any order and may be separated by a separator character, a blank, or nothing at all (i.e. yymmddhhttss).

*/

```
typedef char* DTI_mask_ptr_typ;
```

```
/* Misc defines */
```


+++ DTI - Date Time Interface +++

```
#define DTI_NODATE          -2000000000L    /* Undefined date */

/* Date/time structure */

typedef struct DTI_tm_struct {
    int    tm_sec;      /* 0..59 */
    int    tm_min;      /* 0..59 */
    int    tm_hour;     /* 0..23 */
    int    tm_mday;     /* 1..31 */
    int    tm_mon;      /* 0..11 */
    int    tm_year;     /* 1900 and something */
    int    tm_wday;     /* 0..6, Sunday = 0 */
    int    tm_yday;     /* 0..365 */
    int    tm_isdst;    /* non-zero if is daylight savings time */
} DTI_tm_struct_typ;

#define DTI_MASKSIZE      256 /* maximum mask size (also NLT.c NLT_MASKSIZE) */

#if defined(NOT_USED)
/*
** Convert Unix date mask to NT date mask.
** Replace all 'm' with 'M' ('m' on NT is minutes)
** Must use "old" (before RCI 1768) date masks.
*/

#define DTI_Unix2NT( date_mask ) \
{ \
    char    *    mask_p; \
    mask_p = date_mask; \
    while (*mask_p) { \
        if (*mask_p == 'm' && *(mask_p+1) != 'o' && *(mask_p+1) != 'e') \
            *mask_p = 'M'; \
        mask_p++; \
    } \
} \

/*
** Convert NT date mask to Unix date mask.
** Replace all 'M' with 'm' ('mm' is minutes on NT and months on Unix)
** Must use "old" (before RCI 1768) date masks.
*/

#define DTI_NT2Unix( date_mask ) \
{ \
    char    *    mask_p; \
    mask_p = date_mask; \
    while (*mask_p) { \

```

+++ DTI - Date Time Interface +++

```
    if (*mask_p == 'M') \  
        *mask_p = 'm'; \  
    mask_p++; \  
} \  
#endif /* NOT_USED */
```

```
#define err_DTI                err_DTM  
  
#define DTI_InvalidMask                err_encode( err_DTI, 0, 3 )  
/*$M Invalid date/time mask */  
#define DTI_TooManyMaskItems            err_encode( err_DTI, 0, 4 )  
/*$M Too many items used in the date/time mask. Limit is 24. */  
#define DTI_NoCurrentTime                err_encode( err_DTI, 0, 6 )  
/*$M cannot read the current time */  
#define DTI_SetTimeFailed                err_encode( err_DTI, 0, 9 )  
/*$M failed to set the system time */  
#define DTI_InvalidInput                err_encode( err_DTI, 0, 13 )  
/*$M Invalid date/time input */  
#define DTI_BadMaskMonth                 err_encode( err_DTI, 0, 14 )  
/*$M could not find the month specified by the mask */  
#define DTI_BadMaskDay                   err_encode( err_DTI, 0, 16 )  
/*$M could not find the abbreviated day of week specified by the mask */  
#define DTI_BadMaskHour                   err_encode( err_DTI, 0, 17 )  
/*$M Invalid hour specified for 12 hour mask (am/pm) */  
#define DTI_InconsistantSep              err_encode( err_DTI, 0, 18 )  
/*$M inconsistency between separators in the input and the mask */  
#define DTI_NoNullTerm                   err_encode( err_DTI, 0, 19 )  
/*$M date string does not end with a NULL as specified by the mask */  
#define DTI_NumericLength                 err_encode( err_DTI, 0, 21 )  
/*$M length of numeric data too long */  
#define DTI_MissingNumeric                err_encode( err_DTI, 0, 22 )  
/*$M no numeric data found although specified by the mask */  
#define DTI_AlphaLength                   err_encode( err_DTI, 0, 23 )  
/*$M length of alphabetic data too great */  
#define DTI_MissingAlpha                  err_encode( err_DTI, 0, 24 )  
/*$M no alphabetic data found although specified by the mask */  
#define DTI_SecondRange                   err_encode( err_DTI, 0, 28 )  
/*$M second specified out of range -- valid range (0-59) */  
#define DTI_MinuteRange                   err_encode( err_DTI, 0, 29 )  
/*$M minute specified out of range -- valid range (0-59) */  
#define DTI_HourRange                     err_encode( err_DTI, 0, 30 )  
/*$M hour specified out of range -- valid range (0-23) */  
#define DTI_DayOfWeekRange                err_encode( err_DTI, 0, 31 )  
/*$M day of week specified out of range -- valid range (0-6) */  
#define DTI_DayOfYearRange                err_encode( err_DTI, 0, 32 )  
/*$M day of year specified out of range -- valid range (0-365) */  
#define DTI_28DayRange                    err_encode( err_DTI, 0, 34 )
```

+++ DTI - Date Time Interface +++

```
/*$M day of month specified > 28 in February in a normal year */
#define DTI_29DayRange err_encode( err_DTI, 0, 35 )
/*$M day of month specified > 29 in February in a leap year */
#define DTI_DayOfMonthRange err_encode( err_DTI, 0, 36 )
/*$M day of month specified out of range */
#define DTI_MonthRange err_encode( err_DTI, 0, 37 )
/*$M month out of range */
#define DTI_YearRange err_encode( err_DTI, 0, 38 )
/*$M year out of range */
#define DTI_InvalidType err_encode( err_DTI, 0, 39 )
/*$M invalid date/time type */
#define DTI_GeneralRange err_encode( err_DTI, 0, 40 )
/*$M date/time out of range */
#define DTI_InvalidAmPm err_encode( err_DTI, 0, 41 )
/*$M Invalid am/pm specification */
#define DTI_InvalidTimeZone err_encode( err_DTI, 0, 42 )
/*$M Invalid time zone specification */
#define DTI_LanguageUnavail err_encode( err_DTI, 0, 43 )
/*$M Language requested not available */
#define DTI_VersionMismatch err_encode( err_DTI, 0, 90 )
/*$M DTM version mismatched */
#define DTI_SetConstFail err_encode( err_DTI, 0, 91 )
/*$M error in setting up the date/time constants */
#define DTI_UnlinkFail err_encode( err_DTI, 0, 92 )
/*$M error in unlinking the abstract */
#define DTI_ConstantFileReadFail err_encode( err_DTI, 0, 94 )
/*$M error in reading the date/time constant file */
#define DTI_NoMaskMemory err_encode( err_DTI, 0, 95 )
/*$M Invalid date/time mask specified */
#define DTI_TwoWordsReqd err_encode( err_DTI, 0, 96 )
/*$M Syntax error in month/day names on PC */
#define DTI_GetLocaleInfoFail err_encode( err_DTI, 0, 97 )
/*$M Failed to get locale info */
#define DTI_UnsupportedPosixMask err_encode( err_DTI, 0, 98 )
/*$M Unsupported mask */
#define DTI_FNMaskSizeTooSmall err_encode( err_DTI, 0, 99 )
/*$M The allocated space is too small to store the converted mask */
#endif /* DTI.defs */
```

15. DTI SUBROUTINES

15.1. DTI.UTC_to_local_struct

error_typ FN_ENTRY

+++ DTI - Date Time Interface +++

```
DTI.UTC_to_local_struct(  
    DTI_tm_struct_typ      *local_out_p,      (pg179.16)  
    DTI_tm_struct_typ      *UTC_in_p)        (pg179.16)
```

FUNCTION: DTI.UTC_to_local_struct

PURPOSE: This routine converts a UTC time stored in a DTI time structure to local time stored in a different DTI time structure. All the fields of the date-time structure will be valid, including tm_wday, and tm_yday.

PARAMETERS:

local_out_p:

The fields of local_out_p will be set to the local time converted from *UTC_in_p.

UTC_in_p:

*UTC_in_p contains the UTC time to be converted to local time.

RETURNS: Standard error tuple.

PRECONDITIONS:

local_out_p != UTC_in_p (i.e., must be two different structures).

REMARKS:

This routine was created for WorkFlo so that it could work with triggers firing after year 2038.

=====*/

15.2. DTI_addtime

error_typ FN_ENTRY

```
DTI_addtime (sess_num, struct1_p, struct2_p, result_struct_p)  
    ASE_session_number_typ sess_num;      /* IN: IMS handle */      (pg483.24)  
    DTI_tm_struct_typ*      struct1_p;    /* IN: Date/time structure */ (pg179.16)  
    DTI_tm_struct_typ*      struct2_p;    /* IN: Date/time structure */ (pg179.16)  
    DTI_tm_struct_typ*      result_struct_p; /* OUT: Date/time structure */ (pg179.16)
```

/* Computes *result_struct_p = #struct1_p + *struct2_p.

Note: DTI_addtime must be used with caution when adding one date to another. If the dates involve time calculation (rather than just year/month/day), then the resulting date is converted to the number of seconds before or after 1970. If this value exceeds a signed long word (ASE_time_typ), then new date is considered invalid (because it cannot be used in other DTI routines) and the error DTI_GeneralRange is returned. However, the calculated values in the result_struct_p will be correct.

*/

15.3. DTI_addtimeUTC

error_typ FN_ENTRY

+++ DTI - Date Time Interface +++

```
DTI_addtimeUTC(  
    DTI_tm_struct_typ      *UTC_out_p,          (pg179.16)  
    DTI_tm_struct_typ      *UTC_in_p,          (pg179.16)  
    DTI_tm_struct_typ      *increment_p)      (pg179.16)
```

FUNCTION: DTI_addtimeUTC

PURPOSE:

This routine adds an increment to a base date-time. (This routine does NOT work with dates, only with date-times.) All the date-times involved MUST be in Coordinated Universal Time. WARNING: The `tm_wday`, `tm_yday`, and `tm_isdst` fields will not be valid in the output structure.

NOTE: This routine will work for date-times beyond year 2038.

PARAMETERS:

UTC_out_p:

The fields of `UTC_out_p` will be set to the resulting date-time. If the resulting date-time is beyond year 9999, an error is returned. WARNING: The following fields will NOT be valid (they will be zero):
`tm_wday`
`tm_yday`
`isdst`

UTC_in_p:

This points to the base date-time expressed in UTC. It is OK if `UTC_out_p == UTC_in_p`, in which case the structure will be incremented in place.

increment_p:

Points to the structure that holds the increment that will be added to the base date-time. The following fields must ALL have valid values:
`tm_sec`
`tm_min`
`tm_hour`
`tm_mon`
`tm_dmon`
`tm_year`

The fields that do not contribute to the increment value must be zero. The fields that contribute to the date-time value must be POSITIVE. Any subset of the above fields may be positive. Most of the fields are not subjected to the normal range restrictions. For example, `tm_sec` may be greater than 59; `tm_min` may be greater than 59; `tm_hour` may be greater than 23; `rm_dmon` may be greater than 31.

The rest of the fields of the structure must be zero, and are otherwise ignored.

RETURNS: Standard error tuple.

PRECONDITIONS:

`UTC_in_p != increment_p & UTC_out_p != increment_p`.

REMARKS:

This routine was created for WorkFlo so that it could work with triggers

+++ DTI - Date Time Interface +++

firing after year 2038.

=====*/

15.4. DTI_dtnum_to_string

```
error_typ FN_ENTRY
DTI_dtnum_to_string (sess_num, dtnum_type, dtnum, string_zone, mask_p, str_p)
  ASE_session_number_typ  sess_num; /* IN: IMS handle */ (pg483.24)
  DTI_dtkind_typ          dtnum_type; /* IN: DTI_DATEONLY or DTI_DATETIME (pg176.20)
                               type of dtnum */
  DTI_datetime_typ        dtnum; /* IN: Date or time number */ (pg176.16)
  DTI_zone_typ            string_zone; /* IN: DTI_LOCAL or DTI_GMT, (pg176.3)
                               timezone of *str_p */
  DTI_mask_ptr_typ        mask_p; /* IN: Mask for string format, or (pg178.46)
                               NULL to indicate use system
                               default mask. */
  DTI_dtstring_ptr_typ    str_p; /* OUT: String time returned */ (pg176.38)

/* Converts a date or time number to a string. */
```

15.5. DTI_dtnum_to_string_no_pad

```
error_typ FN_ENTRY
DTI_dtnum_to_string_no_pad (
  ASE_session_number_typ  sess_num, /* IN: IMS handle */ (pg483.24)
  DTI_dtkind_typ          dtnum_type, /* IN: DTI_DATEONLY or DTI_DATETIME */ (pg176.20)
  DTI_datetime_typ        dtnum, /* IN: Date or time number */ (pg176.16)
  DTI_zone_typ            string_zone, /* IN: DTI_LOCAL or DTI_GMT */ (pg176.3)
  DTI_mask_ptr_typ        mask_p, /* IN: Mask for string format, or (pg178.46)
                               NULL to indicate use system
                               default mask. */
  DTI_dtstring_ptr_typ    str_p) /* OUT: String time returned */ (pg176.38)

/* Converts a date or time number to a string. */
```

15.6. DTI_dtnum_to_struct

```
error_typ FN_ENTRY
DTI_dtnum_to_struct (sess_num, dtnum_type, dtnum, struct_zone, struct_p)
  ASE_session_number_typ  sess_num; /* IN: IMS handle */ (pg483.24)
  DTI_dtkind_typ          dtnum_type; /* IN: DTI_DATEONLY or DTI_DATETIME, (pg176.20)
                               type of dtnum */
  DTI_datetime_typ        dtnum; /* IN: Date or time number */ (pg176.16)
  DTI_zone_typ            struct_zone; /* IN: DTI_LOCAL or DTI_GMT, (pg176.3)
                               timezone of *struct_p */
```

+++ DTI - Date Time Interface +++

DTI_tm_struct_typ* struct_p; /* OUT: Date/time structure returned */ (pg179.16)

/* Converts a date or time number to a date/time structure */

15.7. DTI_get_cur_UTC_struct

error_typ FN_ENTRY

DTI_get_cur_UTC_struct(
DTI_tm_struct_typ *UTC_p) (pg179.16)

FUNCTION: DTI_get_cur_UTC_struct

PURPOSE:

This routine returns the current time in UTC (Coordinated Universal Time) as a DTI time structure. All the fields of the date-time structure will be valid, including tm_wday, and tm_yday.

WARNING: THIS ROUTINE ONLY WORKS UP TO YEAR 2038! BEFORE THEN, IF THIS SOFTWARE IS STILL IN PRODUCTION, THIS ROUTINE WILL HAVE TO BE ENHANCED. A LOT OF OTHER STUFF IN IMAGE SERVICES WILL ALSO HAVE TO BE ENHANCED, AS WELL AS A LOT OF OTHER SOFTWARE IN THE INDUSTRY, AND A LOT OF FILES ON DISK AND OPTICAL DISK. BASICALLY, ALL SOFTWARE WILL HAVE TO BE COMPILED IN A 64-bit COMPILATION ENVIRONMENT, DATE-TIMES ON PERMANENT STORAGE WILL HAVE TO BE CONVERTED, AND THE EPOCH (NUMBER OF SECONDS SINCE January 1, 1970 UTC) WILL BE 64 BITS INSTEAD OF 32 BITS.

PARAMETERS:

struct_p:

The fields of struct_p will be set appropriately to reflect the current value of Coordinated Universal Time.

RETURNS: void

REMARKS:

This routine was created for WorkFlo so that it could work with triggers firing after year 2038.

=====*/

15.8. DTI_get_cur_datetime_num

error_typ FN_ENTRY

DTI_get_cur_datetime_num (sess_num, type, dtnum_p)
ASE_session_number_typ sess_num; /* IN: IMS handle */ (pg483.24)
DTI_dtkind_typ type; /* IN: If want a date or time */ (pg176.20)
DTI_datetime_typ* dtnum_p; /* OUT: Current date or time */ (pg176.16)

/* Returns the current local time of the local station (not the server) as either a date number or a time number. */

+++ DTI - Date Time Interface +++

15.9. DTI_get_cur_datetime_string

error_typ FN_ENTRY

DTI_get_cur_datetime_string (sess_num, mask_p, str_p)

ASE_session_number_typ sess_num; /* IN: IMS handle */ (pg483.24)

DTI_mask_ptr_typ mask_p; /* IN: Mask for string format or NULL (pg178.46)
for system date/time mask */

DTI_dtstring_ptr_typ str_p; /* OUT: String time returned */ (pg176.38)

/* Returns the current local time of the local station as a string */

15.10. DTI_get_cur_datetime_struct

error_typ FN_ENTRY

DTI_get_cur_datetime_struct (sess_num, struct_p)

ASE_session_number_typ sess_num; /* IN: IMS handle */ (pg483.24)

DTI_tm_struct_typ* struct_p; /* OUT: Date/time structure returned */ (pg179.16)

/* Returns the current local time of the local station as a structure. */

15.11. DTI_getmask

error_typ FN_ENTRY

DTI_getmask(mask, type)

char * mask; /* OUT: the system default mask */

DTI_dtkind_typ type; /* IN: DTI_DATEONLY, DTI_DATETIME, or DTI_TIMEONLY

*/(pg176.20)

/*

* NAME : DTI_getmask

*

* PURPOSE : This function will retrieve the default

* system date mask for a given date type.

* If the default system mask is invalid, a

* hard-coded mask is returned. The return

* status is always success. This routine

* should be called when a valid mask must

* be obtained, regardless of the system

* default mask. Applications which must

* deal with invalid system masks need to

* call DTI_mask_value() instead.

*

* Same functionality as DTM_getmask()

*

* PARAMETERS : NAME : mask

* I/O : output

* TYPE : char pointer

+++ DTI - Date Time Interface +++

```
*      DESC : the mask
*
*      NAME : type
*      I/O  : input
*      TYPE : DTI_dtkind_typ
*      DESC : the date, date/time, or time only type
*
* ERRORS : always returns success
*/
```

15.12. DTI_local_to.UTC_struct

```
error_typ FN_ENTRY
DTI_local_to.UTC_struct(
    DTI_tm_struct_typ    *UTC_out_p,           (pg179.16)
    DTI_tm_struct_typ    *local_in_p)        (pg179.16)
```

FUNCTION: DTI_local_to.UTC_struct

PURPOSE: This routine converts a UTC time stored in a DTI time structure to local time stored in a different DTI time structure. All the fields of the date-time structure will be valid, including tm_wday, and tm_yday.

PARAMETERS:

UTC_out_p:
The fields of UTC_out_p will be set to the local time converted from *local_in_p.

local_in_p:
*local_in_p contains the local time to be converted to UTC.

RETURNS: Standard error tuple.

PRECONDITIONS:

UTC_out_p != local_in_p (i.e., must be two different structures).

REMARKS:

This routine was created for WorkFlo so that it could work with triggers firing after year 2038.

=====*/

15.13. DTI_make4y

```
error_typ FN_ENTRY
DTI_make4y(mask)
char *mask; /* IN/OUT: pointer to date mask */

/* DTI_make4y(): */
/* Make sure there are four "y"s in date format */
/* e.g. mm/dd/yy -> mm/dd/yyyy */
/* mm/dd/yyyy -> mm/dd/yyyy */
```

+++ DTI - Date Time Interface +++

```
/*      yy/mm/dd   ->  yyyy/mm/dd           */
/*      yyyy/mm/dd ->  yyyy/mm/dd           */
/*      mm/dd/ccyy ->  mm/dd/yyyy           */
/*      ccyy/mm/dd ->  yyyy/mm/dd           */

/*
 * Note: the array containing the mask must be padded by at least two
 *       characters in order to allow the insertion of the extra 'yy'
 */
```

15.14. DTI_mask_strlen

```
error_typ FN_ENTRY
DTI_mask_strlen (sess_num, mask_p, length_p)
    ASE_session_number_typ  sess_num; /* IN:  IMS handle */           (pg483.24)
    DTI_mask_ptr_typ        mask_p;   /* IN:  Date or time mask */      (pg178.46)
    long*                   length_p; /* OUT: Max length when using mask */

/* If you use the mask in mask_p to create a date/time string, the length of
the string will be at most *length_p bytes (including the null at the
end of the string). */
```

15.15. DTI_mask_value

```
error_typ FN_ENTRY
DTI_mask_value (sess_num, mask_type, mask_p)
    ASE_session_number_typ  sess_num; /* IN:  IMS handle */           (pg483.24)
    DTI_dtkind_typ         mask_type; /* IN:  Date or time mask */      (pg176.20)
    DTI_mask_ptr_typ        mask_p;   /* OUT: Mask returned */         (pg178.46)

/* Gets the system default date or date/time mask. For example, the default
date mask could be "dd/mm/yy", and the default time mask could be
"dd/mm/yy hh:mm:ss". */
```

15.16. DTI_mask_verify

```
error_typ FN_ENTRY
DTI_mask_verify (sess_num, mask_p)
    ASE_session_number_typ  sess_num; /* IN:  IMS handle */           (pg483.24)
    DTI_mask_ptr_typ        mask_p;   /* IN:  Date or time mask */      (pg178.46)

/* Verifies the mask and returns 0 if the mask is valid, and non-zero
if not valid. */
```

+++ DTI - Date Time Interface +++

15.17. DTI_oldmask

```
error_typ FN_ENTRY
DTI_oldmask(mask_p)
DTI_mask_ptr_typ      mask_p;      /* IN:  Mask to check */      (pg178.46)
```

```
/*
DTI_oldmask()      STR 50442
```

Check the mask for backwards compatibility. Check that the mask contains ONLY the following "old" (pre-350) valid masks:

```
    day
    dayname
    dd
    ddd
    mm
    mon
    month
    w
    YY
    YYYY
```

Input: Pointer to the mask string
Return: Success if the mask only contains "old" mask types.
 DTI_InvalidMask if the mask contains anything else.

```
*/
```

15.18. DTI_string_to_dtnum

```
error_typ FN_ENTRY
DTI_string_to_dtnum (sess_num, string_zone, mask_p, str_p, dtnum_type, dtnum_p)
    ASE_session_number_typ  sess_num;      /* IN:  IMS handle */      (pg483.24)
    DTI_zone_typ            string_zone; /* IN:  DTI_LOCAL or DTI_GMT,      (pg176.3)
                                timezone of *str_p. */
    DTI_mask_ptr_typ        mask_p;      /* IN:  Mask for string format, or      (pg178.46)
                                NULL to indicate use system
                                default mask. */
    DTI_dtstring_ptr_typ    str_p;      /* IN:  String time */      (pg176.38)
    DTI_dtkind_typ          dtnum_type; /* IN:  DTI_DATEONLY or DTI_DATETIME,  (pg176.20)
                                type of *dtnum_p */
    DTI_datetime_typ*       dtnum_p;      /* OUT: Date or time returned */      (pg176.16)
```

```
/* Converts a string to a date or time number.
```

+++ DTI - Date Time Interface +++

If the mask does not contain all components of a time (year, month, day, hours, minutes, seconds), the missing elements will be filled in with the current time (not 1 or 0). If filling in a missing the day of the month results in the day of the month being invalid, the day will be decremented until it is valid.

*/

15.19. DTI_string_to_dtnum_c

error_typ FN_ENTRY

DTI_string_to_dtnum_c (sess_num, string_zone, mask_p, str_p, dtnum_type, dtnum_p, century)

ASE_session_number_typ	sess_num;	/* IN: IMS handle */	(pg483.24)
DTI_zone_typ	string_zone;	/* IN: DTI_LOCAL or DTI_GMT, timezone of *str_p. */	(pg176.3)
DTI_mask_ptr_typ	mask_p;	/* IN: Mask for string format, or NULL to indicate use system default mask. */	(pg178.46)
DTI_dtstring_ptr_typ	str_p;	/* IN: String time */	(pg176.38)
DTI_dtkind_typ	dtnum_type;	/* IN: DTI_DATEONLY or DTI_DATETIME, type of *dtnum_p */	(pg176.20)
DTI_datetime_typ*	dtnum_p;	/* OUT: Date or time returned */	(pg176.16)
long	century;	/* IN: century or threshold */	

/* Converts a string to a date or time number.

If the mask does not contain all components of a time (year, month, day, hours, minutes, seconds), the missing elements will be filled in with the current time (not 1 or 0). If filling in a missing the day of the month results in the day of the month being invalid, the day will be decremented until it is valid.

*/

15.20. DTI_string_to_struct

error_typ FN_ENTRY

DTI_string_to_struct (sess_num, mask_p, str_p, struct_p)

ASE_session_number_typ	sess_num;	/* IN: IMS handle */	(pg483.24)
DTI_mask_ptr_typ	mask_p;	/* IN: Mask for string format */	(pg178.46)
DTI_dtstring_ptr_typ	str_p;	/* IN: String time */	(pg176.38)
DTI_tm_struct_typ*	struct_p;	/* OUT: Date/time structure */	(pg179.16)

/* Converts a string to a date/time structure. Note that GMT or local time is not specified on the input or output, because the output time zone will be the same as the input time zone.

+++ DTI - Date Time Interface +++

If the mask does not contain all components of a time (year, month, day, hours, minutes, seconds), the missing elements will be filled in with the current time (not 1 or 0). If filling in a missing the day of the month results in the day of the month being invalid, the day will be decremented until it is valid.

*/

15.21. DTI_string_to_struct_c

error_typ FN_ENTRY

```
DTI_string_to_struct_c (sess_num, mask_p, str_p, struct_p, century)
  ASE_session_number_typ  sess_num;    /* IN:  IMS handle */           (pg483.24)
  DTI_mask_ptr_typ        mask_p;      /* IN:  Mask for string format */ (pg178.46)
  DTI_dtstring_ptr_typ    str_p;       /* IN:  String time */         (pg176.38)
  DTI_tm_struct_typ*     struct_p;     /* OUT: Date/time structure */  (pg179.16)
  long                    century;     /* IN:  Century or threshold */
```

/* Converts a string to a date/time structure. Note that GMT or local time is not specified on the input or output, because the output time zone will be the same as the input time zone.

If the mask does not contain all components of a time (year, month, day, hours, minutes, seconds), the missing elements will be filled in with the current time (not 1 or 0). If filling in a missing the day of the month results in the day of the month being invalid, the day will be decremented until it is valid.

*/

15.22. DTI_struct_to_dtnum

error_typ FN_ENTRY

```
DTI_struct_to_dtnum (sess_num, struct_zone, struct_p, dtnum_type, dtnum_p)
  ASE_session_number_typ  sess_num;    /* IN:  IMS handle */           (pg483.24)
  DTI_zone_typ            struct_zone; /* IN:  DTI_LOCAL or DTI_GMT,   (pg176.3)
                               timezone of *struct_p */
  DTI_tm_struct_typ*     struct_p;     /* IN:  Date/time structure */  (pg179.16)
  DTI_dtkind_typ         dtnum_type;   /* IN:  DTI_DATEONLY or DTI_DATETIME, (pg176.20)
                               type of *dtnum_p */
  DTI_datetime_typ*     dtnum_p;      /* OUT: Date or time returned */ (pg176.16)
```

/* Converts a date structure to a date or time number */

15.23. DTI_struct_to_string

error_typ FN_ENTRY

+++ DTI - Date Time Interface +++

```
DTI_struct_to_string (sess_num, struct_p, mask_p, str_p)
  ASE_session_number_typ  sess_num;    /* IN:  IMS handle */           (pg483.24)
  DTI_tm_struct_typ*      struct_p;     /* IN:  Date/time structure */  (pg179.16)
  DTI_mask_ptr_typ        mask_p;       /* IN:  Mask for string format */ (pg178.46)
  DTI_dtstring_ptr_typ    str_p;        /* OUT: String time */         (pg176.38)
```

/* Converts a date/time structure to a string. Note that GMT or local time is not specified on the input or output, because the output time zone will be the same as the input time zone. */

15.24. DTI_struct_to_string_no_pad

error_typ FN_ENTRY

```
DTI_struct_to_string_no_pad (
  ASE_session_number_typ  sess_num,    /* IN:  IMS handle */           (pg483.24)
  DTI_tm_struct_typ*      struct_p,     /* IN:  Date/time structure */  (pg179.16)
  DTI_mask_ptr_typ        mask_p,       /* IN:  Mask for string format */ (pg178.46)
  DTI_dtstring_ptr_typ    str_p)        /* OUT: String time */         (pg176.38)
```

/* Converts a date/time structure to a string. Note that GMT or local time is not specified on the input or output, because the output time zone will be the same as the input time zone. */

15.25. DTI_subtime

error_typ FN_ENTRY

```
DTI_subtime (sess_num, struct1_p, struct2_p, result_struct_p)
  ASE_session_number_typ  sess_num;    /* IN:  IMS handle */           (pg483.24)
  DTI_tm_struct_typ*      struct1_p;    /* IN:  Date/time structure */  (pg179.16)
  DTI_tm_struct_typ*      struct2_p;    /* IN:  Date/time structure */  (pg179.16)
  DTI_tm_struct_typ*      result_struct_p; /* OUT: Date/time structure */  (pg179.16)
```

/* Computes *result_struct_p = *struct1_p - *struct2_p
tm_isdst is only used in struct1_p (ignored in struct2_p).
tm_wday and tm_yday are ignored in struct1_p and struct2_p.

Note: DTI_subtime must be used with caution when subtracting one date from another. If the dates involve time calculation (rather than just year/month/day), then the resulting date is converted to the number of seconds before or after 1970. If this value exceeds a signed long word (ASE_time_typ), then new date is considered invalid (because it cannot be used in other DTI routines) and the error DTI_GeneralRange is returned. However, the calculated values in the result_struct_p will be correct.

*/

+++ DTI - Date Time Interface +++

15.26. DTI_subtimeUTC

error_typ FN_ENTRY

DTI_subtimeUTC(

DTI_tm_struct_typ	*UTC_out_p,	(pg179.16)
DTI_tm_struct_typ	*UTC_in_p,	(pg179.16)
DTI_tm_struct_typ	*decrement_p)	(pg179.16)

FUNCTION: DTI_subtimeUTC

PURPOSE:

This routine subtracts a increment from a base date-time. (This routine does NOT work with dates, only with date-times.) All the date-times involved MUST be in Coordinated Universal Time. WARNING: The tm_wday, tm_yday, and tm_isdst fields will not be valid in the output structure.

NOTE: This routine will work for date-times beyond year 2038.

PARAMETERS:

UTC_out_p:

The fields of UTC_out_p will be set to the resulting date-time. If the resulting date-time is beyond year 9999, an error is returned.

WARNING: The following fields will NOT be valid (they will be zero):

tm_wday
tm_yday
isdst

UTC_in_p:

This points to the base date-time expressed in UTC.

It is OK if UTC_out_p == UTC_in_p, in which case the structure will be incremented in place.

decrement_p:

Points to the structure that holds the decrement that will be subtracted from the base date-time. The following fields must ALL have valid values:

tm_sec
tm_min
tm_hour
tm_mon
tm_dmon
tm_year

The fields that do not contribute to the increment value must be zero.

The fields that contribute to the date-time value must be POSITIVE.

Any subset of the above fields may be positive.

Most of the fields are not subjected to the normal range restrictions.

For example, tm_sec may be greater than 59; tm_min may be greater than 59; tm_hour may be greater than 23; rm_dmon may be greater than 31.

The rest of the fields of the structure must be zero, and are otherwise ignored.

+++ DTI - Date Time Interface +++

RETURNS: Standard error tuple.

PRECONDITIONS:

UTC_in_p!=decrement_p & UTC_out_p!=decrement_p.

REMARKS:

This routine was created for WorkFlo so that it could work with triggers firing after year 2038.

=====*/

16. FFI DECLARATIONS

```

/*****

      FFI -- FileNet Formats Interface

*****/

This module handles data objects which are represented in other interfaces
as streams of unformatted bytes. This module includes routines for
encoding and decoding document pages which are of type text or image. */

#ifndef FFI_defs
#define FFI_defs

/*****Buffer structure*****/

/* If this structure is listed as an input ("IN:") on a procedure call,
   then all parameters, including the data in the buffer, are passed in
   to the procedure.

   If this structure is listed as an input/output ("IN/OUT:") of a procedure,
   then buf_size and buf_p must be set prior to the call, and the data will be
   output in the buffer, and the data_size field will be returned with a
   value as is appropriate for the routine called. */

typedef struct
{
    unsigned char*  buf_p;      /* Pointer to the buffer. */
    long            buf_size;   /* #bytes of memory allocated for buffer */
    long            data_size;  /* #bytes of data in buffer. */
} FFI_buf_typ;

/*****Image attributes*****/

/* Image formats */

typedef unsigned short      FFI_image_format_typ;
#define FFI_FMT_FILENET      0
#define FFI_FMT_CALS        1
#define FFI_FMT_TIFF        2

#define FFI_MAX_IMAGE_FORMAT  FFI_FMT_TIFF

/* Resolution in spots per inch. Supported values are 100, 200, 300, 400 */

```

+++ FFI - FileNet Formats Interface +++

```
typedef unsigned short          FFI_resolution_typ;

/* Compression algorithms used on images. Note that if the image format
   is FFI_FMT_CALS, only group 4 compression is supported, but either
   group 3 or 4 are supported for image formats of FFI_FMT_FILENET or
   FFI_FMT_TIFF. */

typedef unsigned short  FFI_image_comp_typ;
#define FFI_ICMP_G3      0 /* Group 3 compression with end of line codes.
                           (CCITT recommendation T4) */
#define FFI_ICMP_G4      1 /* Group 4 compression
                           (CCITT recommendation T6) */

#define FFI_ICMP_MAX     FFI_ICMP_G4

/* Image description structure */

typedef struct
{
    FFI_image_format_typ  image_format; /* type of image */           (pg195.40)
    bool                  tiled;        /* if image is tiled */     (pg491.22)
    FFI_resolution_typ    vres;         /* vertical resolution,
                                         in spots per inch */      (pg196.1)
    FFI_resolution_typ    hres;         /* horizontal resolution,
                                         in spots per inch */      (pg196.1)
    long                  height;       /* height in pixels of image */
    long                  width;        /* width in pixels of image */
} FFI_image_desc_typ;

#define FFI_IMAGE_DESC_VERSION 1

/*****Text attributes*****/

typedef unsigned short          FFI_text_comp_typ;
#define FFI_TCMP_NONE           0 /* not compressed */
#define FFI_TCMP_FILENET       1 /* FileNet compression algorithm */

#define FFI_TCMP_MAX           FFI_TCMP_FILENET

typedef unsigned short          FFI_char_set_typ;
#define FFI_CHAR_SET_FN_INTERNAT 2 /* FileNet international */
#define FFI_CHAR_SET_ARABIC      3 /* FileNet Arabic */
#define FFI_CHAR_SET_8859_1     11 /* ISO standard 8859-1 */
#define FFI_CHAR_SET_8859_2     12 /* ISO standard 8859-2 */
#define FFI_CHAR_SET_8859_3     13 /* ISO standard 8859-3 */
#define FFI_CHAR_SET_8859_4     14 /* ISO standard 8859-4 */
#define FFI_CHAR_SET_8859_5     15 /* ISO standard 8859-5 */
#define FFI_CHAR_SET_8859_6     16 /* ISO standard 8859-6 */
```

+++ FFI - FileNet Formats Interface +++

```

#define FFI_CHAR_SET_8859_7          17 /* ISO standard 8859-7 */
#define FFI_CHAR_SET_8859_8          18 /* ISO standard 8859-8 */
#define FFI_CHAR_SET_8859_9          19 /* ISO standard 8859-9 */
#define FFI_CHAR_SET_SHIFT_JIS       30 /* Japanese */
#define FFI_CHAR_SET_EUC_J           31 /* Japanese */
#define FFI_CHAR_SET_KSC_5601        40 /* Korean */

typedef struct
{
    FFI_text_comp_typ      compression; /* if text is compressed */           (pg196.34)
    FFI_char_set_typ       char_set;    /* character set of text */           (pg196.40)
    long                   hdr_len;     /* length of header in bytes. An
                                     uncompressed (FFI_TCMP_NONE),
                                     FileNet page has the first byte
                                     of text immediately following the
                                     the header (at byte offset
                                     "hdr_len" in the page). */
    long                   text_len;    /* length of text data. text_len +
                                     hdr_len = total size of page. */
} FFI_text_desc_typ;

#define FFI_TEXT_DESC_VERSION        1

/*****Page attributes (image or text)*****/

typedef unsigned short              FFI_page_type_typ;
#define FFI_IMAGE_PAGE_TYPE          1 /* Compressed image */
#define FFI_TEXT_PAGE_TYPE           2 /* Text data */

#define FFI_MIN_PAGE_BYTES            1024

typedef struct
{
    FFI_page_type_typ              page_type;           (pg197.26)
    union
    {
        FFI_text_desc_typ          text;               (pg197.20)
        FFI_image_desc_typ          image;              (pg196.28)
    } u;
} FFI_page_desc_typ;

#define FFI_PAGE_DESC_VERSION        1

/*****Bitmap attributes*****/

/* Fill order. Whether bit map is built with most significant bit as the
first bit of each byte, or the least significant bit as first bit
of each byte. */

```

+++ FFI - FileNet Formats Interface +++

```
typedef unsigned short          FFI_fillorder_typ;
#define FFI_FILLORDER_MSB_FIRST 0 /* default */
#define FFI_FILLORDER_LSB_FIRST 1

/* The byte multiple to pad a scan line to.  For example, use 1 for pad to
   a multiple of 8 pixels, 2 for 16, 3 for 24, etc.  A pad value of 4
   bytes gives highest performance.  */

typedef unsigned short          FFI_pad_typ;

/* ??? Add FFI_pad type to image encode options??? */

/* Indicator of whether a 1 in a bit means a black or white dot.  */

typedef unsigned short          FFI_black_color_typ;
#define FFI_BLACK_COLOR_IS_0    0 /* black pixel is a bit value = 0 */
#define FFI_BLACK_COLOR_IS_1    1 /* black pixel is a bit value = 1 */

/*****Rectangle definitions*****/

/* Upper left hand corner of rectangle (top, left) = (0, 0) */

typedef struct
{
    unsigned long top;          /* pixel # of top of rectangle */
    unsigned long left;        /* pixel # of left hand margin of rectangle */
} FFI_rect_loc_typ;

typedef struct
{
    unsigned long height;      /* height of rectangle in pixels */
    unsigned long width;       /* width of rectangle in pixels */
} FFI_rect_size_typ;

typedef struct
{
    FFI_rect_loc_typ loc;      (pg198.28)
    FFI_rect_size_typ sz;      (pg198.34)
} FFI_rectangle_typ;

/*****Encode image options*****/

typedef unsigned short          FFI_iencode_attr_typ;
#define FFI_IENCODE_IMAGE_FORMAT 0 /* Default = FFI_FMT_TIFF */
#define FFI_IENCODE_IMAGE_COMPRESSION 1 /* Default = FFI_ICMP_G4 */
#define FFI_IENCODE_TILED 2 /* Default = value which gives
                               best tradeoff (size dependent) */
```

+++ FFI - FileNet Formats Interface +++

```
#define FFI_IENCODE_VRES          3 /* Default = 200 spi */
#define FFI_IENCODE_HRES          4 /* Default = 200 spi */
#define FFI_IENCODE_BITMAP_FILLORDER 5 /* Default = MSB_FIRST */
```

```
/* Note: vertical resolution must be the same as horizontal resolution for
image formats of FFI_FMT_FILENET and FFI_FMT_CALS, and may only differ if
the image format is FFI_FMT_TIFF. */
```

```
typedef struct
{
    FFI_iencode_attr_typ      attr_type;                (pg198.44)
    union
    {
        FFI_image_format_typ  image_format; /* image format */      (pg195.40)
        FFI_image_comp_typ    compression; /* compression type */  (pg196.8)
        bool                  tiled; /* if tiled */                (pg491.22)
        FFI_resolution_typ    vres; /* vertical resolution (spi) */ (pg196.1)
        FFI_resolution_typ    hres; /* horizontal resolution (spi) */ (pg196.1)
        FFI_fillorder_typ     fillorder;                (pg198.2)
        long                  unused; /* make union 4 bytes */
    } opt;
} FFI_iencode_opt_typ;
```

```
/******Decode image options******/
```

```
/* The quick reduction option does simple scaling. The values allowed for
this option are as follows:
```

- 1 - No size reduction
- 2 - Reduces height and width to 1/2 the original size
- 4 - Reduces height and width to 1/4 the original size
- 8 - Reduces height and width to 1/8 the original size

```
When using quick reduction, the size of the image width or height
can be calculated by dividing the original size by the FFI_quick_reduce_typ
value. */
```

```
typedef unsigned short          FFI_quick_reduce_typ;
```

```
/* Decode options */
```

```
typedef unsigned short          FFI_idecode_attr_typ;
#define FFI_IDECODE_BITMAP_FILLORDER 0 /* default = FFI_FILLORDER_MSB_FIRST */
#define FFI_IDECODE_BLACK_COLOR      1 /* default = FFI_BLACK_COLOR_IS_1 */
#define FFI_IDECODE_REDUCE            2 /* default = no reduction */
```

```
typedef struct
{
```

+++ FFI - FileNet Formats Interface +++

```

FFI_idecode_attr_typ          attr_type;          (pg199.42)
union
{
    FFI_fillorder_typ        fillorder;          (pg198.2)
    FFI_black_color_typ      black;              (pg198.16)
    FFI_quick_reduce_typ     quick_reduce;       (pg199.38)
    long                     unused;             /* make union 4 bytes */
} opt;
} FFI_idecode_opt_typ;

```

/******Encode text options******/

```

typedef unsigned short          FFI_tencode_attr_typ;
#define FFI_TENCODE_CHAR_SET    0 /* Default = 8859-1 */
#define FFI_TENCODE_TEXT_COMPRESSION 1 /* Default = TCMP_FILENET */

```

```

typedef struct
{
    FFI_tencode_attr_typ        attr_type;          (pg200.13)
    union
    {
        FFI_char_set_typ        char_set; /* character set of text */ (pg196.40)
        FFI_text_comp_typ       compression; /* compression type */ (pg196.34)
        long                    unused; /* unused--make union 4 bytes */
    } opt;
} FFI_tencode_opt_typ;

```

/******Decode text options******/

/* Note that there are no decode text options defined at the present time. */

```

typedef unsigned short          FFI_tdecode_attr_typ;

typedef struct
{
    FFI_tencode_attr_typ        attr_type;          (pg200.13)
    union
    {
        long                    unused; /* unused--make union 4 bytes */
    } opt;
} FFI_tdecode_opt_typ;

```

/******Uncompressed text header options******/

```

typedef unsigned short          FFI_hdr_attr_typ;
#define FFI_HDR_CHAR_SET        0 /* Default = 8859-1 */

```

```

typedef struct

```

+++ FFI - FileNet Formats Interface +++

```
{
  FFI_hdr_attr_typ      attr_type;                (pg200.45)
  union
  {
    FFI_char_set_typ    char_set; /* character set of text */ (pg196.40)
    long                unused; /* unused--make union 4 bytes */
  } opt;
} FFI_hdr_opt_typ;
```

/*****Annotation formats*****/

/* FFI_annot_loc_typ indicates the (x,y) position of the annotation.
If not present, the default is (0,0). "x" is the horizontal distance
from the left margin of the page to the left margin of the annotation.
"y" is the vertical distance from the top of the page to the
top of the annotation. Both parameters are in 1/100ths of an inch. */

```
typedef struct
{
  unsigned short      x;
  unsigned short      y;
} FFI_annot_loc_typ;
```

/* FFI_annot_size_typ describes the size of various aspects of the
annotation. */

```
typedef struct
{
  unsigned short      height;
  unsigned short      width;
} FFI_annot_size_typ;
```

/* FFI_annot_text_typ contains the text of an annotation. Note that
annotations which are highlights only do not contain any text. */

```
typedef struct
{
  long                len; /* Number of bytes in text_p, including null
                           terminators. */
  char*               text_p; /* One or more null terminated lines of text.
                              A blank line is denoted by its own null
                              terminator (resulting in two nulls in a
                              row). */
} FFI_annot_text_typ;
```

/* FFI_annot_color_typ describes the color of various aspects of the
annotation. This three byte value represents the intensities of
the colors red, green, blue. White is full intensity on all colors,

+++ FFI - FileNet Formats Interface +++

which is (255,255,255). */

```
typedef struct
{
    unsigned char    red;
    unsigned char    green;
    unsigned char    blue;
} FFI_annot_color_typ;
```

```
typedef unsigned char    FFI_annot_brush_typ;
```

/* FFI_var_data_typ is used by VARs to implement their own private data types. These data types will be ignored by the FileNet software on display, but will be preserved on modify. */

```
typedef struct
{
    long            len;        /* length of data, in bytes */
    unsigned char*  data_p;    /* pointer to data bytes */
} FFI_var_data_typ;
```

/* Annotation attribute types */

```
typedef unsigned short    FFI_annot_attr_typ;
#define FFI_ANNOT_LOC      1  /* Default (x,y) = (0,0) */
#define FFI_ANNOT_CHAR_SIZE 2  /* Default height=1, width=10 */
#define FFI_ANNOT_TEXT    3
#define FFI_ANNOT_TEXT_BCOLOR 4  /* Default = white */
#define FFI_ANNOT_TEXT_FCOLOR 5  /* Default = black */
#define FFI_ANNOT_CHAR_SET 6  /* Default = FFI_CHAR_SET_8859_1 */
#define FFI_ANNOT_VAR      7
#define FFI_ANNOT_ABS      8
#define FFI_ANNOT_BRUSH_STYLE 9
#define FFI_ANNOT_BRUSH_COLOR 10
```

/* An annotation record consists of an array of the following types. Each type can only occur once in each annotation. */

```
typedef struct
{
    FFI_annot_attr_typ    attr_type;                (pg202.25)
    union
    {
        FFI_annot_loc_typ    loc;                /* Indicates the location of the upper (pg201.22)
                                                    left hand corner of the annotation.
                                                    Upper left hand corner of page is
                                                    (x,y) = (0,0). */
    }
}
```


+++ FFI - FileNet Formats Interface +++

```
FFI_annot_size_typ  char_size; /* Size of the box the annotation is      (pg201.31)
                               displayed in.  If the box size is
                               smaller than the annotation text,
                               then only what fits in the box will
                               be initially displayed, and the
                               user can optionally redisplay the
                               box larger to see all of the text.*/

FFI_annot_size_typ  abs_size;      (pg201.31)
FFI_annot_text_typ  text;          /* Text of the annotation */      (pg201.44)
FFI_annot_color_typ text_fcolor; /* Foreground text color */      (pg202.8)
FFI_annot_color_typ text_bcolor; /* Background text color */      (pg202.8)
FFI_annot_brush_typ brush_style; /* Brush style */                  (pg202.10)
FFI_annot_color_typ brush_color; /* Brush color text color */      (pg202.8)
FFI_char_set_typ    char_set;      /* Character set of text data */  (pg196.40)
FFI_var_data_typ    var;           /* Private data type used by VARs */ (pg202.20)
} opt;
} FFI_annot_opt_typ;

/*****Errors*****/

#define ffi_err(x)      err_encode (err_FFI, 0, x)

#define FFI_err_input_underflow      ffi_err(1)
/*$M Input buffer too small.
While trying to decode data, ran off the end of the input buffer. */

#define FFI_err_output_overflow      ffi_err(2)
/*$M Output buffer too small.
More data was requested than space exists in the output buffer. */

#define FFI_err_unknown_page_type    ffi_err(3)
/*$M Unrecognized page type.
The page type is not recognized or supported by this software. */

#define FFI_err_wrong_page_type      ffi_err(4)
/*$M Bad page type for operation requested.
For example, you cannot call image_decode with a text page. */

#define FFI_err_invalid_reduce_option ffi_err(5)
/*$M Invalid reduce option type */

#define FFI_err_invalid_tcmp_option  ffi_err(6)
/*$M Invalid text compression option */

#define FFI_err_invalid_option      ffi_err(7)
/*$M Invalid attr_type field in option list */

#define FFI_err_invalid_image_format ffi_err(8)
```

+++ FFI - FileNet Formats Interface +++

```
/*$M Invalid image format paramter */

#define FFI_err_invalid_fillorder      ffi_err(9)
/*$M Invalid fillorder parameter */

#define FFI_err_invalid_black          ffi_err(10)
/*$M Invalid black color parameter */

#define FFI_err_invalid_text_comp      ffi_err(11)
/*$M Invalid text compression parameter */

#define FFI_err_invalid_char_set       ffi_err(12)
/*$M Invalid character set parameter */

#define FFI_err_invalid_image_comp     ffi_err(13)
/*$M Invalid image compression parameter */

#define FFI_err_invalid_annot_len      ffi_err(14)
/*$M Invalid annotation length
The annotation includes part of but not all of a tagged attribute structure. */

#define FFI_err_annot_too_long         ffi_err(15)
/*$M Annotation too long */

#define FFI_err_invalid_annot_option   ffi_err(16)
/*$M Undefined annotation attribute */

#define FFI_err_badifacelevel          ffi_err(17)
/*$M bad interface level linking to FFI */

#define FFI_err_ano_dec                 ffi_err(18)
/*$M logic error in FFI_annot_decode */

#define FFI_err_ano_free                ffi_err(19)
/*$M bad pointer passed to FFI_annot_free */

#endif
```

17. FFI SUBROUTINES

17.1. FFI_annot_decode

error_typ

FFI_annot_decode (data_p, len, num_opts_p, opts_p p)

unsigned char* data_p; /* IN: Pointer to the "annot" field of the

+++ FFI - FileNet Formats Interface +++

```
unsigned short      len;          /* IN:  "annot_len" field from the
                                   DOC_annotation_desc_typ record,
                                   which is the number of bytes in
                                   "data_p". */
unsigned short*     num_opts_p; /* OUT: number of elements in the
                                   opts_p_p array */
FFI_annot_opt_typ** opts_p_p;   /* OUT: array of options for annotation.      (pg203.17)
                                   Memory for this array is allocated
                                   by this routine, and must be
                                   freed when not needed by calling
                                   FFI_annot_free. */
/* This entry point is used to decode FileNet format annotations into
   a usable C structure.
```

This entry point skips over items in an annotation which are an obsolete format, or are a format invented after this version of FFI was produced. Therefore you may find that certain annotations contain no significant data when decoded.

ERRORS:

```
    FFI_err_invalid_annot_len;
*/
```

17.2. FFI_annot_encode

error_typ

```
FFI_annot_encode (num_opts, opts_p, data_p, len_p)
```

```
    unsigned short      num_opts; /* IN:  number of elements in the
                                   opts_p_p array */
    FFI_annot_opt_typ*  opts_p;   /* IN:  array of options for annotation. */ (pg203.17)
    unsigned char*      data_p;   /* OUT: Pointer to the "annot" field of the
                                   DOC_annotation_desc_typ record. */
    unsigned short*     len_p;    /* OUT: Length of annotation data in
                                   data_p returned here. */
```

```
/* This entry point is used to build FileNet format annotations.
   Takes an annotation option array as input, and packs it into the format
   used by the "annot" field of the DOC_annotation_desc_typ record.
```

ERRORS:

```
*/
```

17.3. FFI_annot_free

```
void
```

+++ FFI - FileNet Formats Interface +++

```
FFI_annot_free (opts_p_p)
  FFI_annot_opt_typ**  opts_p_p; /* IN: Pointer to pointer to annotation
                                option array. *opts_p_p is
                                zeroed upon return. */ (pg203.17)

/* Returns the memory allocated by FFI_annot_decode. */
```

17.4. FFI_build_uncomp_text_header

```
error_typ
FFI_build_uncomp_text_header (src_length, num_options, options_p, dest_p)
  long          src_length; /* IN: Number of bytes of text */
  unsigned short num_options; /* IN: Number of options in options_p */
  FFI_hdr_opt_typ *options_p; /* IN: Array of option variables */ (pg201.8)
  FFI_buf_typ    *dest_p; /* IN/OUT: dest_p->buf_siz should be input
                            as FFI_MIN_PAGE_BYTES or greater to
                            accomodate current and future expected
                            header sizes. dest_buf_p->data_size will
                            be returned as the size of the actual
                            header created. */ (pg195.34)
```

/* This routine is used to build a FileNet format page which contains uncompressed text data. The header created by this routine is put at the beginning of the cache object, file, etc., followed immediately by the uncompressed text data.

This routine is used if you wish write an uncompressed text page, but are not able to put all the text data in memory prior to writing the page. Note that the FFI_text_encode requires all text data in memory prior to the call, whereas this routine only requires the length of the data, but not the data itself.

To read uncompressed text pages, call FFI_get_text_attr on the first FFI_MIN_PAGE_BYTES of the page to get the size of the header, and then the bytes immediately following the header will be the text data.

ERRORS:

```
  FFI_err_input_underflow
  FFI_err_output_overflow
*/
```

17.5. FFI_get_page_attr

```
error_typ
FFI_get_page_attr (desc_version, buf_p, desc_p)
  unsigned short desc_version; /* IN: desc_p version, always pass in
```

+++ FFI - FileNet Formats Interface +++

```

                                                    FFI_PAGE_DESC_VERSION */
FFI_buf_typ*      buf_p;          /* IN:  contains start of page */      (pg195.34)
FFI_page_desc_typ* desc_p;       /* OUT: Attributes of page */      (pg197.40)
```

/* Gets the attributes of a text or image page. Only the first FFI_MIN_PAGE_BYTES (or the entire page if it's smaller than FFI_MIN_PAGE_BYTES) need to be in the buffer for this routine to be called with a text page, but the entire page must be in the buffer for this routine to be called with an image.

ERRORS:

```

    FFI_err_unknown_page_type
*/
```

17.6. FFI_get_page_type

error_typ

```

FFI_get_page_type (buf_p, page_type_p)
    FFI_buf_typ*      buf_p;          /* IN:  buffer containing start of page */ (pg195.34)
    FFI_page_type_typ* page_type_p; /* OUT: type of page in buffer. */      (pg197.26)
```

/* Determines if the page is image or text data. Only the first FFI_MIN_PAGE_BYTES (or the entire page if it's smaller than FFI_MIN_PAGE_BYTES) need to be in the buffer for this routine to be called.

It's expected that this entry point will be called first to determine if the page is text or image data, and then the appropriate routine (either FFI_image_decode or FFI_text_decode) will be called to get the data into a displayable format.

ERRORS:

```

    FFI_err_unknown_page_type
*/
```

17.7. FFI_image_decode

error_typ

```

FFI_image_decode (image_p, num_options, options_p, bitmap_p)
    FFI_buf_typ      *image_p;      /* IN:  image size and image data      (pg195.34)
                                     image_p->buf_p points to the
                                     beginning of a FileNet page. */
    unsigned short   num_options; /* IN:  number of decode options */
    FFI_idcode_opt_typ *options_p; /* IN:  array of decode options */      (pg200.9)
    FFI_buf_typ      *bitmap_p;     /* IN/OUT: size and pointer to bitmap.
                                     bitmap_p->buf_size is input      (pg195.34)
```

+++ FFI - FileNet Formats Interface +++

as the buffer size,
bitmap_p->buf_p is input as
a pointer to the bitmap data,
bitmap_p->data_size is output
as the number of bytes used in the
bitmap, and the bitmap data is
output in bitmap_p->buf_p */

/* Decompresses a page of a document which is of type FFI_IMAGE_PAGE_TYPE.
The resultant bitmap is padded to 16 pixel boundaries (FFI_pad_typ == 2).

ERRORS:

FFI_err_input_underflow
FFI_err_output_overflow
FFI_err_wrong_page_type
FFI_err_unknown_page_type

*/

17.8. FFI_image_encode

error_typ

FFI_image_encode (bitmap_p, bitmap_size_p,

num_options, options_p, image_buf_p)

FFI_buf_typ	*bitmap_p;	/* IN: bitmap to compress */	(pg195.34)
FFI_rect_size_typ	*bitmap_size_p;	/* IN: size of bitmap, in pixels */	(pg198.34)
unsigned short	num_options;	/* IN: #encode options */	
FFI_iencode_opt_typ	*options_p;	/* IN: array of image options */	(pg199.22)
FFI_buf_typ	*image_buf_p;	/* IN/OUT: size and pointer to compressed image.	(pg195.34)

image_p->buf_size is input
as the buffer size,
image_p->buf_p is input as
a pointer to the image buffer,
image_p->data_size is output
as the number of bytes in the
image, and the image data is
output in image_p->buf_p */

/* Compresses a bitmap and generates a FileNet page of type image.

The bitmap width (bitmap_size_p->width) must be a multiple of
16 pixels.

ERRORS:

FFI_err_input_underflow
FFI_err_output_overflow

*/

17.9. FFI_text_decode

error_typ

FFI_text_decode (src_p, num_options, options_p, dest_p)

```
FFI_buf_typ      *src_p;      /* IN:  Data to be decoded.                (pg195.34)
                                src_p->buf_p points to the
                                beginning of a FileNet page. */
unsigned short   num_options; /* IN:  number of options in array */
FFI_tdecode_opt_typ *options_p; /* IN:  array of decode options */ (pg200.41)
FFI_buf_typ      *dest_p;     /* IN/OUT: The decoded text.          (pg195.34)
                                dest_p->data_size is returned as
                                the length of the decoded data. */
```

/* Decompresses a page of a document which is of type FFI_TEXT_PAGE_TYPE.
Note that the input page will have a header at the beginning of it, but the
decompressed data has no header and is text to be displayed/printed.

The maximum number of bytes (src_p->data_size) which will be returned by
this routine is 63*1024.

ERRORS:

```
FFI_err_input_underflow
FFI_err_output_overflow
FFI_err_wrong_page_type
FFI_err_unknown_page_type
```

*/

17.10. FFI_text_encode

error_typ

FFI_text_encode (src_p, num_options, options_p, dest_p)

```
FFI_buf_typ      *src_p;      /* IN/OUT: Text data to be encoded.          (pg195.34)
                                NOTE: Contents of this buffer are
                                destroyed by this call, because
                                is used to hold an intermediate
                                form of the result. */
unsigned short   num_options; /* IN:  #options in array below */
FFI_tencode_opt_typ *options_p; /* IN:  array of text options */ (pg200.26)
FFI_buf_typ      *dest_p;     /* IN/OUT: The encoded text.          (pg195.34)
                                dest_p->data_size is returned as
                                the length of the encoded data */
```

/* Compresses text data. The input data is text with no header preceding it.
The output data is compressed text which has a header preceding it,
and is suitable for writing to the FileNet system as a complete page of a

+++ FFI - FileNet Formats Interface +++

document. Note that due to the header, the text for a single page cannot be compressed in chunks via multiple calls to this routine.

The maximum number of bytes (src_p->data_size) which can be compressed is 63*1024

NOTE: THIS ROUTINE DESTROYS THE DATA IN THE INPUT BUFFER. The compression algorithm uses multiple passes, and the input buffer (src_p->buf_p) is modified by the first pass prior to the second pass putting the compressed data into the output buffer.

ERRORS:

FFI_err_input_underflow
FFI_err_output_overflow

*/

+++ FP - Floating Point Numbers +++

```
/* Parameters describing packed decimal (bcd) numbers */
typedef struct
{
    int                format_level;
# define FP_BCD_FORMAT_LEVEL    1

    int                sign_position;
# define FP_BCD_SIGN_LEAD      1
# define FP_BCD_SIGN_TRAIL    2

    int                is_unsigned;
# define FP_BCD_UNSIGNED      1

    int                plus_value;
# define FP_BCD_PLUS_ALLBASE  12
# define FP_BCD_PLUS_SYBASE   12

    int                minus_value;
# define FP_BCD_MINUS_ALLBASE 13
# define FP_BCD_MINUS_SYBASE  13

    int                unsigned_value;
# define FP_BCD_UNNS_ALLBASE   15
# define FP_BCD_UNNS_SYBASE    15

    int                pad;
# define FP_BCD_PAD_LEAD      0
# define FP_BCD_PAD_TRAIL    1

    int                round;
# define FP_BCD_TRUNC         0;
# define FP_BCD_ROUND         1;
}
    FP_bcd_style_typ;

/* Error number defines (category is err_FP) */

#define FP_oflo    1
#define FP_Overflow    err_encode( err_FP, 0, FP_oflo )
/*$M Numeric range overflow */

#define FP_undefopnd    2 /* input operand has the canonical illegal val */
#define FP_Undefined    err_encode( err_FP, 0, FP_undefopnd )
/*$M An input parameter has the canonical illegal numeric value.
* Having the canonical illegal value does not mean that the value is null.
* Null is the absence of a value.
```

+++ FP - Floating Point Numbers +++

* When one or more input numbers are invalid, the output FP_number is
* set to the canonical illegal value, and an error tuple is returned.
*/

```
#define FP_ilglfmt      3 /* Illegal string format for number */
#define FP_IllegalFmt  err_encode( err_FP, 0, FP_ilglfmt )
/*$M Illegal format for input number string.
*/
```

```
#define FP_BadMask      4
#define FP_InvalidMask err_encode( err_FP, 0, FP_BadMask )
/*$M Invalid numeric format mask */
```

```
#define FP_BadStyle     5
#define FP_InvalidStyle err_encode( err_FP, 0, FP_BadStyle )
/*$M Invalid packed-decimal style specification */
```

```
#define FP_BadDigits    6
#define FP_InvalidDigits err_encode( err_FP, 0, FP_BadDigits )
/*$M Packed-decimal digits specification out of range */
```

```
#define FP_BadNum       7
#define FP_IllegalNum   err_encode( err_FP, 0, FP_BadNum)
/*$M An input FP_number has an illegal value other than
* the canonical illegal value.
*/
```

```
#define FP_BadAlign     8
#define FP_IllegalAlign err_encode( err_FP, 0, FP_BadAlign )
/*$M An input or output FP_number pointer is not aligned on a longword boundary
* or is a null pointer.
* This is always a client software error.
*/
```

```
#define FP_IlglVers     1000
#define FP_badVersion   err_encode( err_FP, 0, FP_IlglVers )
/*$M Bad version number for FP abstract */
```

```
#define FP_xmain        1
```

```
#define FP_num_add      1
#define FP_num_subtract 2
#define FP_num_multiply 3
#define FP_num_divide   4
#define FP_num_neg       5
#define FP_num_abs       6
#define FP_num_trunc     7
```

+++ FP - Floating Point Numbers +++

```
#define FP_num_num2long      8
#define FP_num_long2num     9
#define FP_num_str2num     10
#define FP_num_num2str     11
#define FP_num_round       12
#define FP_num_rounddisp   13
#define FP_num_setmode     14
#define FP_num_num2ora     15
#define FP_num_ora2num     16
#define FP_num_num2sci     17
#define FP_num_num2mask    18
#define FP_num_num2dbl     19
#define FP_num_dbl2num     20
#define FP_num_num2unslong 21
#define FP_num_unslong2num 22

#endif
```

19. FP SUBROUTINES

19.1. FP_abs

```
error_typ FN_ENTRY
FP_abs(fnresult_p, fnum_p)
    FP_number    fnresult_p; /* OUT: Absolute value of fnum_p */      (pg211.20)
    FP_number    fnum_p;    /* IN:  Number to take absolute value of */ (pg211.20)

/* Sets *fnresult_p = absolute value of *fnum_p
*/
```

19.2. FP_add

```
error_typ FN_ENTRY
FP_add(fnresult_p, fnum1_p, fnum2_p)
    FP_number    fnresult_p; /* OUT: sum of fnum1_p and fnum2_p */  (pg211.20)
    FP_number    fnum1_p;    /* IN:  first number to add */        (pg211.20)
    FP_number    fnum2_p;    /* IN:  second number to add */       (pg211.20)

/* Sets *fnresult_p = *fnum1_p + *fnum2_p.
*/
```

19.3. FP_assign

+++ FP - Floating Point Numbers +++

```
void FN_ENTRY
FP_assign(fnresult_p, fnum_p)
    FP_number    fnresult_p; /* OUT: Number set to *fnum_p */      (pg211.20)
    FP_number    fnum_p;     /* IN:  Input number */              (pg211.20)

/* Sets *fnresult_p = *fnum_p
*/
```

19.4. FP_dbl2num

```
error_typ FN_ENTRY
FP_dbl2num(fnum_p, dblnum_p)
    FP_number    fnum_p; /* OUT: FileNet number */                (pg211.20)
    double      *    dblnum_p; /* IN:  C floating point number */

/* Converts a C floating point number to a FileNet number */
```

19.5. FP_divide

```
error_typ FN_ENTRY
FP_divide(fnresult_p, fnum1_p, fnum2_p)
    FP_number    fnresult_p; /* OUT: result of fnum1_p / fnum2_p */ (pg211.20)
    FP_number    fnum1_p;    /* IN:  dividend */                  (pg211.20)
    FP_number    fnum2_p;    /* IN:  divisor */                   (pg211.20)

/* Computes *fnresult_p = *fnum1_p / *fnum2_p */
```

19.6. FP_eql

```
BOOL FN_ENTRY
FP_eql(fnum1_p, fnum2_p) /* 1 -> fnum1_p == fnum2_p */
    FP_number    fnum1_p; /* IN:  FileNet number */              (pg211.20)
    FP_number    fnum2_p; /* IN:  FileNet number */              (pg211.20)

/* Returns 1 if *fnum1_p == *fnum2_p, 0 otherwise
*/
```

19.7. FP_geq

```
BOOL FN_ENTRY
FP_geq(fnum1_p, fnum2_p)
    FP_number    fnum1_p; /* IN:  FileNet number */              (pg211.20)
    FP_number    fnum2_p; /* IN:  FileNet number */              (pg211.20)
```

+++ FP - Floating Point Numbers +++

```
/* Returns 1 if *fnum1_p >= *fnum2_p, 0 otherwise
*/
```

19.8. FP_getmode

```
FP_mode_typ FN_ENTRY
FP_getmode()
```

```
/* The value of this function is FP_EUROPEAN if European format is used when
* converting FileNet numbers to/from strings, and FP_DOMESTIC for US format.
*/
```

19.9. FP_gtr

```
BOOL FN_ENTRY
FP_gtr(fnum1_p, fnum2_p)
    FP_number    fnum1_p; /* IN: FileNet number */           (pg211.20)
    FP_number    fnum2_p; /* IN: FileNet number */           (pg211.20)
```

```
/* Returns 1 if *fnum1_p > *fnum2_p, 0 otherwise
*/
```

19.10. FP_isundef

```
BOOL FN_ENTRY
FP_isundef(fnum_p)
    FP_number    fnum_p; /* IN: FileNet number */           (pg211.20)
```

```
/* Returns 1 if the number has an undefined or illegal value or is
* not aligned on a mod 4 byte boundary, 0 otherwise
*/
```

19.11. FP_leq

```
BOOL FN_ENTRY
FP_leq(fnum1_p, fnum2_p)
    FP_number    fnum1_p; /* IN: FileNet number */           (pg211.20)
    FP_number    fnum2_p; /* IN: FileNet number */           (pg211.20)
```

```
/* Returns 1 if *fnum1_p <= *fnum2_p, 0 otherwise
*/
```

+++ FP - Floating Point Numbers +++

19.12. FP_long2num

```
void FN_ENTRY
FP_long2num(fnum_p, longnum)
    FP_number      fnum_p; /* OUT: FileNet number */      (pg211.20)
    long           longnum; /* IN: long word */

/* Converts a signed longword into a FileNet number
*/
```

19.13. FP_lss

```
BOOL FN_ENTRY
FP_lss(fnum1_p, fnum2_p)
    FP_number      fnum1_p; /* IN: FileNet number */      (pg211.20)
    FP_number      fnum2_p; /* IN: FileNet number */      (pg211.20)

/* Returns 1 if *fnum1_p < *fnum2_p, 0 otherwise
*/
```

19.14. FP_multiply

```
error_typ FN_ENTRY
FP_multiply(fnresult_p, fnum1_p, fnum2_p)
    FP_number      fnresult_p; /* OUT: result of multiplication */ (pg211.20)
    FP_number      fnum1_p; /* IN: first number to multiply */ (pg211.20)
    FP_number      fnum2_p; /* IN: second number to multiply */ (pg211.20)

/* Multiplies two FileNet numbers.
   *fnresult_p = *fnum1_p times *fnum2_p */
```

19.15. FP_neg

```
error_typ FN_ENTRY
FP_neg(fnresult_p, fnum_p)
    FP_number      fnresult_p; /* OUT: Negative *fnum_p */      (pg211.20)
    FP_number      fnum_p; /* IN: FileNet number to negate */ (pg211.20)

/* Negates a FileNet number. *fnresult_p = - *fnum_p; */
```

19.16. FP_neq

```
BOOL FN_ENTRY
FP_neq(fnum1_p, fnum2_p)
```

+++ FP - Floating Point Numbers +++

```
FP_number          fnum1_p; /* IN: FileNet number */      (pg211.20)
FP_number          fnum2_p; /* IN: FileNet number */      (pg211.20)
```

```
/* Returns 1 if *fnum1_p != *fnum2_p, 0 otherwise
*/
```

19.17. FP_num2dbl

error_typ FN_ENTRY

FP_num2dbl(dblnum_p, fnum_p)

```
double            *   dblnum_p; /* OUT: C floating point number */
FP_number         fnum_p; /* IN: FileNet number */      (pg211.20)
```

```
/* Converts a FileNet number to a C floating point number */
```

19.18. FP_num2long

error_typ FN_ENTRY

FP_num2long(longnum_p, fnum_p)

```
long*            longnum_p; /* OUT: Signed long word */
FP_number        fnum_p; /* IN: FileNet number */      (pg211.20)
```

```
/* Converts a FileNet number to a signed long */
```

19.19. FP_num2mask

error_typ FN_ENTRY

FP_num2mask(str_result_p, fnum_p, mask_p)

```
char            *   str_result_p; /* OUT: String representation of number */
FP_number       fnum_p; /* IN: FileNet number to convert */
char            *   mask_p; /* IN: mask */      (pg211.20)
```

```
/* Converts a FileNet number to a string using a mask. The mask is
a character string consisting of the characters:
```

```
'<', '#', '0', '+', '-', british pound sign, '$', '.', ',', '-', ' '
```

Each character denotes the placement and type of data located
in the resultant string representation of the number.
The mask characters have the following meanings:

- '<' - forces left justification. The default is right justification.
Must be first character of the mask if used.
- '#' - a digit, with leading zeroes suppressed.
- '0' - a digit, with leading zeroes present if necessary

+++ FP - Floating Point Numbers +++

- '-' - If before or after all '#' and '0' characters, causes a minus sign to appear for a negative number, and a blank to appear for a positive number. If inbetween '#' or '0' characters, the minus sign is always put in the output string (used to put dashes in phone numbers, social security numbers, etc.).
- +' - If at the beginning or end of the mask, causes a plus/minus sign to appear to indicate postive/negative value of number. If inbetween '#' or '0' characters, the plus sign is always put in the output string.
- '\$' or british pound sign - Indicates the position of a dollar or pound sign in the resultant string.
- '.' - Indicates where to put the decimal point
- ',' - Indicates where to put a comma.
- ' ' - A space in the mask is put into the output string.

Note that the use of ',' and '.' switches in both the mask and the numeric string when the mode is changed from DOMESTIC to EUROPEAN (see FP_set_mode).

If a mask does not contain either a '+' or '-' and a negative number is input, no minus sign will appear in the output. Therefore either a '+' or '-' should always be present in a mask.

Examples:

Number	Mask	Result
+1234.56	+###,###.##	+ 1,234.56
+1234.56	+000,000.00	+001,234.56
+1234.56	-###,###.##	1,234.56
-1234.56	+###,###.##	- 1,234.56
-1234.56	-###,###.##	- 1,234.56
-1234.56	###,###.##-	1,234.56-
+1234.56	<+###,###.##	+1,234.56
+1234.56	+\$###,###.##	+\$1,234.56
+1234.56	\$+###,###.##	\$+1,234.56
+1234.56	###,###.##+\$	1,234.56+\$
+1234.56	###,###.##\$+	1,234.56\$+
+1234567	###-###	123-4567
+1234567	### #	123 4567

*/

19.20. FP_num2sci

error_typ FN_ENTRY

FP_num2sci(str_result_p, fnum_p)

char * str_result_p; /* OUT: string representation of number */

+++ FP - Floating Point Numbers +++

FP_number fnum_p; /* IN: FileNet number */ (pg211.20)

/* Converts a FileNet number to a string in scientific notation.
* Scientific notation is "+.#####E+eee",
* where "+" stands for a plus or minus sign, the "#####..." stands
* for the mantissa, and "eee" stands for the exponent.
*/

19.21. FP_num2str

error_typ FN_ENTRY
FP_num2str(result_str_p, fnum_p)
char * result_str_p; /* OUT: string representation of number */
FP_number fnum_p; /* IN: FileNet number */ (pg211.20)

/* Converts a FileNet number to a string. */

19.22. FP_num2unslong

error_typ FN_ENTRY
FP_num2unslong(unslongnum_p, fnum_p)
unsigned long* unslongnum_p; /* OUT: Unsigned long word */
FP_number fnum_p; /* IN: FileNet number */ (pg211.20)

/* Converts a FileNet number to an unsigned long word */

19.23. FP_parsenum

error_typ FN_ENTRY
FP_parsenum(fpnum_p, left_digits_p, right_digits_p)
FP_number fpnum_p; /* IN: FileNet number */ (pg211.20)
short * left_digits_p; /* OUT: number digits left of decimal */
short * right_digits_p; /* OUT: number digits right of decimal */

19.24. FP_retsign

short FN_ENTRY
FP_retsign(fnum_p)
FP_number fnum_p; /* IN: FileNet number */ (pg211.20)

/* Returns 1 if *fnum_p is positive, 0 if equal 0, and -1 if negative
*/

+++ FP - Floating Point Numbers +++

19.25. FP_round

```
error_typ FN_ENTRY
FP_round(fnresult_p, fnnum_p, n) /* fnresult_p:=(fnnum_p div 10^^n)*10^^n */
  FP_number      fnresult_p; /* OUT: FileNet number */           (pg211.20)
  FP_number      fnnum_p;    /* IN:  FileNet number */           (pg211.20)
  short          n;          /* IN:  Exponent to round off at */

/* Rounds off *fn_result_p to a power of 10.
  Computes *fn_result_p = (*fnnum_p div 10^^n) * 10^^n */
```

19.26. FP_rounddisp

```
error_typ FN_ENTRY
FP_rounddisp(fnresult_p, fnnum_p)
  FP_number      fnresult_p; /* OUT: FileNet number */           (pg211.20)
  FP_number      fnnum_p;    /* IN:  FileNet number */           (pg211.20)

/* Rounds off *fnresult_p for display purposes. For example,
  1.9999999999999999 would be rounded off to 2.0 */
```

19.27. FP_setmode

```
error_typ FN_ENTRY
FP_setmode(mode)
  FP_mode_typ    mode; /* IN:  FP_DOMESTIC or FP_EUROPEAN */       (pg211.36)

/* Sets the mode to either domestic (US) or European for FileNet
* number to/from string conversions. This setting applies only to the
* process which calls this routine.
*/
```

19.28. FP_setundef

```
void FN_ENTRY
FP_setundef(fnnum_p)
  FP_number      fnnum_p; /* OUT: Set to the canonical undefined value */ (pg211.20)

/* Sets a FileNet number to the canonical illegal value.
*/
```

19.29. FP_str2num

```
error_typ FN_ENTRY
```

+++ FP - Floating Point Numbers +++

```
FP_str2num(fnresult_p, fnstr_p)
  FP_number      fnresult_p; /* OUT: FileNet number */      (pg211.20)
  char          *      fnstr_p; /* IN: string representation of number */

/* Converts an ascii string representation of a number to a FileNet number */
```

19.30. FP_subtract

```
error_typ FN_ENTRY
FP_subtract(fnresult_p, fnum1_p, fnum2_p)
  FP_number      fnresult_p; /* OUT: FileNet number */      (pg211.20)
  FP_number      fnum1_p; /* IN: FileNet number */          (pg211.20)
  FP_number      fnum2_p; /* IN: FileNet number */          (pg211.20)

/* Computes *fnresult_p = *fnum1_p - *fnum2_p */
```

19.31. FP_trunc

```
error_typ FN_ENTRY
FP_trunc(fnresult_p, fnum_p)
  FP_number      fnresult_p; /* OUT: Truncated FileNet number */ (pg211.20)
  FP_number      fnum_p; /* IN: FileNet number */ (pg211.20)

/* Truncates a FileNet number */
```

19.32. FP_unslong2num

```
void FN_ENTRY
FP_unslong2num(fnum_p, unslongnum)
  FP_number      fnum_p; /* OUT: FileNet number */ (pg211.20)
  unsigned long  unslongnum; /* IN: unsigned long word */

/* Converts an unsigned long to a FileNet number
*/
```

+++ IMS - Image Management Service +++

20. IMS DECLARATIONS

=====
=====

IMS - Image Management Service

=====
=====

This module contains routines to facilitate access to an Image Management Service. It contains handle management routines, and also some miscellaneous functions.

```
#ifndef IMS_defs
#define IMS_defs
```

```
#ifndef SEC_defs
#include <SEC.defs>
#endif
```

```
typedef      unsigned short      IMS_svc_typ;
#define      IMS_num_svc_typs     8      /* #of values in IMS_svc_type */
#define      IMS_csm_svc_typ      0
#define      IMS_doc_svc_typ      1
#define      IMS_inx_svc_typ      2
#define      IMS_pri_svc_typ      3
#define      IMS_bes_svc_typ      4
#define      IMS_wqs_svc_typ      5
#define      IMS_sec_svc_typ      6
#define      IMS_sqi_svc_typ      7
```

```
#define IMS_err_invalid_IMS_handle \
    err_encode(err_IMS, 0, 0)
/*$M Invalid IMS handle. */
#define IMS_err_invalid_IMS_service_type \
    err_encode(err_IMS, 0, 1)
/*$M Invalid IMS service type. */
#define IMS_err_release_failed \
    err_encode(err_IMS, 0, 2)
/*$M Failed to release an IMS service handle */
#define IMS_err_reserve_failed \
```

+++ IMS - Image Management Service +++

```
    err_encode(err_IMS, 0, 3)
/*$M Failed to reserve an IMS service handle */
#define IMS_err_not_impl \
    err_encode(err_IMS, 0, 4)
/*$M IMS function not implemented this platform/release */

#endif
```

21. IMS SUBROUTINES

21.1. IMS_free_domain_names

```
void
IMS_free_domain_names (list_p)
    ASE_domain_name_typ* list_p;    /* IN: Array of names */           (pg486.33)

/* Returns the memory allocated in an IMS_list_domain_names call */
```

21.2. IMS_free_organization_names

```
void
IMS_free_organization_names (list_p)
    ASE_organization_name_typ* list_p;           (pg486.38)

/* Returns the memory allocated in a IMS_list_organizations call */
```

21.3. IMS_free_svc_names

```
void
IMS_free_svc_names (list_p)
    ASE_service_name_typ* list_p;    /* IN: Array of names */       (pg486.28)

/* Returns the memory allocated in an IMS_list_svc_names call. */
```

21.4. IMS_get_message

```
error_typ
IMS_get_message (ims_handle, error, maxlen, firstline, msg)
    ASE_session_number_typ ims_handle; /* IN: IMS handle or 0 */       (pg483.24)
    error_typ error; /* IN: Error tuple to get message for*/(pg493.26)
    long maxlen; /* IN: Maximum length to return. */
    bool firstline; /* IN: TRUE => return only first line (pg491.22)
```

+++ IMS - Image Management Service +++

```

                                of message; FALSE => return
                                all of message. */
char*          msg;          /* OUT: Message returned. */
/*
This function allows the caller to specify whether to retrieve the message
text using the local or the Image Services server version of error message
catalog. The local version of the message catalog must be used if the
caller has not logged on to the server. This is done by passing in a 0 for
the 'ims_handle'. The server version of error message catalog is the
preferred method when the user has logged on to the Image Services server
and that the server release number is higher than the client. This ensures
an up-to-date version of message text is returned.

If 'firstline' is set to true, then the entire message for a given error
tuple is returned. Otherwise, only the first line of message is returned.
The first line of this message is a brief description of the error, and
the subsequent lines provides detail on the error. Not all error
tuples have more than one line descriptions.

If the message retrieved is longer than maxlen bytes long, an error
will be returned by this routine. The recommended maxlen for entire
messages is 2000 bytes, and for one line messages is 81 bytes. */
```

21.5. IMS_get_svc_handle

error_typ

```
IMS_get_svc_handle (ims_handle, svc_name_p, svc_type, svc_handle_p)
  ASE_session_number_typ  ims_handle;          /* IN:  IMS handle */          (pg483.24)
  ASE_service_name_typ*   svc_name_p;         /* IN:  service name */      (pg486.28)
  IMS_svc_typ             svc_type;          /* IN:  service type */      (pg223.27)
  ASE_session_number_typ* svc_handle_p;       /* OUT: service handle */    (pg483.24)
```

```
/* Gets a service handle for the indicated service. Either logs on
to the service if the client has not logged on yet, or returns the
same handle from a prior call to this routine. Note that this entry
point is more efficient than issuing a sequence of logon and logoff
calls around a single entry point of a service.
```

Each process has its own pool of service logon handles, and this routine will not return the same handle to two different processes. There will be only one handle for each instance of a service in the handle pool. The handle pool will contain all handles created via this routine, and also the default service handles contained within the ims handle which are used when an ims handle is passed directly to a service.

If this routine is used to get a SEC service handle for a domain other

+++ IMS - Image Management Service +++

than the one given to IMS_logon, it's assumed that the user name, password, and terminal are unchanged from the IMS_logon call.

The handle pool will not contain handles created by calls to DOC_logon, CSM_logon, INX_logon, etc. The service logon routines are to be used to create additional instances of handles to the same service, if needed. */

21.6. IMS_get_version_number

error_typ

IMS_get_version_number(ims_name, version_p)

ASE_service_name_typ *ims_name; /* IN : Service name */

(pg486.28)

ASE_version_number_typ *version_p; /* OUT: Version number of the server*/(pg490.16)

/* This function returns the IMS version number of the server specified in
* ims_name.
*/

21.7. IMS_list_domain_names

error_typ

IMS_list_domain_names (ref_domain_p, pattern_p, num_domains_p, list_p_p)

ASE_domain_name_typ* ref_domain_p; /* IN: Domain to get list from */

(pg486.33)

ASE_domain_name_typ* pattern_p; /* IN: Pattern to find */

(pg486.33)

long* num_domains_p; /* OUT: Number of domains found */

ASE_domain_name_typ** list_p_p; /* OUT: Domain names found */

(pg486.33)

/* Gets a list of all domains known to the reference domain "ref_domain_p".
The domains known to the reference domain are configured on the
reference domain server using the Initialization and Configuration
tools.

The "pattern_p" may contain a "*" on the end of the domain name to match
all domains with a prefix equal to the characters prior to the "*".

If no services exist for the given input conditions, *num_domains_p
will be zero but no error will be returned by this routine.

The list returned will be allocated in local memory, and a pointer
to this memory will be returned in *list_p_p. This memory must be
freed with the IMS_free_domain_names described in this module
whenever this routine returns a non-zero value in *list_p_p (*list_p_p
will be returned as zero when this routine returns an error or when
*num_domains_p is returned as zero).

*/

21.8. IMS_list_organization_names

error_typ

```
IMS_list_organization_names (ref_domain_p, pattern_p, num_org_p, list_p_p)
  ASE_domain_name_typ*      ref_domain_p; /* IN: Domain to get list from */(pg486.33)
  ASE_organization_name_typ* pattern_p;    /* IN: Pattern to find */      (pg486.38)
  long*                      num_org_p;    /* OUT: # organizations found */
  ASE_organization_name_typ** list_p_p;    /* OUT: Organizations found */  (pg486.38)
```

/* Gets a list of all organizations known to the reference domain "ref_domain_p". The organizations known to the reference domain are configured on the reference domain server using the Initialization and Configuration tools.

The "pattern_p" may contain a "*" on the end of the organization name to match all organizations with a prefix equal to the characters prior to the "*".

If no services exist for the given input conditions, *num_organizations_p will be zero but no error will be returned by this routine.

The list returned will be allocated in local memory, and a pointer to this memory will be returned in *list_p_p. This memory must be freed with the IMS_free_organization_names described in this module whenever this routine returns a non-zero value in *list_p_p (*list_p_p will be returned as zero when this routine returns an error or when *num_organizations_p is returned as zero).

*/

21.9. IMS_list_svc_names

error_typ

```
IMS_list_svc_names (pattern_p, svc_type, num_objects_p, list_p_p)
  ASE_service_name_typ*      pattern_p;    /* IN: Match pattern. */      (pg486.28)
  IMS_svc_typ                svc_type;     /* IN: Type of service */    (pg223.27)
  long*                      num_objects_p; /* OUT: # of names returned. */
  ASE_service_name_typ**     list_p_p;    /* OUT: Array of names. */  (pg486.28)
```

/* Gets a list of all services of the specified type. The "pattern_p" may contain "*" on the end of the object name to match all services with a prefix equal to the characters prior to the "*".

If no services exist for the given input conditions, *num_objects_p will be zero but no error will be returned by this routine.

+++ IMS - Image Management Service +++

The list returned will be allocated in local memory, and a pointer to this memory will be returned in `*list_p_p`. This memory must be freed with the `IMS_free_svc_names` described in this module whenever this routine returns a non-zero value in `*list_p_p` (`*list_p_p` will be returned as zero when this routine returns an error or when `*num_objects_p` is returned as zero).

*/

21.10. IMS_logoff

error_typ

IMS_logoff (ims_handle)

ASE_session_number_typ ims_handle; /* IN: Handle allocated by IMS_logon */ (pg483.24)

/* Logs the user off of the IMS. All service handles allocated via `IMS_get_svc_handle` or `IMS_reserve_svc_handle` are invalidated with this call. This call should be made prior to exiting the application program. */

21.11. IMS_logon

error_typ

IMS_logon (user_name_p, password_p, terminal_p, ims_name_p, version,
ims_handle_p)

SEC_name_typ user_name_p; /* IN: Name of user logging on */ (pg371.25)

SEC_password_typ password_p; /* IN: User's password */ (pg371.24)

SEC_terminal_typ terminal_p; /* IN: Terminal logging in from */ (pg371.26)

ASE_service_name_typ* ims_name_p; /* IN: Name of the IMS. Note that (pg486.28)
the object name in this
structure can be a NULL
string to get the default
IMS of the domain. */

long version; /* IN: Always "IMS_version". */

ASE_session_number_typ* ims_handle_p; /* OUT: IMS handle returned. */ (pg483.24)

/* Logs the user on to the IMS. The `ims_handle` can then be used as an argument to any `CSM`, `DOC`, `INX`, `BES`, `PRI`, `WQS`, `SEC` routine which requires a handle from `CSM_logon`, `DOC_logon`, `INX_logon`, etc.

If a parent process logs on via `IMS_logon` and then forks, the child process may not use `"*ims_handle_p"` created by the parent. A fork and then an `exec` are recommended. Multiple threads are also not supported, and attempts to call any application service entry point when using threads may have unpredictable results.

`IMS_logon` will call `SEC_logon`, and the comments about inheriting logons

+++ IMS - Image Management Service +++

If parameter `svc_name_p` is NULL, this routine will use the default service name as its service name. When a handle allocated by `IMS_reserve_svc_handle` is no longer needed, a subsequent call to `IMS_release_svc_handle` should be made.

This routine is thread safe for NT only and supports multiple threads. However multiple processes are not supported.

*/

21.14. `IMS_shmat`

char*

`IMS_shmat` (SharedMemoryID, SharedMemoryFlag)

int SharedMemoryID; /* IN: Shared memory id shmget routine */
int SharedMemoryFlag; /* IN: Flags (same values as in shmat system call). */

/* This routine attaches a shared memory segment to the address space of the calling process. This routine should be used by applications calling any Application Service library instead of the system call "shmat", because use of this routine will insure that a client library can attach to a client library shared memory segment needed to execute a client library routine.

The value returned by this routine is the address at which the memory was attached.

This routine only applies and only need be used by System V versions of operating systems. */

21.15. `IMS_str_to_svc`

void

`IMS_str_to_svc` (object_p, service_p)

char *object_p; /* IN */
ASE_service_name_typ *service_p; /* OUT */

(pg486.28)

/* Convert a string into a three part name. */

21.16. `IMS_svc_to_str`

void

`IMS_svc_to_str` (svc_name_p, str_p)

ASE_service_name_typ *svc_name_p; /* IN */
SEC_name_typ str_p; /* OUT */

(pg486.28)

(pg371.25)

+++ IMS - Image Management Service +++

```
/*  
  Convert a three part name into a string. The string can be any format  
  listed below:  
  1. object  
  2. object:domain  
  3. object:domain:organization  
*/
```

22. INX DECLARATIONS

/*=====

INX - Index Services

=====

This interface allows an application to access the information contained in the Index Database. This database contains information about documents and folders which contain data the customer is saving, and also data dictionary information such as the definition of document classes, indexes, and other related information.

Several of the entry points in index services lock structures for inspection by the calling application. These entry points may have capability records associated with them (`INX_capability_typ`), or they may have no capability record in the interface but may use one internally. If two applications try to lock the same record at the same time, one will successfully obtain the lock, and the other will wait for the lock to be released for a short period of time, and if the release does not occur will receive an "INX_err_record_busy" error.

The client applications should try to avoid locking structures for long periods of time when other applications may try to inspect or modify this structure during this interval. If an application gets an `INX_err_record_busy`, however, it should unlock all locks it currently has, wait for a short period (depending on how long other applications keep records locked), and then retry the sequence of operations which preceded the busy error. Note that all locks must be relinquished when the retry is done, because otherwise a deadlock can occur because, for example, one application has record A locked and tries to lock B, but can't because another application has record B locked and tries to lock record A. */

/*
* C data definitions for Index Services
*/

/*
* Several structures here vary in size.
* These have as their last element an array of length one [1],
* or another variable length structure.
* The way C works, it is not possible to put anything AFTER
* the variable length part of a structure.

+++ INX - Index Services +++

```
*
* These structures are indicated by the word INX_VARYING
* in front of struct. (INX_VARYING is #defined to be <empty>.)
* Instances of these structures should NOT be declared.
* Instead, an appropriate amount of memory should be allocated
* (usually with getarea()) and a pointer to the structure
* pointed at the area.
*/
```

```
#ifndef INX_defs
#define INX_defs
```

```
#ifndef INX_err_h
#include <INX_err.h>
#endif
```

```
#ifndef SEC_defs
#include <SEC.defs>
#endif
```

```
#ifndef FP_defs
#include <FP.defs>
#endif
```

```
#ifndef AS_externals_h
#include <AS_externals.h>
#endif
```

```
#define INX_VARYING
```

```
/*      Format level for data dictionary:
*          controls description structures
*          which may be stored in PDBs.
*      Level 2: added menus to dictionary
*      Level 3: added description field to dcl_desc
*      Level 4: added source_flag to dcl_index_desc
*      Level 5: added no_catalog_flag, unused1,unused2
*      Level 6: added ixid to key_desc to be used internally
*/
```

```
#define INX_DICT_LEVEL 6
#define MAX_DMA_DISPLAY_NAME_LENGTH 30
```

```
/*
* Commonly used types
```

+++ INX - Index Services +++

```
*/

typedef ASE_session_number_typ INX_session_handle_typ;                (pg483.24)

typedef ASE_service_name_typ INX_dict_id_typ; /* May carry which db later */ (pg486.28)

typedef unsigned short INX_relational_op_typ; /* same as filter op */

/* Types of documents. */

typedef byte INX_doc_type_typ;                                        (pg491.23)
#define INX_NULL_DOC      (unsigned char)'0'
#define INX_IMAGE_DOC    (unsigned char)'\0'
#define INX_TEXT_DOC     (unsigned char)'1'
#define INX_FORM_DOC     (unsigned char)'2'
#define INX_MIXED_DOC    (unsigned char)'3'
#define INX_SEP_SHEET    (unsigned char)'9'
#define INX_ANNOT_DOC    (unsigned char)'4'
#define INX_OTHER_DOC    (unsigned char)'5'
#define INX_P8_MODEL_DOC (unsigned char)'6'

/* Document format and location types */

typedef char          INX_doc_format_typ[ASE_MAX_STR_LEN + 1];
typedef char          INX_doc_location_typ[ASE_MAX_STR_LEN + 1];

/* Document and Folder retention types

The retention attributes of a document or folder indicate how long the
document/folder will be retained in the database, and what to do with the
document/folder when it's removed from the database. */

/* The retention base indicates a base date used to determine when
to remove the document/folder from the database. The retention base
can have the following values and meanings:

ASE_REL_TO_CLOSING      - Remove the item after a prescribed period
                        of time from the closing date.
ASE_REL_TO_ENTRY        - Remove the item after a prescribed period of
                        time from the entry (creation) date.

The value in INX_retent_base_typ[0] is the retention base, and the
value in INX_retent_base_typ[1] is a pad character of '\0'.
*/

typedef char          INX_retent_base_typ[2];

/* The retention offset is the number of months when added to the base
```


+++ INX - Index Services +++

date which indicate the date at which the document/folder will be eligible to be removed from the database. */

typedef unsigned long INX_retent_offset_typ;

/* The retention disposition indicates what action to take when the retention period has expired. The values are:

- ASE_ARCHIVE - Put the document into an archive database, and then delete from the main index database.
- ASE_DELETE - Delete the document from the index database (document is not saved in any other database)

At a time which is convenient for the user, the archive creation or document deletion program is run, and this program will remove/archive all documents which are ready according to their retention parameters. Note that the documents/folders are not automatically removed/archived by the system--an operator must initiate the process by running the appropriate utility.

The value in INX_retent_disp_typ[0] is the retention disposition, and the value in INX_retent_disp_typ[1] is a pad character of '\0'.

*/

typedef char INX_retent_disp_typ[2];

/*

* Definitions of indexes for DIRs and keys: names, types, sizes

*/

typedef char INX_index_name_typ [ASE_MAX_IXNAME + 1];

typedef INX_index_name_typ INX_key_name_typ;

(pg235.31)

typedef unsigned short INX_index_id_typ;

#define INX_INVALID_INDEX_ID 0

#define INX_MAX_IXVALIDATION_LEN ASE_MAX_ANYNAMELEN

typedef char INX_index_validation_typ [INX_MAX_IXVALIDATION_LEN + 1];

typedef INX_index_validation_typ INX_validation_name_typ;

(pg235.38)

typedef unsigned short INX_dcl_id_typ; /* document class id */

/* Index types. Note that user defined indexes (those associated with an index class) can only be of type INX_VT_FPNUM, INX_VT_ASCII, INX_VT_DATE, or INX_VT_MENU. Only system indexes (non-user indexes) can be of all the types listed below. */

typedef unsigned short INX_value_type_typ;

+++ INX - Index Services +++

```
#define INX_VT_BOOLEAN      'A'
#define INX_VT_BYTE        'B'
#define INX_VT_UNNS_BYTE   'C'
#define INX_VT_SHORT       'D'
#define INX_VT_UNNS_SHORT  'E'
#define INX_VT_LONG        'F'
#define INX_VT_UNNS_LONG   'G'
#define INX_VT_FPNUM       '1'    /* FileNet NUMBER      */
#define INX_VT_ASCII       '2'    /* STRING              */
#define INX_VT_DATE        '8'    /* DATE               */
#define INX_VT_TIME        '3'    /* TIME               */
#define INX_VT_MENU        '4'    /* MENU SELECTION     */
#define INX_VT_NULL        '0'

/* Oracle types */

typedef unsigned short INX_ora_type_typ;

#define INX_OT_OTHER        0
#define INX_OT_CHAR        1
#define INX_OT_NUM_DEPRECATED 2 /*
 * STR 62314: this #define fails to
 * correctly distinguish between Oracle
 * "number" (floating point) and
 * "number(10)" (fixed, e.g. doc number or
 * date stamp)
 */
#define INX_OT_DATE        3
#define INX_OT_FPNUM       4 /* STR 62314: replace INX_OT_NUM with */
#define INX_OT_FIXED       5 /* new types INX_{OT_FPNUM, OT_FIXED} */

/* In-memory lengths of values of the various types.
These defines indicate the length of an index value, with the exception
that INX_LEN_BYTE includes 1 extra pad byte (for historical reasons). Note
that the routine INX_index_data_size should be used to determine the size of
the index value plus the number of pad bytes which are inserted at the
end of the index value. */

#define INX_LEN_BOOLEAN    2
#define INX_LEN_BYTE       2
#define INX_LEN_UNNS_BYTE  2
#define INX_LEN_SHORT      2
#define INX_LEN_UNNS_SHORT 2
#define INX_LEN_LONG       4
#define INX_LEN_UNNS_LONG  4
#define INX_LEN_FPNUM      16
```

+++ INX - Index Services +++

```
/* ASCII is variable length */
#define INX_LEN_LENGTH      2      /* String length occupies 2 bytes */
#define INX_LEN_DATE        4
#define INX_LEN_TIME        4
#define INX_LEN_MENU        2
#define INX_LEN_NULL        0
#define INX_LEN_VARIABLE    255    /* Signal that something is funny */
#define INX_LONG_STRING     256    /* This is what we handle */
#define INX_LONGER_STRING   4096

/*
 * o Within INX_doc_index_rec_typ records, index_values begin on either 2 or
 * 4 byte boundaries, depending on the machine on which this runs, and
 * the data in a single index is padded to a 2 or 4 byte boundary as is
 * necessary to keep the correct alignment.
 * o To advance a pointer in the INX_doc_index_rec_typ from one index to the
 * next, see the description of the "INX_index_data_size" subroutine. This
 * routine will calculate the correct amount of pad bytes.
 * o The index_id is 2 bytes.
 * o The type is 2 bytes.
 * o If the index type is a string (INX_VT_ASCII), then the first two bytes
 * of the "data" field are the length of the string. The string starts
 * in data[3] and is "length" bytes long. Note that this string is not
 * null terminated.
 * Normally the string used with INX_VT_ASCII is an ascii string, except
 * if the value of INX_index_value_typ.index_id equals INX_IN_ACCESSRIGHTS,
 * in which case the length will always be 12 and the string is actually
 * a binary string which should be copied to an SEC_access_restrictions_typ
 * structure (note that the long words in the SEC_access_restrictions_typ
 * are not long word aligned, so typecasting in place won't work for
 * platforms which require long on long alignment).
 * o If the index type is INX_VT_MENU, then data[0] is equal to the numeric
 * value of the menu (the "value" field from the INX_menu_item_typ
 * record), and data[1] is a null.
 * o The "data" field is typecast as follows for the remaining index types:
 *
 * INX_index_value_typ.type      Typecast "data" to
 * -----
 * INX_VT_BOOLEAN                FN_bool16_t
 * INX_VT_BYTE                   char
 * INX_VT_UNNS_BYTE              unsigned char
 * INX_VT_SHORT                  short
 * INX_VT_UNNS_SHORT             unsigned short
 * INX_VT_LONG                   long
 * INX_VT_UNNS_LONG              unsigned long
 * INX_VT_FPNUM                  FP_number
 * INX_VT_DATE                   ASE_date_typ
 * INX_VT_TIME                   ASE_time_typ

```

+++ INX - Index Services +++

```

*      INX_VT_NULL                (means no value specified for field)
*/

typedef INX_VARYING struct {
    INX_index_id_typ      index_id;          (pg235.34)
    INX_value_type_typ    type;             (pg235.48)
    char                  data[2]; /* type-dependent, can be longer */
} INX_index_value_typ;

/* Variable structure for general record ("raw row") */

typedef INX_VARYING struct INX_doc_index_rec_typ {
    unsigned short        value_len; /* Number of bytes in the values */
                                /* Does NOT include fixed fields */
    unsigned short        value_num; /* Number of values */
    INX_index_value_typ   values[1]; /* var. number of var. length items */ (pg238.8)
} INX_doc_index_rec_typ;
typedef INX_doc_index_rec_typ INX_dir_typ; /* Abbreviated name */ (pg238.17)

#define INX_dir_size(dir_p) \
    (((char *)dir_p->values - (char *) (dir_p)) + dir_p->value_len)

/*
* The following is an INX_index_value_typ without the index_id.
* This form appears in the constants of a filter.
* To determine the length of one of these records, use the routine
* 'INX_index_data_size'.
*/

typedef INX_VARYING struct {
    short                 unused; /* Use to long align "data" below */
    INX_value_type_typ    type;
    char                  data[2]; (pg235.48)
} INX_index_choice_typ;

/* The dups type gives a hint as to how many duplicates of an index
   exist, and is used as a hint to optimize queries. */

typedef unsigned short INX_dups_typ;
#define INX_DUPS_MANY 1
#define INX_DUPS_FEW 2
#define INX_DUPS_NONE 3

#define INX_MAX_MENU_NAME_LEN 14
typedef char INX_menu_name_typ [INX_MAX_MENU_NAME_LEN + 1];

#define INX_MAX_MENU_ITEM_NAME_LEN 40
typedef char INX_menu_item_name_typ [INX_MAX_MENU_ITEM_NAME_LEN + 1];

```

+++ INX - Index Services +++

```

typedef char INX_mask_typ [ASE_MAX_IXMASKLEN + 1];

#define GUID_REGISTRY_FORMAT_LENGTH 36
typedef char INX_guids_typ [GUID_REGISTRY_FORMAT_LENGTH + 1];
typedef struct INX_guid_list_typ {
    INX_guids_typ  guid;                                (pg239.5)
    struct INX_guid_list_typ  *next;                    (pg239.9)
} INX_guid_list_typ;

#define MAX_NUMBER_OF_GUIDS 10

typedef struct INX_index_desc_typ {
    struct INX_index_desc_typ  *next_p;                (pg239.34)
    INX_index_name_typ  name;                          (pg235.31)
    INX_index_name_typ  int_name;                      (pg235.31)
    INX_index_id_typ  id;                             (pg235.34)
    INX_index_id_typ  col_id;                         (pg235.34)
    FN_bool16_t  is_system;                           (pg491.12)
    INX_value_type_typ  type;                         (pg235.48)
    INX_ora_type_typ  ora_type;                       (pg236.17)
    INX_dups_typ  dups;                               (pg238.39)
    FN_bool16_t  upper;                               (pg491.12)
    unsigned short  max_string_len;
    INX_menu_name_typ  menu;                          (pg238.45)
    INX_mask_typ  mask;                              (pg239.2)
    char  description[ASE_MAX_IXDESCLEN + 1];
#ifdef MSDOS
    byte  fill1;                                     (pg491.23)
#endif
    unsigned long  numberguids;
    char  dma_name[MAX_DMA_DISPLAY_NAME_LENGTH + 1];
    INX_guids_typ  guid_list[MAX_NUMBER_OF_GUIDS];    (pg239.5)
} INX_index_desc_typ;

typedef struct INX_validation_desc_typ {
    struct INX_validation_desc_typ  *next_p; /* Linked list */ (pg239.41)
    INX_index_id_typ  id; (pg235.34)
    INX_validation_name_typ  valident; (pg235.39)
} INX_validation_desc_typ;

/*
 * Document Record Definitions
 */

typedef char INX_dcl_name_typ [ASE_MAX_DCNAME_SIZE + 1];

```

+++ INX - Index Services +++

```
/*
 * CE_ID type Object Store ID
 */
#define INX_CEID_UNDEFINED 0
#define INX_CEID_MIN      1000
typedef FN_int16_t  INX_ce_id_t;                                (pg491.6)

/*
 * Capability structure - basically a record lock
 */

/* values for capability type */
#define INX_CAP_AVAIL      0
#define INX_CAP_DOC        1
#define INX_CAP_FLD        2
#define INX_CAP_IDX        3
#define INX_CAP_DCL        4
#define INX_CAP_CLU        5
#define INX_CAP_TAB        6
#define INX_CAP_FORM_MENU  7
#define INX_CAP_FORM_VALTAB 8
#define INX_CAP_DCL_UNLOCK 15

typedef struct {
    long          type;
    long          id;
    long          timestamp;
    long          index;
} INX_capability_typ;

/*
 * INX_doc_desc: Fixed structure for FileNet fields
 * Data can be transferred here for convenience from doc_index_recs.
 *
 * INX_dir_to_doc_desc2() replaces obsolete entrypoint INX_dir_to_doc_desc();
 * Requires a format level defined below:
 *   Level 2: added doc_format and doc_location.
 *   Level 3: added ceid.
 */

/* Current INX_doc_desc_typ format level */
#define INX_DOC_DESC_LEVEL      3

typedef struct {
    ASE_doc_id_typ      doc_id;                                (pg483.37)
    INX_dcl_id_typ      dcl_id;                                (pg235.41)
    ASE_date_typ        entry_date;                            (pg489.28)
}
```

+++ INX - Index Services +++

```

char                dummy[2];          /* was annot_flag */
ASE_date_typ        archive_date;       (pg489.28)
ASE_date_typ        delete_date;        (pg489.28)
INX_retent_base_typ retent_base;         (pg234.46)
INX_retent_disp_typ retent_disp;         (pg235.25)
INX_retent_offset_typ retent_offset;     (pg235.4)
unsigned short      page_num;            /* Number of pages */
SEC_access_restrictions security;        (pg385.2)
INX_doc_type_typ    doc_type;            (pg234.11)
ASE_doc_id_typ      physical_doc_id;     (pg483.37)
/* physical_doc_id NO LONGER USED - delete on next version change */
FN_booll6_t         closed;              (pg491.12)
INX_doc_format_typ  doc_format;          (pg234.24)
INX_doc_location_typ doc_location;       (pg234.25)
INX_ce_id_t         ce_os_id;            (pg240.6)
} INX_doc_desc_typ;

```

```

/* Indexes are identified by number.
 * There is one set of numbers.
 * 1-30 are reserved for FileNet columns.
 * User columns start at 31 (leaving a gap) and are derived from the
 * names (A31, etc.).
 * If more than 30 FileNet columns are defined and these numbers are
 * recorded permanently, we'll have problems.
 * Otherwise, it may be possible to redefine the name->number function,
 * e.g., f(Annn) = nnn+10 instead of f(Annn) = n.
 *
 * The predetermined type for the item is given as a comment.
 */

```

```

#define INX_IN_DOCNUMBER          1      /* INX_VT_UNLONG */
#define INX_IN_DOCCLASSNUMBER     2      /* INX_VT_UNSHORT */
#define INX_IN_ENTRYDATE          3      /* INX_VT_DATE */
#define INX_IN_ANNOTATIONFLAG     4      /* INX_VT_BOOLEAN */
#define INX_IN_ARCHIVEDATE        5      /* INX_VT_DATE */
#define INX_IN_DELETEDATE         6      /* INX_VT_DATE */
#define INX_IN_RETENTBASE         7      /* INX_VT_UNBYTE */
#define INX_IN_RETENTDISP         8      /* INX_VT_UNBYTE */
#define INX_IN_RETENTOFFSET       9      /* INX_VT_UNLONG */
#define INX_IN_PAGES              10     /* INX_VT_UNSHORT */
#define INX_IN_DOCTYPE            11     /* INX_VT_UNBYTE */
#define INX_IN_ACCESSRIGHTS       12     /* INX_VT_ASCII */
#define INX_IN_DOCFORMAT          13     /* INX_VT_ASCII */
#define INX_IN_DOCLOCATION         14     /* INX_VT_ASCII */
#define INX_IN_CE_OSID            15     /* INX_VT_UNSHORT */
#define INX_IN_ACCESSRIGHTS_RD    16     /* INX_VT_LONG */
#define INX_IN_ACCESSRIGHTS_WR    17     /* INX_VT_LONG */
#define INX_IN_ACCESSRIGHTS_AX    18     /* INX_VT_LONG */

```

+++ INX - Index Services +++

```

/* The following columns are not stored in the resident database */
#define INX_IN_CLOSED          29      /* INX_VT_BOOLEAN */
/* PHYSICAL DOCID was      30 */

#define INX_IN_LAST_SYS        INX_IN_ACCESSRIGHTS_AX
#define INX_fake_sys_col(n)    ((n) >INX_IN_LAST_SYS && (n) <INX_IN_FIRST_USER)
#define INX_IN_FIRST_USER      31

typedef INX_VARYING struct INX_query_match_typ {
    struct INX_query_match_typ    *next_p;                (pg242.13)
    ASE_folder_id_typ             folder_id; /* Folder where found or 0 */ (pg484.13)
    INX_dir_typ                   dir;                    (pg238.18)
} INX_query_match_typ;
#define INX_dir_offset(match) ((char *)&(match).dir - (char *)&(match))

typedef unsigned short INX_key_id_typ;
#define INX_MAX_KEY_INDEXES      1          /* Bigger later */

typedef struct INX_key_desc_typ {
    struct INX_key_desc_typ    *next_p; /* Linked list */ (pg242.30)
    INX_key_id_typ             id;       (pg242.16)
    INX_key_name_typ           name;     (pg235.32)
#ifdef MSDOS
    byte                       fill1;    (pg491.23)
#endif
    unsigned short             index_num; /* Number of indexes */
    INX_index_id_typ           indexes[INX_MAX_KEY_INDEXES]; (pg235.34)
    short                      ixid; /* used internally by INX1 to store
                                     DBMS index id for key,if needed */
} INX_key_desc_typ;

/*
 * Menu Definitions
 *
 * A menu is a way to represent a set of character strings with a single
 * byte. For example, for a menu which has an INX_menu_name_typ of
 * "color", the "item_name"s in the INX_menu_item_typ record may be
 * "red", "green", "blue", and the corresponding "value"s in
 * INX_menu_item_typ may be 1, 2, or 3. When a document is stored, only
 * the value is saved, and then the display program can map the value
 * (1 to red, 2 to green, 3 to blue) to an ascii string.
 */

typedef struct INX_menu_item_typ {
    unsigned short             value; /* declared as a short, but value
                                     fits in 8 bits */
    INX_menu_item_name_typ    item_name; (pg238.48)
#ifdef MSDOS

```


+++ INX - Index Services +++

```

        byte                fill1;                                (pg491.23)
#endif
} INX_menu_item_typ;

typedef INX_VARYING struct INX_menu_desc_typ {
    struct INX_menu_desc_typ *next_p;                            (pg243.13)
    INX_menu_name_typ        name;                                (pg238.45)
#ifdef MSDOS
    byte                    fill1;                                (pg491.23)
#endif
    unsigned short          item_num;        /* Number of items */
    INX_menu_item_typ      items[1];        /* Probably more than one */ (pg243.3)
} INX_menu_desc_typ;

typedef INX_menu_desc_typ INX_menu_typ;        /* TEMP:  compatability */ (pg243.13)

typedef char INX_form_name_typ [ASE_MAX_DCFORMLLEN + 1];

/*
 * Family Definitions
 */

#define INX_MAX_FAM_NAME_LEN 18
typedef char  INX_fam_name_typ [INX_MAX_FAM_NAME_LEN + 1];
typedef unsigned long INX_fam_id_typ;

/*
 * Cluster Definitions
 */

typedef unsigned short INX_cluster_space_typ;
typedef unsigned short INX_cluster_id_typ[3];

#define INX_INVALID_CLUSTER_SPACE 0xffff

typedef struct {
    INX_cluster_space_typ  cluster_space;    (pg243.31)
    INX_cluster_id_typ    cluster_id;       (pg243.32)
} INX_cluster_key_typ;

#define INX_INVALID_CLUSTER_KEY {0, {0,0,0}}

/* Cluster index description type */
typedef struct INX_cluster_desc_typ {
    struct INX_cluster_desc_typ *next_p;    (pg244.3)
    INX_cluster_space_typ      id;          (pg243.31)
    INX_index_name_typ         index_name;  (pg235.31)
    INX_fam_name_typ           fam_name;    (pg243.24)

```

+++ INX - Index Services +++

```

    INX_fam_id_typ      fam_id;
    long               cluster_size;
} INX_cluster_desc_typ;

```

(pg243.25)

```

/*
 * Document Class Definitions
 */

/* Maximum size of optional_de string */
#define INX_MAX_OPT_DE      16

/* Defines for characters found in optional_de field */
#define INX_OPT_DE_VFY_IMAGES  '2'
#define INX_OPT_DE_VFY_INDEXES '6'
#define INX_OPT_DE_BATCH_TOTAL '7'

/* Defines for committal type */
/* -- committal type not used -- */

/* Data about an index relative to a document class. This structure defines
   a user index, and if the INX_index_desc_typ structure is retrieved
   corresponding to the index_id in the structure below, the is_system
   flag will be FALSE, and the type field will be either INX_VT_ASCII,
   INX_VT_DATE, INX_VT_MENU, or INX_VT_FPNUM. */

typedef struct {
    INX_index_id_typ index_id;
    FN_bool16_t      required;
    FN_bool16_t      batch_total_flag;
    FN_bool16_t      verify_flag;
    unsigned short   source_flag;
} INX_dcl_index_desc_typ;

```

(pg235.34)
(pg491.12)
(pg491.12)
(pg491.12)

```

/* Values for source_flag */
#define INX_SRC_UNKNOWN      0
#define INX_SRC_MANUAL      1
#define INX_SRC_OCR         2
#define INX_SRC_APERTURE    3

#define INX_MAX_DCL_INDEXES  224 /* Max indexes per doc class */

#define INX_MAX_DBCOLS      ASE_MAX_DBCOLS

typedef INX_VARYING struct INX_dcl_desc_typ {
    struct INX_dcl_desc_typ *next_p; /* Linked list */
    INX_dcl_name_typ      name; /* odd len */
    char                  description[ASE_MAX_DCDESCLEN+1]; /* odd len */
    INX_dcl_id_typ        id;

```

(pg245.41)
(pg239.47)
(pg235.41)

+++ INX - Index Services +++

```

    INX_form_name_typ      form;                                (pg243.17)
    char                  wfl_q[ASE_MAX_WFQUEUEUENAME + 1];
    char                  wfl_sys[ASE_MAX_WFSYSNAME + 1];
#ifdef MSDOS
    byte                  fill1;                                (pg491.23)
#endif
    INX_fam_id_typ        fam_id;          /* family id (note: if
                                        clustering used, this
                                        field is not applicable) */
                                        (pg243.25)
    INX_cluster_space_typ cluster_space; /* which cluster space */
                                        (pg243.31)
    INX_index_id_typ      cluster_index; /* if clustering used, id of
                                        cluster index, otherwise
                                        INX_INVALID_INDEX_ID */
                                        (pg235.34)

    SEC_access_restrictions security;                                (pg385.2)
    INX_retent_disp_typ    retent_disp;                                (pg235.25)
    INX_retent_base_typ    retent_base;                                (pg234.46)
    INX_retent_offset_typ  retent_offset;                                (pg235.4)
    FN_bool16_t            tab_out_flag;                                (pg491.12)
    long                  pages_per_doc;
    FN_bool16_t            verify_images;                                (pg491.12)
    FN_bool16_t            verify_indexes;                                (pg491.12)
    FN_bool16_t            incomplete;      /* repl committal_type */
                                        (pg491.12)
    FN_bool16_t            batch_total;                                (pg491.12)
    long                  batch_size;
    INX_fam_name_typ       fam_name;                                (pg243.24)
    FN_bool16_t            no_catalog_flag; /* don't catalog documents */
                                        (pg491.12)
    long                  migrate_delay; /* during committal,
                                        migrate to optical disk is
                                        delayed by this number of
                                        seconds by default,
                                        -1 implies never migrate to
                                        optical disk. */

    INX_ce_id_t            ce_os_id;                                (pg240.6)
    short                  unused3;
    unsigned long          numberguids;
    char                  dma_name[MAX_DMA_DISPLAY_NAME_LENGTH + 1];
    INX_guids_typ          guid_list[MAX_NUMBER_OF_GUIDS];            (pg239.5)
    /* the following indices are user indices only--no system indices*/
    unsigned short         index_num;      /* Number of indexes */
    INX_dcl_index_desc_typ indexes[1];    /* Variable length */
                                        (pg244.32)
} INX_dcl_desc_typ;

#define INX_MAX_APERCARDFILE_LEN ASE_MAX_ANYNAMELEN
typedef char INX_dcl_apercardfile_typ [INX_MAX_APERCARDFILE_LEN +1];
typedef INX_dcl_apercardfile_typ      INX_aperfile_name_typ;            (pg245.44)

#define INX_MAX_APERCARDIXLOC_LEN 60
typedef char      INX_aperixloc_data_typ[INX_MAX_APERCARDIXLOC_LEN+1];

```

+++ INX - Index Services +++

```
typedef struct INX_dcl_apercardixloc_typ {
    INX_index_id_typ      id;                (pg235.34)
    INX_aperixloc_data_typ aperixloc;       (pg245.48)
} INX_dcl_apercardixloc_typ;
```

```
typedef struct INX_aperfile_desc_typ {
    struct INX_aperfile_desc_typ *next_p; /* Linked list */ (pg246.11)
    INX_dcl_id_typ                id;        (pg235.41)
    INX_aperfile_name_typ         aperfile;  (pg245.45)
} INX_aperfile_desc_typ;
```

```
typedef struct INX_aperixloc_desc_typ {
    struct INX_aperixloc_desc_typ *next_p; /* Linked list */ (pg246.18)
    INX_dcl_id_typ                id;        (pg235.41)
    unsigned short                index_num;
    INX_dcl_apercardixloc_typ     aperixlocs[1]; (pg246.5)
} INX_aperixloc_desc_typ;
```

```
/*
 * Folder Definitions
 */
```

```
#define INX_MAX_FLD_LEVELS 8
#define INX_MAX_FLD_COMP_NAME 18
#define INX_MAX_FLD_NAME (INX_MAX_FLD_LEVELS * (INX_MAX_FLD_COMP_NAME+1))
#define INX_BEGINNING_OF_FOLDER 0
#define INX_END_OF_FOLDER 0xffffffff
```

```
/* A folder name consists of 1 to INX_MAX_FLD_LEVELS components separated
by slash characters. A component is a string of 1 to INX_MAX_FLD_COMP_NAME
characters of alphabets, decimal digits, and underscore characters.
The "folder" is the last component in the folder name. Folders may be
nested within one another.
```

```
For example, we can have folder names of "/a" and "/a/b". In this
example, we have folders "a" and "b", where folders "b" is within
folder "a". */
```

```
typedef char INX_folder_name_typ [INX_MAX_FLD_NAME + 1];
```

```
/* An INX_folder_spec_typ is the same as an INX_folder_name_typ, except that
it allows wildcards of the following form:
```

```
'*' refers to zero or more folder-name characters and may appear at the
end of the last component of the name.
'/' at the end of the name refers to the name specified and all folders
below it.
```

+++ INX - Index Services +++

Examples:

/i refers to the single folder called 'i'.
/i/ refers to 'i' and all folders in folder 'i'.
/i/J* refers to all folders beginning with J in folder i, and if a slash ("/") was appended to this string, it would refer to all folders beginning with J in folder i, and all folders within those folders

*/

typedef INX_folder_name_typ INX_folder_spec_typ; (pg246.40)

#define INX_ROOT_FOLDER_NAME "/"

#define INX_FOLDER_SEPARATOR '/'

/* A search position in the folder tree */

typedef struct {
 INX_folder_name_typ name; (pg246.40)
 ASE_doc_id_typ doc_id; (pg483.37)
} INX_folder_posn_typ;

typedef struct INX_folder_desc_typ {
 struct INX_folder_desc_typ *next_p; /* Linked list */ (pg247.41)
 ASE_folder_id_typ id; (pg484.13)
 INX_folder_name_typ name; (pg246.40)
#ifdef MSDOS
 byte fill1; (pg491.23)
#endif
 FN_bool16_t leaf; (pg491.12)
 FN_bool16_t non_leaf; (pg491.12)
 FN_bool16_t closed; (pg491.12)
 ASE_date_typ create_date; (pg489.28)
 ASE_date_typ archive_date; (pg489.28)
 ASE_date_typ delete_date; (pg489.28)
 short auto_del_period;
 SEC_access_restrictions security; (pg385.2)
 INX_retent_disp_typ retent_disp; (pg235.25)
 INX_retent_base_typ retent_base; (pg234.46)
 INX_retent_offset_typ retent_offset; (pg235.4)
} INX_folder_desc_typ;

typedef struct INX_folder_content_item_typ {
 ASE_folder_id_typ folder_id; (pg484.13)
 ASE_doc_id_typ doc_id; (pg483.37)
} INX_folder_content_item_typ;

typedef struct INX_fc_doc_ord_item_typ {

+++ INX - Index Services +++

```

        ASE_doc_id_typ  doc_id;
        char            ordinal[23];
} INX_fc_doc_ord_item_typ;

/*
 * DIR query definitions
 */

typedef unsigned short INX_query_direct_typ;
#define INX_QUERY_FORWARD      0

/*
 * Initial query positions: before the start or after the end
 */

#define INX_REC_BEGINNING      0
#define INX_REC_ENDING        0xffffffff

/*
 * For selecting open or closed docs and folders
 */

typedef unsigned short INX_closed_typ;
#define INX_CL_ACTIVE         0
#define INX_CL_CLOSED        1
#define INX_CL_ALL            2

/*
 * Key Condition definitions
 */

#define INX_KEY_SIMPLE         0
#define INX_KEY_RANGE         1

typedef INX_VARYING struct {
    INX_relational_op_typ  key_op;
    unsigned short         index_num;      /* Number of values */
    INX_index_value_typ    indexes[1];    /* variable length values */
} INX_key_comp_typ;

typedef INX_VARYING struct {
    unsigned short         key_id;
    unsigned short         key_cond_type; /* Simple or range (1 or 2 values) */
    INX_key_comp_typ       key_compare[1];
} INX_key_cond_typ;

/*
 * The following allows for two single index key comps

```

(pg483.37)

(pg234.7)

(pg238.8)

(pg248.39)

+++ INX - Index Services +++

```
* with INX_LONG_STRING string values.
*/
```

```
#define INX_MAX_KEY_COND \
( 2 + 2 + 2*(2 + 2 + 2 + INX_LEN_LENGTH + INX_LONG_STRING) )
/*      id type      op num id      len          val      */
```

```
#define INX_SERVICE_STR_LEN \
(ASE_MAX_OBJECT_LEN+1+ASE_MAX_DOMAIN_LEN+1+ASE_MAX_ORGANIZATION_LEN)
/* 1 is for ':' */
```

```
/*
 * Filter Ops
 */
```

```
typedef unsigned short INX_filter_op_code_typ;
#define INX_VALOP_INDEX 0
#define INX_VALOP_CONST 1
#define INX_RELOP_LSS 2
#define INX_RELOP_LEQ 3
#define INX_RELOP_EQL 4
#define INX_RELOP_GEQ 5
#define INX_RELOP_GTR 6
#define INX_RELOP_NEQ 7
#define INX_RELOP_LIKE 8
#define INX_BOOLOP_AND 9
#define INX_BOOLOP_OR 10
#define INX_BOOLOP_NOT 11
#define INX_MISCOP_HALT 12
#define INX_MAXOP 12 /* Highest value */
```

```
/* The INX_filter_op_typ record is a variable length record because the
" id " field may or may not exist. If the " op " field is INX_VALOP_INDEX
or INX_VALOP_CONST, then the id is present, but for any other values of
" op ", " id " is not present. An example of stepping through an array of
INX_filter_op_typ records is as follows:
```

```
long          tokens_num;
long          i;
INX_filter_op_typ* tokens_p;

for (i = 0; i < tokens_num; i++)
{
    switch (tokens_p->op)
    {
        case INX_VALOP_INDEX:
        case INX_VALOP_CONST:
            tokens_p++;
    }
}
```

+++ INX - Index Services +++

```
        break;
    default:
        tokens_p = (INX_filter_op_typ*)((unsigned long)tokens_p
            + sizeof (*tokens_p) - sizeof(tokens_p->id));
    }
}
*/

typedef INX_VARYING struct {
    INX_filter_op_code_typ  op;
    unsigned short         id; /* Identifies indexes and constants
                               Presence depends on op */
} INX_filter_op_typ;

/* Only records which satisfy the filter defined in the structure below
will be returned from a query. Note that the tokens_p and the
constants_p are pointers to variable length arrays of variable
size structures, so you cannot access array elements via
tokens_p[0], tokens_p[1], ... or constants_p[0], constants_p[1], ...
See descriptions of INX_filter_op_typ and INX_index_choice_typ to
determine how long each array element is. */

typedef struct {
    long                tokens_num; /* number of tokens */
    INX_filter_op_typ*  tokens_p;
    long                constants_num; /* number of constants */
    INX_index_choice_typ* constants_p;
#ifdef MSDOS
    /* Used for relocation of filter on a PC */
    long                tokens_offset;
    long                constants_offset;
#endif
} INX_filter_typ;
/*
 * The main DIR query structure
 */

typedef INX_VARYING struct {
    FN_bool16_t         continuation;
    unsigned short      max_recs;
    INX_query_direct_typ direction;
    unsigned long       current_rec;
    INX_closed_typ      doc_closed_filter;
    FN_bool16_t         use_folder;
    FN_bool16_t         use_key;
    FN_bool16_t         use_filter;
    INX_folder_spec_typ folder_orig;
    INX_folder_posn_typ folder_leftoff;
```

(pg249.16)

(pg250.13)

(pg238.34)

(pg491.12)

(pg248.9)

(pg248.23)

(pg491.12)

(pg491.12)

(pg491.12)

(pg247.11)

(pg247.21)

+++ INX - Index Services +++

```

    INX_closed_typ      folder_closed_filter;                (pg248.23)
    INX_filter_typ      filter;          /* Pointers to variable stuff */ (pg250.33)
    INX_key_cond_typ    key_cond;        /* Variable length */      (pg248.45)
} INX_query_typ;

/* Defines for check_level on import */

#define INX_VAL_DEF 0x01
#define INX_VAL_DCL 0x02
#define INX_VAL_ALL 0x0f

/* TAB defines and structures */

#define INX_MAX_TAB_LENGTH      240

typedef struct {
    ASE_folder_id_typ      folder_id;                (pg484.13)
    ASE_doc_id_typ         doc_id;                   (pg483.37)
    ASE_page_number_typ    page;                     (pg483.41)
    ASE_tab_id_typ         tab_id;                   (pg484.22)
    SEC_access_restrictions security;                (pg385.2)
    ASE_date_typ           last_mod;                 (pg489.28)
    unsigned short         tab_length;
    byte                   tab_text[INX_MAX_TAB_LENGTH+1]; (pg491.23)
} INX_tab_desc_typ ;

typedef INX_VARYING struct {                          /* use for archiving query only */
    unsigned short         rec_cnt;
    INX_folder_content_item_typ fol_doc[1];          (pg247.46)
} INX_query_miss_doc_typ;

/* DAM Menus and Validation Table structures: */

#define INX_MAX_FORM_MENU_ITEM_DESC_LEN 40
#define INX_MAX_FORM_MENU_DESC_LEN 177

typedef char INX_form_menu_item_code_typ;
typedef char INX_form_menu_item_desc_typ[INX_MAX_FORM_MENU_ITEM_DESC_LEN+1];
typedef char INX_form_menu_description_typ[INX_MAX_FORM_MENU_DESC_LEN+1];

typedef struct INX_form_menu_item_typ {
    INX_form_menu_item_code_typ code;                (pg251.37)
    INX_form_menu_item_desc_typ desc;                (pg251.38)
} INX_form_menu_item_typ;

typedef unsigned short INX_menu_id_typ;

typedef struct INX_form_menu_desc_typ {

```

+++ INX - Index Services +++

```

struct INX_form_menu_desc_typ  *next_p;                (pg252.11)
INX_menu_id_typ                id;                    (pg251.46) /* id of the menu */
INX_menu_name_typ              name;                  (pg238.45) /* name of the menu */
long                           usercode;             /* ?????? */
ASE_date_typ                   lastmod;              (pg489.28) /* create or modify time */
INX_form_menu_description_typ  desc;                 (pg251.39)
long                           translaterule; /* rule for translation */
char                           language;
unsigned short                 numitems;             /* no. of menu's items */
INX_form_menu_item_typ         *items;               (pg251.44) /* list of menu's items */
} INX_form_menu_desc_typ;

```

```

#define INX_MAX_FORM_VALTAB_NAME_LEN    ASE_MAX_ANYNAMELEN
#define INX_MAX_FORM_VALTAB_DESC_LEN    177
#define INX_MAX_VALTAB_ITEM_CODE_LEN    20
#define INX_MAX_VALTAB_ITEM_DESC_LEN    40

```

```

typedef char INX_form_valtab_description_typ[INX_MAX_FORM_VALTAB_DESC_LEN+1];
typedef char INX_form_valtab_item_code_typ [INX_MAX_VALTAB_ITEM_CODE_LEN+1];
typedef char INX_form_valtab_item_desc_typ [INX_MAX_VALTAB_ITEM_DESC_LEN+1];

```

```

typedef struct INX_form_valtab_item_typ {
    INX_form_valtab_item_code_typ  code;                (pg252.19)
    INX_form_valtab_item_desc_typ  desc;                (pg252.20)
} INX_form_valtab_item_typ;

```

```

typedef unsigned short INX_valtab_id_typ;
typedef char INX_valtab_name_typ[INX_MAX_FORM_VALTAB_NAME_LEN+1];

```

```

typedef struct INX_form_valtab_desc_typ {
    struct INX_form_valtab_desc_typ *next_p;            (pg252.39)
    INX_valtab_id_typ                id;                (pg252.27)
    INX_valtab_name_typ              name;              (pg252.28)
    INX_form_valtab_description_typ  desc;              (pg252.18)
    long                             maxcodesize;
    long                             maxdescsize;
    unsigned short                   numitems;
    INX_form_valtab_item_typ         *items;            (pg252.25)
} INX_form_valtab_desc_typ;

```

```

/* special purpose struct: contains document id and document class id
in DOCTABA.
*/

```

```

#define INX_MAX_DOC_DCL_ID_REC  100
#define INX_MAX_KEY_LEN  100
#define INX_MAX_FILTER_LEN     5120

```

```

typedef struct INX_doc_dcl_id_typ {

```

+++ INX - Index Services +++

```

    ASE_doc_id_typ      doc_id;          (pg483.37)
    INX_dcl_id_typ     dcl_id;          (pg235.41)
} INX_doc_dcl_id_typ;

```

```

/* INX_index_size is a macro which calls the INX_index_data_size
   subroutine and makes manipulation of index value pointers easier
   to write. */

```

```

#define INX_index_size(ip) (INX_index_data_size((ip)->type,(ip)->data) + \
                          ((unsigned long)((ip)->data) - (unsigned long)(ip)))

```

```

#define INX_MAX_CE_DOMAIN_NAME 64
#define INX_MAX_CE_HOST_ID 80
#define INX_MAX_CE_OBJECT_STORE_NAME 64
#define INX_MAX_CE_DIR_LOG_SIZE 2000
#define INX_MAX_CE_RETURN_REC 500
#define INX_CE_GUID_LENGTH 38
#define INX_CE_NULL_GUID "{00000000-0000-0000-0000-000000000000}"

```

```

typedef FN_byte_t INX_ce_domain_name_t[INX_MAX_CE_DOMAIN_NAME+1]; (pg491.10)
typedef FN_byte_t INX_ce_host_id_t[INX_MAX_CE_HOST_ID+1]; (pg491.10)
typedef FN_byte_t INX_ce_objectstore_name_t[INX_MAX_CE_OBJECT_STORE_NAME+1]; (pg491.10)

```

```

/* Use fnc_get_clock () to get current FN_et_nsec_t. */

```

```

typedef struct INX_ce_log_seq_t
{
    FN_int32_t seq_num1; (pg491.8)
    FN_int32_t seq_num2; (pg491.8)
}
INX_ce_log_seq_t;

```

```

typedef FN_int16_t INX_ce_action_t; (pg491.6)
#define INX_ce_action_insert 1
#define INX_ce_action_export 2
#define INX_ce_action_update 3
#define INX_ce_action_delete_doctaba_export 4
#define INX_ce_action_delete 5

```

```

typedef struct INX_ce_complete_t
{
    INX_ce_id_t ce_os_id; (pg240.6)
    INX_ce_log_seq_t seq_nums; (pg253.30)
    INX_ce_action_t action; (pg253.33)
    FN_bool_t cat_in_doctaba; (pg491.11)
    ASE_doc_id_typ doc_id; (pg483.37)
    error_typ err; (pg493.26)
}

```

+++ INX - Index Services +++

```
}
INX_ce_complete_t;

#define INX_EXPORT_LOG_LEVEL 1
typedef INX_VARYING struct INX_ce_rec_list_t
{
    struct INX_ce_rec_list_t      *next_p;           (pg254.12)
    INX_ce_complete_t            completion_rec;     (pg254.2)
    INX_dir_typ                  *old_dir_p;        (pg238.18)
    INX_dir_typ                  new_dir;          (pg238.18)
}
INX_ce_rec_list_t;

#define INX_ce_new_dir_offset(rec) ((char *)&(rec).new_dir - (char *)&(rec))

typedef struct INX_ce_os_info_t
{
    INX_ce_domain_name_t         ce_domain_name;     (pg253.20)
    FN_GUID_char8_t              ce_domain_guid;     (pg491.48)
    INX_ce_objectstore_name_t    object_store_name;  (pg253.22)
    FN_GUID_char8_t              object_store_guid;  (pg491.48)
    FN_bool_t                    default_object_store; (pg491.11)
}
INX_ce_os_info_t;

typedef struct INX_ce_dcl_info_guid_t
{
    INX_dcl_name_typ             dcl_name;           (pg239.47)
    FN_GUID_char8_t              default_ce_domain_guid; (pg491.48)
    FN_GUID_char8_t              default_object_store_guid; (pg491.48)
}
INX_ce_dcl_info_guid_t;

/* INX_ce_save/read_export_file definitions */

#define EXPORT_CFG_FILE_PATH fnc_path(FNSW_LOCAL_DRIVE, "/fnsw/local/sd/export/")
#define EXPORT_FILENAME fnc_path(FNSW_LOCAL_DRIVE, "/fnsw/local/sd/export/export.xml")

typedef struct INX_doc_class_export_t
{
    INX_dcl_name_typ             dcl_name;           (pg239.47)
    ASE_doc_id_typ              start_doc_id;        (pg483.37)
    ASE_doc_id_typ              end_doc_id;          (pg483.37)
    FN_bool_t                    delete_after_export; (pg491.11)
    FN_bool_t                    re_export;          (pg491.11)
    FN_GUID_char8_t              ce_object_store_guid; (pg491.48)
    INX_ce_objectstore_name_t    ce_object_store_name; (pg253.22)
    FN_GUID_char8_t              ce_domain_guid;     (pg491.48)
}
```

+++ INX - Index Services +++

```
    INX_ce_domain_name_t      ce_domain_name;      (pg253.20)
    FN_bool_t                 annot_only;          (pg491.11)
}
INX_doc_class_export_t;
```

#endif /* INX_defs */

```
/*
 * INX_err.h
 *
 * $Header: wal_manual,v 1.59 08/10/08 14:02:43 gteng Exp $
 */
```

```
#ifndef INX_err_h
#define INX_err_h
```

```
#define INX_err(_x,_z) err_encode(err_INX, _x, _z)
```

```
#define INX_XIN_PROTOCOL 0
#define INX_ZOTHER_ERROR 1
```

```
/*      Protocol Errors      */
```

```
#define INX_err_other                INX_err(0,INX_ZOTHER_ERROR)
/*$M      Real error tuple is parameter to this protocol error */
```

```
#define INX_err_invalid_handle        INX_err(0,2)
/*$M      Invalid session handle      */
```

```
#define INX_err_no_permission         INX_err(0,3)
/*$M      Permission denied           */
```

```
#define INX_err_session_busy          INX_err(0,4)
/*$M      Session in use              */
```

```
#define INX_err_dup_record            INX_err(0,5)
/*$M      Duplicate database entry     */
```

```
#define INX_err_no_record             INX_err(0,6)
/*$M      Requested record not found   */
```

```
#define INX_err_record_busy           INX_err(0,7)
/*$M      Record already locked        */
```

The caller should sleep for a second and retry this error a couple of times

+++ INX - Index Services +++

before giving up trying to lock the record. If this fails, then setting the 'override' flag is an option for some entry point calls */

```
#define INX_err_no_menu                INX_err(0,8)
/*$M    Specified menu does not exist */

#define INX_err_no_folder              INX_err(0,9)
/*$M    No folder with name and state specified exists */

#define INX_err_not_in_folder         INX_err(0,10)
/*$M    Document not filed in specified folder */

#define INX_err_already_in_folder     INX_err(0,11)
/*$M    Document already filed in specified folder */

#define INX_err_inv_query             INX_err(0,12)
/*$M    Query specification is invalid */

#define INX_err_no_query              INX_err(0,13)
/*$M    No query in progress */

#define INX_err_descendent_dest       INX_err(0,14)
/*$M    Cannot move/copy folder to its own descendent */

#define INX_err_no_capability         INX_err(0,15)
/*$M    No capability (lock) obtained for operation */

#define INX_err_inv_record            INX_err(0,16)
/*$M    Document index record is not valid */

#define INX_err_no_dcl               INX_err(0,17)
/*$M    Specified document class does not exist */

#define INX_err_no_index              INX_err(0,18)
/*$M    Specified index does not exist */

#define INX_err_reqd_is_null         INX_err(0,19)
/*$M    One or more required items is null */

#define INX_err_no_key               INX_err(0,20)
/*$M    Specified key does not exist */

#define INX_err_wrong_handle         INX_err(0,21)
/*$M    Active session needs same handle */

#define INX_err_date_conflict        INX_err(0,22)
/*$M    Conflicting dates in folder description */
```

+++ INX - Index Services +++

```
#define INX_err_inv_retent_base          INX_err(0,23)
/*$M   Invalid retention base          */

#define INX_err_not_imported            INX_err(0,24)
/*$M   DIR not imported                */

#define INX_err_doc_id_range            INX_err(0,25)
/*$M   Document id number out of permitted range */

#define INX_err_bad_pages                INX_err(0,26)
/*$M   Values for pages outside of allowed range */

#define INX_err_dcl_index_twice         INX_err(0,27)
/*$M   Index defined in document class twice */

#define INX_err_xs_dcl_indexes          INX_err(0,28)
/*$M   More than allowed number of indexes for document class */

#define INX_err_inv_sys_col             INX_err(0,29)
/*$M   System index has wrong type or value */

#define INX_err_unk_sys_col             INX_err(0,30)
/*$M   Unknown system column          */

#define INX_err_index_in_rec_twice      INX_err(0,31)
/*$M   Two values for index in doc index record */

#define INX_err_inv_retent_disp         INX_err(0,32)
/*$M   Invalid retention disposition  */

#define INX_err_bad_value_type          INX_err(0,33)
/*$M   Invalid index value type in doc index record */

#define INX_err_doc_is_filed            INX_err(0,34)
/*$M   Cannot delete document - still in folders */

#define INX_err_bad_direction           INX_err(0,35)
/*$M   Direction value in query is invalid */

#define INX_err_bad_current_rec         INX_err(0,36)
/*$M   Current record value in query is invalid */

#define INX_err_unknown_op              INX_err(0,37)
/*$M   Unknown query filter operator  */

#define INX_err_inv_source              INX_err(0,40)
/*$M   Unrecognized value for document source */
```

+++ INX - Index Services +++

```
#define INX_err_not_pdb                INX_err(0,41)
/*$M    Function is not implemented for portable database */

#define INX_err_query_non_index        INX_err(0,42)
/*$M    Cannot perform query on non-stored index */

#define INX_err_inv_dcl_name           INX_err(0,43)
/*$M    Invalid document class name */

#define INX_err_folder_closed          INX_err(0,44)
/*$M    Folder is closed */

#define INX_err_interrupted           INX_err(0,45)
/*$M    Query was interrupted */

#define INX_err_index_not_in_dcl       INX_err(0,46)
/*$M    Index in DIR not defined in document class */

#define INX_err_cannot_change_dcl      INX_err(0,47)
/*$M    DIR update cannot change document class */

#define INX_err_bad_cap_type           INX_err(0,48)
/*$M    Invalid capability type */

#define INX_err_too_deep               INX_err(0,49)
/*$M    Attempt to create too many folder levels */

#define INX_err_no_more_cols           INX_err(0,50)
/*$M    No more user columns available */

#define INX_err_bad_folder_atts        INX_err(0,51)
/*$M    Invalid value(s) in folder description */

#define INX_err_not_empty              INX_err(0,52)
/*$M    Deletion of non-empty folder (but not contents) requested */

#define INX_err_bad_folder_name        INX_err(0,53)
/*$M    Invalid folder name */

#define INX_err_no_sys_in_dcl          INX_err(0,54)
/*$M    Cannot define system indexes in document class */

#define INX_err_inv_cluster_col        INX_err(0,55)
/*$M    Invalid cluster index */

#define INX_err_no_cluster             INX_err(0,56)
/*$M    No cluster index is defined */
```


+++ INX - Index Services +++

```
#define INX_err_change_dcl_id          INX_err(0,57)
/*$M  Cannot change document class id */

#define INX_err_change_dcl_name       INX_err(0,58)
/*$M  Cannot change document class name */

#define INX_err_dup_dcl               INX_err(0,59)
/*$M  Document class already exists */

#define INX_err_bad_user_type         INX_err(0,60)
/*$M  Invalid type for user index */

#define INX_err_cluster_exists        INX_err(0,61)
/*$M  Index cluster already exists */

#define INX_err_invalid_cap_for_id    INX_err(0,62)
/*$M  Invalid capability for specified id */

#define INX_err_not_clu_cap           INX_err(0,63)
/*$M  Capability not for cluster */

#define INX_err_incomplete_dcl       INX_err(0,64)
/*$M  Document class not completely defined */

#define INX_err_duplicate_index       INX_err(0,65)
/*$M  Index already exists */

#define INX_err_not_idx_cap           INX_err(0,66)
/*$M  Capability not for index */

#define INX_err_already_inverted      INX_err(0,67)
/*$M  Index is already inverted */

#define INX_err_table_busy            INX_err(0,68)
/*$M  Operation is not allowed to a table which is in use by other process */

#define INX_err_not_inverted          INX_err(0,69)
/*$M  Index is not inverted */

#define INX_err_invalid_duplicate     INX_err(0,70)
/*$M  Invalid duplicates specification */

#define INX_err_doc_is_tabbed         INX_err(0,71)
/*$M  Cannot delete document - document is tabbed */

#define INX_err_sys_idx_not_allowed   INX_err(0,72)
/*$M  System index not allowed */
```

+++ INX - Index Services +++

```
#define INX_err_menu_not_allowed      INX_err(0,73)
/*$M   Menu not allowed for type      */

#define INX_err_mask_not_allowed      INX_err(0,74)
/*$M   Mask not allowed for type      */

#define INX_err_len_for_ascii_only    INX_err(0,75)
/*$M   Length allowed for strings only*/

#define INX_err_invalid_index_name    INX_err(0,76)
/*$M   Invalid index name             */

#define INX_err_invalid_security      INX_err(0,77)
/*$M   Invalid security name          */

#define INX_err_bad_row_buf           INX_err(0,78)
/*$M   Buf_len and row data are inconsistent */

#define INX_err_inv_ora_typ_idx       INX_err(0,80)
/*$M   Index has invalid oracle type  */

#define INX_err_index_aperture_differ INX_err(0,81)
/*$M   The Index ids in dcl desc and aperixlocs are different */

#define INX_err_share_cnt_ovfl        INX_err(0,82)
/*$M   The claim share count for dict_lock is overflow */

#define INX_err_share_cnt_udfl        INX_err(0,83)
/*$M   The claim share count for dict_lock is underflow */

#define INX_err_inv_query_archive     INX_err(0,84)
/*$M   Primary key condition is required to query Archive IMS */

#define INX_err_no_validation         INX_err(0,85)
/*$M   This index does not have validation table      */

#define INX_err_no_aperfile           INX_err(0,86)
/*$M   This document class does not have aperture card file table*/

#define INX_err_no_aperixloc          INX_err(0,87)
/*$M   This document class does not have multiple barcode information */

#define INX_err_duplicate_validation  INX_err(0,88)
/*$M   Validation already exists      */

#define INX_err_duplicate_aperfile    INX_err(0,89)
/*$M   Aperture Card File already exists */
```

+++ INX - Index Services +++

```
#define INX_err_duplicate_aperixloc      INX_err(0,90)
/*$M  Aperture card index location already exists */

#define INX_err_validation_not_allowed  INX_err(0,91)
/*$M  Validation not allowed with this index type */

#define INX_err_bad_menu_name           INX_err(0,92)
/*$M  The menu name must be alphanumeric */

#define INX_err_no_form_menu            INX_err(0,93)
/*$M  menu does not exist */

#define INX_err_bad_valtab_name         INX_err(0,94)
/*$M  The validation table name must be alphanumeric */

#define INX_err_no_form_valtab         INX_err(0,95)
/*$M  validation table does not exist */

#define INX_err_bad_valtab_numitems    INX_err(0,96)
/*$M  The number of items in validation_tab(_items) are different*/

#define INX_err_bad_menu_numitems      INX_err(0,97)
/*$M  The number of menu items in menu and menu_items are different*/

#define INX_err_arc_in_progress        INX_err(0,98)
/*$M  Update operation not allowed since archiving is in progress */

#define INX_err_cluster_not_zero_base  INX_err(0,99)
/*$M  Cluster space number is not zero relative: run dbupgrade */

#define INX_err_run_out_of_cluster_num INX_err(0,100)
/*$M  The highest cluster space number is reached */

#define INX_err_renum_ordinal          INX_err(0,101)
/*$M  Renumbering of folder ordinal failed */

#define INX_err_query_in_progress      INX_err(0,102)
/*$M  Operation not allowed since query is in progress */

#define INX_err_cluster_in_use        INX_err(0,103)
/*$M  Cannot delete cluster index while in use by document classes.
Check the error log for their names. */

#define INX_err_prec_scale_overflow    INX_err(0,104)
/*$M  precison and scale specified in numeric index mask cause overflow */

#define INX_err_logon_use_sct_names    INX_err(0,105)
/*$M  For clients, use_sct_names is not set when logon */
```

+++ INX - Index Services +++

```
#define INX_err_rename_duplicate_menu    INX_err(0,106)
/*$M    menu is renamed due to duplicate name */

#define INX_err_sct_name_too_long        INX_err(0,107)
/*$M    SCT name is too long */

#define INX_err_menu_ref_by_index        INX_err(0,108)
/*$M    Cannot delete menu while in use by an index(s) */

#define INX_err_valtab_ref_by_index      INX_err(0,109)
/*$M    Cannot delete validation table while in use by an index(s) */

#define INX_err_cor_reject_msg           INX_err(0,110)
/*$M    Courier reject message received, see error log for details */

#define INX_err_info                      INX_err(0,111)
/*$M    Please view error log for detail information */

/*      Non-protocol errors      */

#define INX_err_not_impl                  INX_err(1,1)
/*$M    Function is not implemented */

#define INX_err_cannot_convert            INX_err(1,2)
/*$M    Conversion from database type to INX type not supported */

#define INX_err_no_init                   INX_err(1,3)
/*$M    Could not initialize server */

#define INX_err_bad_net_data              INX_err(1,4)
/*$M    Incorrect data passed across network */

#define INX_err_string_too_long           INX_err(1,5)
/*$M    Received string which exceeds size of buffer */

#define INX_err_bad_fork                  INX_err(1,6)
/*$M    Fork of child process failed */

#define INX_err_bad_db                    INX_err(1,7)
/*$M    Bad data found in database */

#define INX_err_internal                  INX_err(1,8)
/*$M    Internal error in index services */

#define INX_err_no_dict_param             INX_err(1,9)
/*$M    Neither id nor name specified for dictionary get desc function */
```

+++ INX - Index Services +++

```
#define INX_err_bad_version          INX_err(1,10)
/*$M   Unrecognized version parameter on abst_link call */

#define INX_err_no_conn              INX_err(1,11)
/*$M   No Courier connection open   */

#define INX_err_have_conn            INX_err(1,12)
/*$M   Already have Courier connection open */

#define INX_err_inv_proc_num         INX_err(1,13)
/*$M   Unknown remote procedure number presented to server */

#define INX_err_inv_rpc_msg          INX_err(1,14)
/*$M   Unknown Courier msg_type     */

#define INX_err_dict_not_found       INX_err(1,15)
/*$M   No dictionary for specified id */

#define INX_err_bad_move_sys_col     INX_err(1,16)
/*$M   INX1 internal error in move_sys_col */

#define INX_err_bad_new_row_buf      INX_err(1,17)
/*$M   Call to expand non-existent buffer made */

#define INX_err_no_service           INX_err(1,18)
/*$M   Requested service name does not exist */

#define INX_err_old_service_def      INX_err(1,19)
/*$M   Unrecognized inx service definition level in NCH record */

#define INX_err_double_bg            INX_err(1,20)
/*$M   Only one INX background process to run per database */

#define INX_err_import_buf_len       INX_err(1,21)
/*$M   Ran off end of import buffer  */

#define INX_err_old_ims_def          INX_err(1,22)
/*$M   Unrecognized IMS definition in NCH record */

#define INXD_err_lock_conflict       INX_err(1,23)
/*$M   Attempt to get exclusive dictionary lock when lock is shared */

#define INX_err_not_impl_archive     INX_err(1,24)
/*$M   Archive is not implemented for this function */

#define INX_err_query_overflow       INX_err(1,25)
/*$M   Query result caused internal buffer to overflow */
```

+++ INX - Index Services +++

```
#define INX_err_exceed_page_size      INX_err(1,26)
/*$M   Exceed page size.  DB2(R) support only */

#define INX_err_invalid_page_size     INX_err(1,27)
/*$M   A non-supported DB2 page size is detected.
Image Services will not function correctly until after the page size is enlarged to 8K or
greater.
Contact DB Administrator to resolve this problem. If this is an upgrade, an export and
reimport
of the database may also be required. */

        /* INXp errors */

#define INX_err_dict_level_wrong      INX_err(3,4)
/*$M   PDB contains unknown dictionary level */

#define INX_err_pdb_folder            INX_err(3,5)
/*$M   Folders not implemented in pdbs */

#define INX_err_pdb_header            INX_err(3,6)
/*$M   Error reading PDB header:  bad size or magic number */

#define INX_err_pdb_negchunks         INX_err(3,7)
/*$M   Negative number of chunks computed from PDB header. */

#define INX_err_pdb_badoffset         INX_err(3,8)
/*$M   Bad offset in PDB header      */

#define INX_err_pdb_shortread        INX_err(3,9)
/*$M   Short read of PDB chunk      */

#define INX_err_pdb_numindexes       INX_err(3,10)
/*$M   Compound keys in PDBs not implemented */

#define INX_err_keycondtype          INX_err(3,11)
/*$M   Illegal key condition type   */

#define INX_err_pdb_tab              INX_err(3,12)
/*$M   Tabs not implemented in pdbs */

#define INX_err_pdb_form_menu        INX_err(3,13)
/*$M   Form menu not impleted in pdb */

#define INX_err_pdb_stgpoolsz        INX_err(3,14)
/*$M   String pool size exceeded    */

#define INX_err_pdb_numoolsz         INX_err(3,15)
/*$M   Number pool size exceeded    */

#define INX_err_pdb_stackuflo        INX_err(3,16)
```

+++ INX - Index Services +++

```
/*$M    Stack underflow                */
#define INX_err_pdb_stackoflo          INX_err(3,17)
/*$M    Stack overflow                  */
#define INX_err_pdb_pushwrong          INX_err(3,18)
/*$M    Push wrong element              */
#define INX_err_pdb_badtype            INX_err(3,19)
/*$M    Bad stack element type          */
#define INX_err_pdb_fconssz           INX_err(3,20)
/*$M    Filter constants pool overflow  */
#define INX_err_pdb_badfcons           INX_err(3,21)
/*$M    Bad filter constant number      */
#define INX_err_pdb_nostopper          INX_err(3,22)
/*$M    Stopper entry missing at end of buffer */
#define INX_err_pdb_form_valtab        INX_err(3,23)
/*$M    Form validation table not impleted in pdb */

/* IAF Library Errors - Reserverd */
#define INX_library_errors             INX_err(4,*)

/* Activity Logging Errors - Reserverd */
#define INX_err_actlog_count           INX_err(5,1)
/*$M    Activity Log count should never be greater than 1 per process.
Abort due to software logic errors. */

/* IS-CE integration Errors */
#define INX_err_ce_DIR_too_big          INX_err(6,2)
/*$M    Document index data exceeds maximum size for export to CE */
#define INX_err_ce_only_one_ce_export  INX_err(6,3)
/*$M    Only one CE Export process may run at one time. */
#define INX_err_ce_user_is_not_admin    INX_err(6,4)
/*$M    User must have administrative privilege to run CE export tool */
#define INX_err_ce_invalid_arguments    INX_err(6,5)
/*$M    Invalid command-line arguments were specified to the CE export tool */
#define INX_err_ce_export_file_not_found INX_err(6,6)
/*$M    The specified document class configuration file (ce export) was not found */
```

+++ INX - Index Services +++

```
#define INX_err_ce_default_IS_not_found    INX_err(6,7)
/*$M    The ce export tool was unable to find the default IS system for the domain */

#define INX_err_ce_default_IS_value_not_found    INX_err(6,8)
/*$M    The ce export tool was unable to retrieve a value for the default IS system */

#define INX_err_ce_index_server_desc_not_found    INX_err(6,9)
/*$M    The ce export tool was unable to retrieve a description for the Index Server */

#define INX_err_ce_index_server_address_not_found    INX_err(6,10)
/*$M    The ce export tool was unable to retrieve an address for the Index Server */

#define INX_err_ce_index_server_not_local    INX_err(6,11)
/*$M    The ce export tool or INX_ce_log_doc entry point must run locally on the Index
Server */

#define INX_err_ce_signals_undefined    INX_err(6,12)
/*$M    The ce export tool was unable to install signal handlers (undefined) */

#define INX_err_ce_unexpected_xml_format    INX_err(6,13)
/*$M    The ce export tool did not detect a valid XML tag */

#define INX_err_ce_array_out_of_bound    INX_err(6,14)
/*$M    The ce export tool encountered an invalid array index for an XML child tag */

#define INX_err_ce_invalid_xml_tag_value    INX_err(6,15)
/*$M    The ce export tool detected an invalid XML tag value */

#define INX_err_ce_duplicate_export_range    INX_err(6,16)
/*$M    The ce export tool detected document class entries with overlapping document ranges
*/

#define INX_err_ce_bad_signal    INX_err(6,17)
/*$M    The ce export tool was abnormally terminated */

#define INX_err_ce_doc_class_failed    INX_err(6,18)
/*$M    The ce export tool failed to process the document class */

#define INX_err_ce_info_doc_already_exported    INX_err(6,20)
/*$M    One or more document index records were previously exported to CE. */

#define INX_err_ce_os_guid_not_config    INX_err(6,21)
/*$M    The Object Store GUID has not been configured on this IS system. */

#define INX_err_ce_no_cached_info    INX_err(6,22)
/*$M    No Cached CE info in shared memory was found for the Object Store and host_id.
This may occur because Image Services has been re-cycled. */

#define INX_err_ce_no_completion_recs    INX_err(6,23)
/*$M    Completion records were expected from CE IS_import_agent but none were given.
```


+++ INX - Index Services +++

This may occur because the CE server was re-cycled. CE_LOG HWM will be reset to the zero value so the CE-IM IS_import_agent(s) may encounter some duplicates. */

```
#define INX_err_ce_completion_recs_not_consistent_w_info INX_err(6,24)
/*$M The CE info shared memory record is not consistent with
completed_recs passed in by the IS_import_agent. CE_LOG HWM will be reset
to the zero value so the CE IS_import_agent(s) may encounter some duplicates. */
```

```
#define INX_err_ce_id_not_found INX_err(6,25)
/*$M CE ID could not found in the CE_ID_MAP table. */
```

```
#define INX_err_ce_invalid_parameter INX_err(6,28)
/*$M Invalid parameter passed to internal routine. See sys_log for explanation. */
```

```
#define INX_err_ce_invalid_action INX_err(6,29)
/*$M Invalid or unsupported action. See sys_log for explanation. */
```

```
#define INX_err_ce_import_agent_info_timeout INX_err(6,30)
/*$M CE IS_import_agent info record has been timeout. */
```

```
#define INX_err_ce_invalid_update INX_err(6,31)
/*$M The CFS-IS Import Agent encounters some data error.
Refer to the error log on the Content Engine server for more information.
The export log entry will be deleted. The document needs to be re-exported. */
```

```
#define INX_err_ce_object_guid_duplicate INX_err(6,33)
/*$M Object Store GUID duplicate. Object Store GUID is being used
in by another CE Domains. */
```

```
#define INX_err_ce_domain_or_object_store_name_change INX_err(6,34)
/*$M CE Domain name or Object Store name was changed in the CE_ID_MAP table. */
```

```
#define INX_err_ce_delete_default_os INX_err(6,35)
/*$M Implicit delete of an Object Store is not allowed via the
INX_ce_config_object_store_dcl_map
entry point. The default Object Store must be first removed from IS by using the
RAC IS Catalog Export Tool prior to this. */
```

```
#define INX_err_ce_os_dcl_map_not_found INX_err(6,36)
/*$M The Object Store to DCL map must first be configured via EM before default Object
Store DCL
can be set. The Object Store GUID has not been configured for the DCL. There is
no entry in the CE_OS_DCL_MAP table. */
```

```
#define INX_err_ce_invalid_guid INX_err(6,38)
/*$M Invalid GUID format.
The following format is expected '{ - - - - }' */
```

```
#define INX_err_ce_export_xml_not_found INX_err(6,39)
/*$M The export.xml file does not exist */
```

+++ INX - Index Services +++

```
#define INX_err_ce_open_export_xml    INX_err(6,40)
/*$M    Could not open export.xml file */

#define INX_err_ce_no_export_data    INX_err(6,41)
/*$M    No document classes specified for export */

#define INX_err_ce_delete_export_xml    INX_err(6,42)
/*$M    Could not delete export.xml file */

#define INX_err_ce_append_status_export_xml    INX_err(6,43)
/*$M    Could not append status information to export_n.xml file */

#define INX_err_ce_rm_filter_off    INX_err(6,44)
/*$M    Indicates that Report Manager Filter is turned off. By default the
RM filter is turned off so that DIRs are exported to CE. */

#define INX_err_ce_dcl_default_os_not_config    INX_err(6,45)
/*$M    Document Class as indicated in the elog does not have a default Object Store
configured. Use the RAC IS Catalog Export Tool to set the default Object Store. */

#define INX_err_ce_rm_bad_magic    INX_err(6,46)
/*$M    The value of the magic number parameter is invalid */

#define INX_err_ce_invalid_domain_name    INX_err(6,47)
/*$M    Specified CE domain name cannot be found or has duplicate entries.
Please confirm the name is not NULL and correctly spelled.
For duplicate entries, consult your service representative for correction
actions. */

#define INX_err_ce_timezone_error    INX_err(6,48)
/*$M    Cannot determine time zone & daylight saving on the Root/Index Server.
Please confirm the TZ environment variable is correctly set for UNIX;
or the 'time zone' is correctly selected on Windows via Control Panel->
'Date and Time' option. */

#define INX_err_IA_config_map    INX_err(6,49)
/*$M    The CFS-IS Import Agent encounters some configuration error.
Refer to the error log on the Content Engine server for more information.
The export log entry will not be deleted. */

#define INX_err_IA_doc_not_processed    INX_err(6,50)
/*$M    This document was not processed by CFS-IS Import Agent. */

#endif
```

+++ INX - Index Services +++

23. INX SUBROUTINES

23.1. INX_check_dict_ts

```
error_typ
INX_check_dict_ts(ims_session)
  INX_session_handle_typ  ims_session; /* IN:  session number */           (pg234.3)

/* INX_check_dict_ts looks at the database and the in memory dictionary
  timestamps.  If they are different, the in memory dictionary
  is updated.

  ERRORS:
    No significant errors
*/
```

23.2. INX_close_DIR

```
error_typ
INX_close_DIR(ims_session, doc_id, cap_p)
  INX_session_handle_typ  ims_session; /* IN:  session number */           (pg234.3)
  ASE_doc_id_typ          doc_id;      /* IN:  document id */             (pg483.37)
  INX_capability_typ*     cap_p;       /* IN:  capability */             (pg240.30)

/* INX_close_DIR sets the "closed" status of a document to TRUE.
  It does so by examining the retention disposition and
  setting the archive/delete dates in such a way that
  the document becomes closed.

  Note that queries of documents can be done based on whether the
  document is closed or not.

  ERRORS
    INX_err_no_record
    INX_err_no_capability
*/
```

23.3. INX_close_connection

```
error_typ
INX_close_connection(ims_session)
  INX_session_handle_typ  ims_session; /* IN:  session number */           (pg234.3)

/* INX_close_connection causes an existing network connection to the
```

+++ INX - Index Services +++

server to be closed. The connection must have been opened by
INX_open_connection. */

23.4. INX_close_folder

```
error_typ
INX_close_folder(ims_session, folder_id, cap_p)
  INX_session_handle_typ  ims_session; /* IN:  session number */           (pg234.3)
  ASE_folder_id_typ       folder_id;   /* IN:  folder id */               (pg484.13)
  INX_capability_typ*     cap_p;        /* IN:  capability from prior      (pg240.30)
                                         INX_get_and_lock_folder call */
/* INX_close_folder sets the "closed" status of a folder to TRUE.
   It does so by examining the retention disposition and setting the
   archive/delete dates in such a way that the folder becomes closed.

   Note that queries of folders can be done based on whether the folder
   is closed or not.

   ERRORS:
     INX_err_no_folder
     INX_err_no_capability
*/
```

23.5. INX_copy_folder

```
error_typ
INX_copy_folder(ims_session, src, dest)
  INX_session_handle_typ  ims_session; /* IN:  session number */           (pg234.3)
  INX_folder_spec_typ     src;         /* IN:  folder(s) to copy */         (pg247.11)
  INX_folder_name_typ     dest;        /* IN:  folder to copy into */      (pg246.40)
/* INX_copy_folder copies the "src" folder into the "dest" folder. The
   new folder name is created by taking the last component of "src" and
   appending it to the name given in "dest".
```

Example:

```
src is "/a/b/c*"
dest is "/e/f"
result is "/e/f/c*"
```

Copying a folder to its own descendent is not allowed.

ERRORS:

```
INX_err_bad_folder_name
INX_err_dup_record
INX_err_no_folder
```

+++ INX - Index Services +++

```
    INX_err_descendent_dest
    INX_err_too_deep
*/
```

23.6. INX_create_DIR

```
error_typ
INX_create_DIR(ims_session, dir_p)
    INX_session_handle_typ  ims_session; /* IN:  session number */           (pg234.3)
    INX_doc_index_rec_typ*  dir_p;      /* IN:  Document Index Record (DIR) */ (pg238.17)
```

/* INX_create_DIR adds a new document index record to the index database. Any omitted indexes will receive the value null. Required items for the document class must have values.

ERRORS:

```
    INX_err_dup_record
    INX_err_reqd_is_null
    INX_err_string_too_long
    INX_err_bad_value_type
    INX_err_index_in_rec_twice
    INX_err_doc_id_range
    INX_err_incomplete_dcl
    INX_err_index_not_in_dcl
    INX_err_no_index
    INX_err_no_dcl
*/
```

23.7. INX_create_folder

```
error_typ
INX_create_folder(ims_session, folder_desc_p)
    INX_session_handle_typ  ims_session; /* IN:  session number */           (pg234.3)
    INX_folder_desc_typ*    folder_desc_p; /* IN:  folder description */      (pg247.41)
```

/* Creates a folder which can be used to contain documents.

In folder_desc_p, the fields are used as follows:

The following are provided by the client:

```
    name
    archive_date (depending on retent fields)
    delete_date (depending on retent fields)
    auto_del_period
    security
    retent_disp
```

+++ INX - Index Services +++

retent_base
retent_offset

The following are put into the database by the service:

id
create_date

The following fields are not used by this function:

next_p
leaf
non_leaf
closed
contents_time

ERRORS:

INX_err_dup_record
INX_err_invalid_security

*/

23.8. INX_delete_DIR

error_typ

INX_delete_DIR(ims_session, doc_id, unfile_doc)

INX_session_handle_typ ims_session; /* session number */ (pg234.3)
ASE_doc_id_typ doc_id; /* document id */ (pg483.37)
FN_bool16_t unfile_doc; /* TRUE=>unfile document from folder, (pg491.12)
if any; FALSE=>return error if
document is in folder */

/* INX_delete_DIR removes a document index record from the index database.

ERRORS:

INX_err_doc_is_filed (if unfile_doc is FALSE)
INX_err_record_busy
INX_err_no_permission

*/

23.9. INX_delete_DIRS

error_typ

INX_delete_DIRS(ims_session, num_docs, docs, unfile_doc, untab_doc)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)
int num_docs; /* IN: # docs to delete */
ASE_doc_id_typ docs[]; /* IN: array of document ids */ (pg483.37)
FN_bool16_t unfile_doc; /* IN: TRUE=>unfile doc from folder, (pg491.12)
FALSE=>return error if doc is

+++ INX - Index Services +++

```
FN_bool16_t          untab_doc;    /* IN:  TRUE=>delete tabs from folder. (pg491.12)
                                FALSE=>return error if doc
                                is tabbed. Note: unfile_doc
                                must be TRUE for untab_doc to
                                take effect */
```

/* INX_delete_DIRS removes document index record(s) from the doctaba.

ERRORS:

```
INX_err_no_record
INX_err_record_busy
INX_err_doc_is_filed
INX_err_doc_is_tabbed
INX_err_no_permission
```

*/

23.10. INX_delete_folder

error_typ

INX_delete_folder(ims_session, folder_spec, unfile_docs)

```
INX_session_handle_typ  ims_session; /* IN:  session number */      (pg234.3)
INX_folder_spec_typ     folder_spec; /* IN:  specifies folders to delete */ (pg247.11)
FN_bool16_t             unfile_docs; /* IN:  TRUE=>unfile docs if any,
                                FALSE=>return error if folder
                                not empty. */      (pg491.12)
```

/* INX_delete_folder deletes a folder. If "unfile_docs" is TRUE, all documents will be removed from the folder but the documents themselves will not be deleted.

ERRORS:

```
INX_err_no_permission
INX_err_not_empty
```

*/

23.11. INX_delete_folders_by_id

error_typ

INX_delete_folders_by_id(ims_session, folder_ids, num_rec)

```
INX_session_handle_typ  ims_session; /* IN:  session number */      (pg234.3)
ASE_folder_id_typ       folder_ids[]; /* IN:  array of folder ids */    (pg484.13)
unsigned short          num_rec;      /* IN:  # of record in array above */
```

/* Deletes the folders specified in the folder_ids array.

+++ INX - Index Services +++

ERRORS:

INX_err_no_folder

*/

23.12. INX_dir_to_doc_desc

error_typ

INX_dir_to_doc_desc(dir_p, dd_p)

INX_dir_typ* dir_p; /* IN: Document Index Record */

(pg238.18)

INX_doc_desc_typ* dd_p; /* OUT: document descriptor derived from DIR */

(pg241.16)

/* INX_dir_to_doc_desc copies system indexes from a DIR to a fixed structure.
This function is superceded by INX_dir_to_doc_desc2 and is here for
compatibility. */

23.13. INX_dir_to_doc_desc2

error_typ

INX_dir_to_doc_desc2(level, dir_p, dd_p)

short level; /* IN: Format level */

INX_dir_typ* dir_p; /* IN: Document Index Record */

(pg238.18)

INX_doc_desc_typ* dd_p; /* OUT: document descriptor derived from DIR */

(pg241.16)

/* INX_dir_to_doc_desc2 copies system indexes from a DIR to a fixed structure
Format levels for INX_doc_desc_typ:
level 2: added doc_format and doc_location */

23.14. INX_file_doc

error_typ

INX_file_doc(ims_session, doc_id, folder_id)

INX_session_handle_typ ims_session; /* IN: session number */

(pg234.3)

ASE_doc_id_typ doc_id; /* IN: document id */

(pg483.37)

ASE_folder_id_typ folder_id; /* IN: folder id */

(pg484.13)

/* INX_file_doc puts document "doc_id" into folder "folder_id". The
document is put at the end of the folder.

ERRORS:

INX_err_no_record

INX_err_no_folder

INX_err_record_busy

*/

+++ INX - Index Services +++

23.15. INX_file_doc_after

error_typ

```
INX_file_doc_after(ims_session, folder_id, num_docs, docs, after_doc)
  INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)
  ASE_folder_id_typ folder_id; /* IN: folder id */ (pg484.13)
  short num_docs; /* IN: number of docs in "docs" */
  ASE_doc_id_typ docs[]; /* IN: ordered array of document ids */ (pg483.37)
  ASE_doc_id_typ after_doc; /* IN: document to file docs after */ (pg483.37)
```

/* This routine files one or more docs into the folder in the position immediately after the "after_doc", in the order specified in the "docs" array. To put the document at the start of the folder, use INX_BEGINNING_OF_FOLDER for "after_doc", and to put it at the end of the folder, use either the last document in the folder or INX_END_OF_FOLDER.

ERRORS:

```
  INX_err_no_record
  INX_err_no_folder
  INX_err_record_busy
```

*/

23.16. INX_find_DIRS

error_typ

```
INX_find_DIRS(ims_session, query_p, match_p_p, folder_left_off_p,
              current_rec_p, last_match_p)
  INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)
  INX_query_typ* query_p; /* IN: query specification */ (pg251.4)
  INX_query_match_typ** match_p_p; /* OUT: linked list of DIRS (pg242.13)
                                     returned. */
  INX_folder_posn_typ* folder_left_off_p; /* IN/OUT: NULL pointer if query (pg247.21)
                                             is not folder restricted.
                                             Otherwise pass in pointer
                                             to zeroed structure on first
                                             call, and result from prior
                                             call on subsequent calls. */
  long* current_rec_p; /* IN/OUT: Pointer to 0 on first
                          call. Pass in result of
                          prior call on subsequent
                          calls. */
  FN_bool16_t* last_match_p; /* OUT: If have all records for (pg491.12)
                               current query. */
```

/* INX_find_DIRS retrieves a set of document index records from the database, returning them as a linked list.

+++ INX - Index Services +++

The memory for list in `match_p_p` is allocated by INX, and the client must deallocate this memory when done with it by calling `INX_free_matches`.

If `*last_match_p` is returned TRUE, then all records have been retrieved for the query and no further calls to this routine should be done. If `*last_match_p` is returned FALSE, however, then a subsequent call should be made this routine to complete the query, or else `INX_terminate_query` or `INX_logoff` should be called to terminate the query and free up query related resources.

See also the description of `PRS_parse` to facilitate building the `INX_query_typ` record.

ERRORS

`INX_err_inv_query`
`INX_err_no_query`

*/

23.17. `INX_find_folders`

`error_typ`

```
INX_find_folders(ims_session, folder_spec, continuation,
                 closed_filter, old_left_off_p, max_folders,
                 folders_p_p, new_left_off_p, last_folder_p)
INX_session_handle_typ  ims_session; /* IN: session number */ (pg234.3)
INX_folder_spec_typ     folder_spec; /* IN: folder specification */ (pg247.11)
FN_bool16_t             continuation; /* IN: if query is a continuation */ (pg491.12)
INX_closed_typ          closed_filter; /* IN: only retrieve folders with
                                     this closed status */ (pg248.23)
INX_folder_name_typ     old_left_off_p; /* IN: Passed in as anything
                                     on first call (continuation
                                     is FALSE). Pass in
                                     new_left_off_p from prior
                                     call on subsequent calls. */ (pg246.40)
short                   max_folders; /* IN: max # folders to return */
INX_folder_desc_typ**   folders_p_p; /* OUT: folders returned */ (pg247.41)
INX_folder_name_typ     new_left_off_p; /* OUT: where query stopped */ (pg246.40)
FN_bool16_t*            last_folder_p; /* OUT: if query is done */ (pg491.12)
```

/* `INX_find_folders` returns a linked list of descriptions of folders contained within a parent folder.

The memory for each element of the list is allocated by INX, and the client must deallocate this memory when done with the data by calling `INX_free_folders`.

+++ INX - Index Services +++

If `*last_folder_p` is returned as `FALSE` and the client does not call this entry point again and also does not call `INX_logoff`, then `INX_terminate_query` should be called to terminate this query and free up system resources (the server process).

ERRORS:

`INX_err_no_folder`
`INX_err_folder_is_leaf`

*/

23.18. `INX_free_DIR`

`error_typ`

`INX_free_DIR(ims_session, dir_p)`

`INX_session_handle_typ` `ims_session;` /* IN: session number */ (pg234.3)
`INX_dir_typ*` `dir_p;` /* IN: DIR record */ (pg238.18)

/* `INX_free_DIR` de-allocates the memory occupied by one doc index record.
It is intended to be used to free memory allocated by
`INX_get_single_DIR`.
*/

23.19. `INX_free_folder_ids`

`error_typ`

`INX_free_folder_ids(ims_session, folder_ids_p)`

`INX_session_handle_typ` `ims_session;` /* IN: session number */ (pg234.3)
`ASE_folder_id_typ**` `folder_ids_p;` /* IN: folder id list */ (pg484.13)

/* `INX_free_folder_ids` frees up the memory allocated in any call that
returns a list of folder IDs (such as `INX_where_filed`). */

23.20. `INX_free_folders`

`error_typ`

`INX_free_folders(ims_session, folders_p)`

`INX_session_handle_typ` `ims_session;` /* IN: session number */ (pg234.3)
`INX_folder_desc_typ*` `folders_p;` /* IN: head of linked list of folders */ (pg247.41)

/* `INX_free_folders` de-allocates the memory occupied by a linked list
of folder descriptions. */

23.21. `INX_free_matches`

+++ INX - Index Services +++

error_typ

INX_free_matches(ims_session, matches_p)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)

INX_query_match_typ* matches_p; /* IN: match list returned by a prior call to a query routine */ (pg242.13)

/* INX_free_matches de-allocates the memory occupied by a linked list of query match records.

It is intended to be used to free memory allocated by INX_find_DIRS.
*/

23.22. INX_get_all_cluster_descs

error_typ

INX_get_all_cluster_descs(ims_session, list_p_p)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)

INX_cluster_desc_typ** list_p_p; /* OUT: linked list of cluster descriptions. */ (pg244.3)

/* INX_get_all_cluster_descs returns a pointer to a linked list of descriptions of all clusters in the database.

The list resides in global memory and is to be treated as read-only by the client.
*/

23.23. INX_get_all_dcl_descs

error_typ

INX_get_all_dcl_descs(ims_session, list_p_p)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)

INX_dcl_desc_typ** list_p_p; /* OUT: linked list of all document class descriptions */ (pg245.41)

/* INX_get_all_dcl_descs returns a pointer to a linked list of all the descriptions of document classes (dcl's) in the database. Each class description, in turn, contains an array of document class index descriptions, for the indexes which pertain to the class.

The list resides in global memory and is to be treated as read-only by the client (do not modify or return).

ERRORS:

No significant errors

+++ INX - Index Services +++

*/

23.24. INX_get_all_index_descs

```
error_typ
INX_get_all_index_descs(ims_session, list_p_p)
  INX_session_handle_typ ims_session; /* IN: session number */           (pg234.3)
  INX_index_desc_typ**   list_p_p;   /* OUT: linked list of index       (pg239.34)
                                     descriptions */
/* INX_get_all_index_descs returns a pointer to a linked list of
descriptions of all indexes in the database.
```

The list resides in global memory and is to be treated as read-only by the client (do not modify or return).

ERRORS:

No significant errors.

*/

23.25. INX_get_all_key_descs

```
error_typ
INX_get_all_key_descs(ims_session, list_p_p)
  INX_session_handle_typ ims_session; /* IN: session number */           (pg234.3)
  INX_key_desc_typ**     list_p_p;   /* OUT: pointer to linked list of  (pg242.30)
                                     key descriptions. */
```

/* INX_get_all_key_descs returns a pointer to a linked list of descriptions of all keys in the database. Each key_desc contains an array of the index id's which make up the key and a count of the number of elements.

The list resides in global memory and is to be treated as read-only by the client (do not modify or return).

*/

23.26. INX_get_all_menu_descs

```
error_typ
INX_get_all_menu_descs(ims_session, list_p_p)
  INX_session_handle_typ ims_session; /* IN: session number */           (pg234.3)
  INX_menu_desc_typ**   list_p_p;   /* OUT: linked list of all menus */ (pg243.13)
```

/* INX_get_all_menu_descs returns a pointer to a linked list of descriptions of all menus in the database.

+++ INX - Index Services +++

The list resides in global memory and is to be treated as read-only by the client (not modified or returned by client).

ERRORS:

No significant errors

*/

23.27. INX_get_and_lock_folder

error_typ

```
INX_get_and_lock_folder(ims_session, folder_id, override, cap_p, folder_p)
  INX_session_handle_typ  ims_session; /* IN: session number */           (pg234.3)
  ASE_folder_id_typ       folder_id; /* IN: folder id */                 (pg484.13)
  FN_bool16_t             override; /* IN: if want to override existing
                                  lock */                               (pg491.12)
  INX_capability_typ*     cap_p; /* OUT: capabilities record */          (pg240.30)
  INX_folder_desc_typ*    folder_p; /* OUT: folder description */        (pg247.41)
```

/* INX_get_and_lock_folder returns the attributes of a folder and locks it for update.

ERRORS:

INX_err_no_folder
INX_err_record_busy

*/

23.28. INX_get_cluster_desc

error_typ

```
INX_get_cluster_desc(ims_session, id, index_name, desc_p_p)
  INX_session_handle_typ  ims_session; /* IN: session number */           (pg234.3)
  INX_cluster_space_typ  id; /* IN: cluster space number or
                              INX_INVALID_CLUSTER_SPACE to
                              use name to specify cluster. */          (pg243.31)
  INX_index_name_typ     index_name; /* IN: name of index on which
                              clustering is done (not used if id
                              is not a valid cluster space#) */        (pg235.31)
  INX_cluster_desc_typ** desc_p_p; /* OUT: cluster description */        (pg244.3)
```

/* INX_get_cluster_desc finds the description of a particular cluster index and returns a pointer to it.

The client is expected to treat this area as read-only (not modified or returned by client).

+++ INX - Index Services +++

ERRORS:

INX_err_no_cluster
INX_err_no_dict_param

*/

23.29. INX_get_dcl_desc

error_typ

```
INX_get_dcl_desc(ims_session, id, name_p, desc_p_p)
  INX_session_handle_typ  ims_session; /* IN: session number */           (pg234.3)
  INX_dcl_id_typ          id;          /* IN: document class id, or 0 to   (pg235.41)
                                     use name to specify class. */
  char*                   name_p;      /* IN: name of document class (not
                                     used if "id" != 0) */
  INX_dcl_desc_typ**      desc_p_p;    /* OUT: description of class */     (pg245.41)
```

/* INX_get_dcl_desc finds the description of a particular document class description (dcl) and returns a pointer to it. The client is expected to treat this area as read-only (should not be modified or returned).

ERRORS:

INX_err_no_dcl
INX_err_no_dict_param

*/

23.30. INX_get_dict_ts

error_typ

```
INX_get_dict_ts(ims_session, local_ts_p, server_ts_p)
  INX_session_handle_typ  ims_session; /* IN: session number */           (pg234.3)
  long*                   local_ts_p; /* OUT: timestamp of local cache */
  long*                   server_ts_p; /* OUT: timestamp of server cache */
```

/* This entry point is intended primarily for debugging, not for normal client use.

INX_get_dict_ts retrieves the update-level-identifying timestamps of the data dictionary. Two values are returned - the locally cached value and the server's value. These timestamps indicate the last time the data dictionary was updated, and will be different if the data dictionaries are not the same (for example, an update was done on the server, but the change has not yet propagated to the client workstation).

ERRORS:

+++ INX - Index Services +++

No significant errors

*/

23.31. INX_get_docs_in_folder

error_typ

INX_get_docs_in_folder(ims_session, folder_id, start_rec_p, match, max_rec,
entry_num_p, last_match_p)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)
ASE_folder_id_typ folder_id; /* IN: folder id */ (pg484.13)
INX_fc_doc_ord_item_typ* start_rec_p; /* IN: Records with values greater
than *start_rec_p will be
returned. To start at the
beginning of a folder, set
start_rec_p->ordinal[0] to
the integer zero ('\0'). */ (pg248.3)
INX_fc_doc_ord_item_typ match[]; /* OUT: documents in folder */ (pg248.3)
unsigned short max_rec; /* IN: max # of records to return
A value of 100 is used if this
field is input > 100 */
unsigned short* entry_num_p; /* OUT: # records returned */
FN_bool16_t* last_match_p; /* OUT: if have all records for query */(pg491.12)

/* INX_get_docs_in_folder returns a list of documents which are in the
folder specified by folder_id. "*last_match_p" is returned TRUE if no more
documents exist for the given query, and FALSE if *start_rec_p may be
set to "match[*entry_num_p-1]" to continue the query and get additional
documents in the folder.

If no documents exist in the folder, this routine returns
*last_match_p=TRUE, *entry_num_p=0, and a return code of 0 (no error)

It is the client's responsibility to allocate and deallocate
memory for the "match" array.

ERRORS:

No significant errors

*/

23.32. INX_get_folder_atts

error_typ

INX_get_folder_atts(ims_session, folder_id, folder_name, folder_p)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)
ASE_folder_id_typ folder_id; /* IN: folder id, or 0 to use
folder_name to specify folder*/ (pg484.13)

+++ INX - Index Services +++

```
char*          folder_name; /* IN:  folder name (not used if
                               folder_id != 0) */
INX_folder_desc_typ*  folder_p; /* OUT: folder description */ (pg247.41)

/* INX_get_folder_atts returns the attributes of a folder
```

```
ERRORS:
    INX_err_no_folder
*/
```

23.33. INX_get_folder_content_desc

```
error_typ
INX_get_folder_content_desc(ims_session, current_date, match, max_rec,
                             entry_num_p, current_rec_p, last_match_p)
INX_session_handle_typ  ims_session; /* IN:  session number */ (pg234.3)
ASE_date_typ            current_date; /* IN:  only folders with auto (pg489.28)
                               deldate < current_date will
                               be returned. */
INX_folder_content_item_typ match[]; /* OUT: folders returned */ (pg247.46)
unsigned short          max_rec; /* OUT: max folders to return.
                               set to 100 if > 100 */
unsigned short*         entry_num_p; /* OUT: # folders returned */
unsigned long*          current_rec_p; /* IN/OUT: *current_rec_p should
                               be zero on the first call,
                               and then subsequent calls
                               should be passed the value
                               returned on the prior
                               call */
FN_bool16_t*           last_match_p; /* OUT: TRUE if no more folders (pg491.12)
                               exist for given query */
```

```
/* INX_get_folder_content_desc retrieves a set of filed documents whose
auto-delete dates are less than "current_date".
```

It is the client's responsibility to allocate and deallocate memory for the match set.

If *last_match_p is returned as FALSE and the client does not call this entry point again and also does not call INX_logoff, then INX_terminate_query should be called to terminate this query and free up system resources.

```
ERRORS:
    No significant errors
*/
```

+++ INX - Index Services +++

23.34. INX_get_folders_by_archivedate

error_typ

```
INX_get_folders_by_archivedate(ims_session, start_arch_date, end_arch_date,
                               match, max_rec, entry_num_p, current_rec_p, last_match_p)
  INX_session_handle_typ  ims_session;      /* IN: session number */          (pg234.3)
  ASE_date_typ            start_arch_date; /* IN: start date */              (pg489.28)
  ASE_date_typ            end_arch_date;   /* IN: end date */                (pg489.28)
  INX_folder_desc_typ     match[];         /* OUT: folder descriptions */    (pg247.41)
  unsigned short          max_rec;         /* IN: max # folders to return. Set
                                         to 100 if > 100. */
  unsigned short*         entry_num_p;     /* OUT: num folders returned */
  unsigned long*          current_rec_p;   /* IN/OUT: initially passed in as
                                         a pointer to zero, and then
                                         the output value is passed in
                                         as the next input to resume
                                         query where it left off */
  FN_bool16_t*           last_match_p;    /* OUT: TRUE if no more folders exist
                                         for given query */
```

/* INX_get_folders_by_archivedate returns a set of folders whose archived dates are <= end_arch_date and >= start_arch_date.

It is the client's responsibility to allocate and deallocate memory for the match set.

If *last_match_p is returned as FALSE and the client does not call this entry point again and also does not call INX_logoff, then INX_terminate_query should be called to terminate this query and free up system resources.

Returns 0 with *last_match_p = TRUE and *entry_num_p = 0 if no records found.

ERRORS:

No significant errors.

*/

23.35. INX_get_index_desc

error_typ

```
INX_get_index_desc(ims_session, id, name_p, desc_p_p)
  INX_session_handle_typ  ims_session;      /* IN: session number */          (pg234.3)
  INX_index_id_typ        id;               /* IN: index id of index, or 0 to use
                                         index name to specify index */  (pg235.34)
  char*                   name_p;          /* IN: index name (not used if
                                         "id" != 0) */
```

+++ INX - Index Services +++

INX_index_desc_typ** desc_p_p; /* OUT: index description */ (pg239.34)

/* INX_get_index_desc finds the description of a particular index and returns a pointer to it. The client is expected to treat this area as read-only (do not modify or return).

ERRORS:

INX_err_no_index
INX_err_no_dict_param

*/

23.36. INX_get_key_desc

error_typ

INX_get_key_desc(ims_session, id, name_p, desc_p_p)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)

INX_key_id_typ id; /* IN: key id of key, or 0 to use key name to specify key. */ (pg242.16)

char* name_p; /* IN: name of key (not used if "id" != 0 */

INX_key_desc_typ** desc_p_p; /* OUT: description of key */ (pg242.30)

/* INX_get_key_desc finds the description of a particular key and returns a pointer to it. The client is expected to treat this area as read-only (do not modify or return).

ERRORS:

INX_err_no_key
INX_err_no_dict_param

*/

23.37. INX_get_menu_desc

error_typ

INX_get_menu_desc(ims_session, menu_name, menu_p_p)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)

char* menu_name; /* IN: menu name */

INX_menu_desc_typ** menu_p_p; /* OUT: menu description */ (pg243.13)

/* INX_get_menu_desc finds the description of a particular menu and returns a pointer to it.

The only menus "known" to index services are those named in index descriptions for menu items. Menus in forms are not returned by this entry point.

+++ INX - Index Services +++

The client is expected to treat this area as read-only (not modified or returned by client).

ERRORS:

 INX_err_no_menu

*/

23.38. INX_get_num_cluster_id

void

INX_get_num_cluster_id(fpnum_p, cid_p)

 FP_number fpnum_p; /* IN: FileNet number */ (pg211.20)

 INX_cluster_id_typ cid_p; /* OUT: cluster id created */ (pg243.32)

/* This procedure is for computing the clusterid for keys which are of type FP_number. Even though the cluster id is defined as an array of 3 shorts, it should be treated a string of 6 bytes. Therefore the cluster id created on either bigender or small ender platforms must be the same. */

23.39. INX_get_single_DIR

error_typ

INX_get_single_DIR(ims_session, doc_id, lock, override, cap_p, dir_p_p)

 INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)

 ASE_doc_id_typ doc_id; /* IN: document id of doc to get */ (pg483.37)

 FN_bool16_t lock; /* IN: if want to lock this document */ (pg491.12)

 FN_bool16_t override; /* IN: if want to override existing lock */ (pg491.12)

 INX_capability_typ* cap_p; /* OUT: capability if lock=TRUE */ (pg240.30)

 INX_doc_index_rec_typ** dir_p_p; /* OUT: DIR retrieved */ (pg238.17)

/* INX_get_single_DIR retrieves one DIR by doc_id.
The record can be locked for a planned update.

Memory for the INX_doc_index_rec_typ record is allocated by INX, and when its no longer needed the client must call INX_free_DIR to free this memory.

See also INX_get_single_match.

ERRORS

 INX_err_no_record

 INX_err_record_busy

*/

+++ INX - Index Services +++

23.40. INX_get_single_match

```
error_typ
INX_get_single_match(ims_session, doc_id, lock, override, cap_p, match_p_p)
  INX_session_handle_typ  ims_session; /* IN: session number */           (pg234.3)
  ASE_doc_id_typ          doc_id;      /* IN: document id */             (pg483.37)
  FN_bool16_t             lock;        /* IN: if want to lock DIR */     (pg491.12)
  FN_bool16_t             override;    /* IN: if want to override existing (pg491.12)
                                     lock */
  INX_capability_typ*     cap_p;       /* OUT: capability if lock=TRUE */  (pg240.30)
  INX_query_match_typ**   match_p_p;  /* OUT: DIR returned */           (pg242.13)
```

```
/* INX_get_single_match is similar to INX_get_single_DIR,
   but the record returned is embedded in a match structure.
   The record can be locked for a planned update.
```

Memory for the INX_doc_index_rec_typ record is allocated by INX, and when its no longer needed the client must call INX_free_matches to free this memory.

See also INX_get_single_DIR.

ERRORS

```
  INX_err_no_record
  INX_err_record_busy
*/
```

23.41. INX_get_str_cluster_id

```
void
INX_get_str_cluster_id(string_p, cid_p)
  char*          string_p; /* IN: string */
  INX_cluster_id_typ cid_p; /* OUT: cluster id created */           (pg243.32)
```

```
/* This procdure is for computing the clusterid for keys which are
   alphanumeric strings.
*/
```

23.42. INX_import_DIRS

```
error_typ
INX_import_DIRS(ims_session, buf_p, buf_len, num_dirs, check_level,
                fatal_p, status, exists, do_wfl)
  INX_session_handle_typ  ims_session; /* IN: session number */           (pg234.3)
  any_ptr                 buf_p;       /* IN: buffer of DIR records */   (pg491.24)
```

+++ INX - Index Services +++

```
long          buf_len;      /* IN:  buffer length */
long          num_dirs;    /* IN:  number of DIR records in buf */
short        check_level; /* IN:  INX_VAL_DEF, INX_VAL_DCL, or
                             INX_VAL_ALL */
FN_bool16_t* fatal_p;     /* OUT: If a fatal error occurred on      (pg491.12)
                             this call, and subsequent
                             import attempts are futile. */
error_typ     status[];    /* OUT: status of each document import   (pg493.26)
                             attempt (import attempt does not
                             stop on first error) */
short        exists[];    /* OUT: if document exists in doctaba */
FN_bool16_t  do_wfl;      /* IN:  TRUE=>document to be inserted   (pg491.12)
                             into WorkFlo queue, if queue
                             defined for class.
                             FALSE=>don't insert into queue,
                             even if queue defined. */
```

/* INX_import_DIRS inserts multiple rows into doctaba.

The error tuple returned by the function is the first error which occurred on a DIR, or "success" if no errors occurred.

The status[i] and exists[i] fields indicate the error and existence status of the "i" DIR in the buffer.

If the check_level value is INX_VAL_DEF, the record is inserted into the database to the extent possible. If a required index is not present or an index is of the wrong type, the index is set to null. If an index is present in a DIR which is not defined for the document class, then the index is dropped from the DIR record inserted into the database.

If the check_level value is INX_VAL_DCL, then any missing indexes, extra indexes, or mis-typed indexes will cause an error and the record will not be inserted into the database.

INX_VAL_ALL is the same as INX_VAL_DCL for the time being since there is only one check option, and therefore the "all" option is the same as the since check option.

ERRORS:

```
INX_err_string_too_long
INX_err_bad_value_type
INX_err_index_in_rec_twice
INX_err_reqd_is_null
INX_err_doc_id_range
INX_err_incomplete_dcl
INX_err_index_not_in_dcl
INX_err_no_index
```

+++ INX - Index Services +++

```
    INX_err_no_dcl
*/
```

23.43. INX_index_data_size

```
long
INX_index_data_size(type, data_p)
    INX_value_type_typ      type;
    char*                  data_p;
                                                                    (pg235.48)
```

```
/* INX_index_data_size returns the length of the "data" field of an index
value, padded with the correct number of bytes for the platform on
which it's run. The structures which have "data" fields are:
```

```
    INX_index_value_typ
    INX_index_choice_typ
    DOC_index_value_typ
```

The length is derived from "type" and, in the case of a string, "data". This routine is used to increment a pointer from one index to another when arrays of index values are used. For example, in the INX_doc_index_rec_typ structure, a loop starting at the first index and going to the last index would be as follows:

```
    INX_index_value_typ*      ip;
    INX_doc_index_rec_typ*    rec_p;

    ip = &rec_p->values[0];
    for (i = 0; i < rec_p->value_num; i++)
    {
        ip = (INX_index_value_typ*) ((unsigned long*)ip +
                                      INX_index_data_size (ip->type, ip->data) +
                                      ((unsigned long)(ip->data) - (unsigned long)ip));

        / * ip now points to the next index value * /
    }
```

Note that the macro "INX_index_size" (in file INX.defs) makes the assignment statement easier to type. The assignment could also be written as follows:

```
    ip = (INX_index_value_typ*)((unsigned long*)ip + INX_index_size (ip));
*/
```

23.44. INX_interrupt_query

+++ INX - Index Services +++

```
void
INX_interrupt_query (set_cancel)
    FN_bool16_t      set_cancel; /* IN:  TRUE => cancel query          (pg491.12)
                                FALSE => clear cancel condition */
```

/* INX_interrupt_query is used to interrupt a query which is in progress. It will interrupt any query in progress for the process which calls it, but will not affect any other process. The queries which this routine affects are:

 INX_find_DIRS
 INX_find_folders

Call "INX_interrupt_query (TRUE)" in a signal handler routine to cause the a query routine to abort and return quickly without finishing the query.

Call "INX_interrupt_query (FALSE)" to clear the cancel query request so subsequent calls to query routines do not abort. This call should be made after the interrupt has been recognized and processed.

*/

23.45. INX_logoff

```
error_typ
INX_logoff(ims_session)
    INX_session_handle_typ ims_session; /* IN:  session number */          (pg234.3)
```

/* INX_logoff terminates an Index Services session and invalidates a session on the server. No further calls should be with the ims_session handle.

*/

23.46. INX_logon

```
error_typ
INX_logon(service_name_p, name_type, leave_open, session_p, timeout_p)
    ASE_service_name_typ* service_name_p; /* IN:  service logging on to */      (pg486.28)
    ASE_nch_name_type_typ name_type;      /* IN:  service name is IMS or          (pg483.28)
                                           Index Service */
    FN_bool16_t          leave_open;      /* IN:  leave connection open */      (pg491.12)
    INX_session_handle_typ* session_p;    /* OUT: session number */            (pg234.3)
    unsigned long*       timeout_p;      /*      obsolete (input NULL) */
```

/* INX_logon establishes an Index Services session and returns a

+++ INX - Index Services +++

session number to the client which identifies the session.
The session number is passed to all other entry points. */

23.47. INX_move_folder

error_typ

INX_move_folder(ims_session, src, dest)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)
INX_folder_spec_typ src; /* IN: folder(s) to move */ (pg247.11)
INX_folder_name_typ dest; /* IN: folder to copy into */ (pg246.40)

/* INX_move_folder moves the "src" folder into the "dest" folder. The new folder name is created by taking the last component of "src" and appending it to the name given in "dest".

Example:

src is "/a/b/c*"
dest is "/e/f"
result is "/e/f/c*"

Moving a folder to its own descendent is not allowed.

ERRORS:

INX_err_bad_folder_name
INX_err_dup_record
INX_err_no_folder
INX_err_descendent_dest
INX_err_too_deep

*/

23.48. INX_open_connection

error_typ

INX_open_connection(ims_session)

INX_session_handle_typ ims_session; /* IN: session number */ (pg234.3)

/* INX_open_connection causes a network connection to the server to be established and maintained. It is used when a client wishes to perform several operations in succession and avoid the overhead of opening and closing a connection for each one. Otherwise, that is done within each operation. In the case of local operation of the service, this call has no user-visible effect.

INX_close_connection is used to close the connection when the series of operations is concluded.

*/

+++ INX - Index Services +++

23.49. INX_reorder_folder

error_typ

```
INX_reorder_folder(ims_session, folder_id, after_doc, num_docs, docs)
  INX_session_handle_typ ims_session; /* IN: session number */      (pg234.3)
  ASE_folder_id_typ      folder_id; /* IN: folder id */             (pg484.13)
  ASE_doc_id_typ         after_doc; /* IN: put docs after this document */ (pg483.37)
  short                  num_docs; /* IN: num docs in "docs" array */
  ASE_doc_id_typ         docs[]; /* IN: ordered array of doc ids */ (pg483.37)
```

/* This routine allows a subset of the documents in a folder to be reordered. The documents in the docs array are removed from their previous location and placed after the "after_doc" in the order given in the docs array.

ERRORS:

```
  INX_err_no_record
  INX_err_no_folder
  INX_err_not_in_folder
  INX_err_record_busy
```

*/

23.50. INX_terminate_query

error_typ

```
INX_terminate_query(ims_session)
  INX_session_handle_typ ims_session; /* IN: session number */      (pg234.3)
```

/* INX_terminate_query is called in place of a continuation call to INX_find_DIRS. It notifies Index Services that continuations will not be taking place and allows server resources associated with the query to be freed.

This function is called internally, automatically when a query reaches end-of-fetch, so it is unnecessary for clients to do so under those circumstances.

ERRORS

```
  INX_err_no_query
```

*/

23.51. INX_unfile_doc

error_typ

+++ INX - Index Services +++

```
INX_unfile_doc(ims_session, doc_id, folder_id)
  INX_session_handle_typ ims_session; /* IN: session number */      (pg234.3)
  ASE_doc_id_typ         doc_id;      /* IN: document id */         (pg483.37)
  ASE_folder_id_typ      folder_id;    /* IN: folder id */         (pg484.13)
```

/* INX_unfile_doc removes document "doc_id" from folder "folder_id".

ERRORS:

```
  INX_err_no_record
  INX_err_no_folder
  INX_err_not_in_folder
  INX_err_record_busy
```

*/

23.52. INX_unfile_docs

```
error_typ
INX_unfile_docs(ims_session, folder_id, num_docs, docs, untab_doc)
  INX_session_handle_typ ims_session; /* IN: session number */      (pg234.3)
  ASE_folder_id_typ      folder_id;    /* IN: folder id */         (pg484.13)
  short                  num_docs;     /* IN: num docs in docs array */
  ASE_doc_id_typ         docs[];       /* IN: array of document ids */  (pg483.37)
  FN_bool16_t           untab_doc;     /* IN: if want to untabify document */ (pg491.12)
```

/* This routine unfiles one or more docs from the folder specified by "folder_id". If untab_doc is TRUE, all doc_id's tabs will be deleted from the folder_tabs table. If untab_doc is FALSE, an error will be returned if the document is tabbed.

ERRORS:

```
  INX_err_no_record
  INX_err_no_folder
  INX_err_doc_is_tabbed
  INX_err_record_busy
```

*/

23.53. INX_update_DIR

```
error_typ
INX_update_DIR(ims_session, cap_p, dir_p, want_new_dir, new_dir_p_p)
  INX_session_handle_typ ims_session; /* IN: session number */      (pg234.3)
  INX_capability_typ*    cap_p;        /* IN: capability returned by prior (pg240.30)
                                         call to INX_get_single_DIR
                                         with lock=TRUE */
  INX_doc_index_rec_typ* dir_p;        /* IN: new DIR */              (pg238.17)
  FN_bool16_t           want_new_dir; /* IN: TRUE=>fetch and return updated (pg491.12)
```

+++ INX - Index Services +++

```

                                                    DIR. */
INX_doc_index_rec_typ** new_dir_p_p; /* IN: where to put new DIR if      (pg238.17)
                                                    want_new_dir=TRUE. Client
                                                    must free this memory with
                                                    INX_free_DIR. */
```

```
/* INX_update_DIR changes the index values of an existing document
index record, and releases the lock held on this record. The document
id and the document class in this record cannot be changed. The
document id must be present in the *dir_p even though it's also in
the *cap_p.
```

If only document id is passed in, INX_update_DIR will release the lock held on this record and return success.

ERRORS

```

    INX_err_no_record
    INX_err_no_capability
    INX_err_string_too_long
    INX_err_index_in_rec_twice
    INX_err_bad_value_type
    INX_err_reqd_is_null
    INX_err_cannot_change_dcl
    INX_err_incomplete_dcl
    INX_err_index_not_in_dcl
```

```
*/
```

23.54. INX_update_folder

error_typ

```

INX_update_folder(ims_session, cap_p, folder_p)
    INX_session_handle_typ ims_session; /* IN: session number */      (pg234.3)
    INX_capability_typ*   cap_p;        /* IN: capability returned from prior (pg240.30)
                                                    call to INX_get_and_lock_folder */
    INX_folder_desc_typ*  folder_p;     /* IN: new folder description */    (pg247.41)
```

```
/* Modifies a folder description. The folder must have been previously
locked by a call to INX_get_and_lock_folder.
```

ERRORS:

```

    INX_err_no_capability
    INX_err_invalid_security
    INX_err_dup_record
    INX_err_no_folder
```

```
*/
```

23.55. INX_where_filed

+++ INX - Index Services +++

error_typ

```
INX_where_filed(ims_session, doc_id, num_folders_p, folder_ids_p)
  INX_session_handle_typ  ims_session; /* IN: session number */           (pg234.3)
  ASE_doc_id_typ          doc_id;      /* IN: document id */             (pg483.37)
  unsigned*               num_folders_p; /* OUT: #folder ids in folder_ids_p */
  ASE_folder_id_typ**     folder_ids_p; /* OUT: array of folder ids. This   (pg484.13)
                                     array is allocated by INX and
                                     must be returned by
                                     calling INX_free_folder_ids */
```

```
/* INX_where_filed returns the list of folder IDs (in folder_ids_p)
   that contain the given document (doc_id).
```

ERRORS:

No significant errors

*/

24. PRI DECLARATIONS

Print Services Interface

#ifndef PRI_defs

#define PRI_defs

#include <FileNet.h>

#include <SEC.defs>

#include <AS_externals.h>

/* Request Statuses: */
/* These values are returned from the PRI_get_request_status and */
/* PRI_find_requests calls. The last two are internal only, and will */
/* be mapped into PRI_RS_QUEUED for clients. */
/* ***** */

typedef unsigned short PRI_request_status_typ;

#define PRI_RS_UNKNOWN 0 /* unknown (user doesn't have read
permission on request). */
#define PRI_RS_QUEUED 1 /* queued to printer, not processed yet */
#define PRI_RS_PRINTING 2 /* currently being printed. */
#define PRI_RS_COMPLETE 3 /* print is completed */
#define PRI_RS_FAILED 4 /* print has failed and is completed */
#define PRI_RS_CANCELLED 5 /* cancelled by operator */
#define PRI_RS_WAITING 6 /* request delayed to start a future time */
#define PRI_RS_FETCHING 7 /* fetch from optical disk in progress. */
#define PRI_RS_SUSPENDED 8 /* suspended by operator */

#define PRI_MAX_REQ_STATUS PRI_RS_SUSPENDED

/* Printer Operational Statuses: */
/* These values represent the operational state of a print server. */
/* They are returned from the PRI_get_service_status call. */
/* ***** */

+++ PRI - Print Services +++

```
typedef unsigned short PRI_printer_op_status_typ;

#define PRI_POS_UNKNOWN      0    /* Not known (network down?) */
#define PRI_POS_DISABLED    1    /* printer has been manually suspended */
#define PRI_POS_AVAILABLE   2    /* Available and ready */
#define PRI_POS_NEEDS_ATTN  3    /* Need paper, toner */
#define PRI_POS_NEEDS_SVC   4    /* Hardware malfunction */

#define PRI_MAX_OP_STATUS    PRI_POS_NEEDS_SVC

/*****
/* Paper Sizes:
/* These values define the various sizes that printers can handle.
/* A value selected must be listed in the record returned by
/* PRI_get_printer_attrs.  Additionally, if the size is also in the record
/* returned by PRI_get_service_status, then that means that the paper size
/* is already loaded on the printer and no operator has to switch paper.
/* Note that the three special sizes, PRI_PS_DONT_CARE, PRI_PS_BEST_AVAIL,
/* and PRI_PS_GLOBAL_DEFAULT are algorithms for selecting a paper size,
/* and not a specific size.  Also, the PRI_PS_DONT_CARE and
/* PRI_PS_GLOBAL_DEFAULT can be determined for every printer, so they are
/* legal for any printer.  PRI_PS_BEST_AVAIL, however, is only available
/* on printers which allow the software to determine the paper sizes loaded.*/
*****/

typedef unsigned short PRI_paper_size_typ;

#define PRI_PS_UNKNOWN      0    /* Invalid paper type */
#define PRI_PS_LETTER       1    /* 8.5" x 11" */
#define PRI_PS_LEGAL        2    /* 8.5" x 14" */
#define PRI_PS_B            3    /* 11" x 17" */
#define PRI_PS_C            4    /* 17" x 22" */
#define PRI_PS_D            5    /* 22" x 34" */
#define PRI_PS_E            6    /* 34" x 44" */
#define PRI_PS_A0           7    /* 841mm x 1189mm */
#define PRI_PS_A1           8    /* 594mm x 841mm */
#define PRI_PS_A2           9    /* 420mm x 594mm */
#define PRI_PS_A3          10    /* 297mm x 420mm */
#define PRI_PS_A4          11    /* 210mm x 297mm */
#define PRI_PS_A5          12    /* 148mm x 210mm */
#define PRI_PS_B4          13    /* 257.3mm x 365.25mm */
#define PRI_PS_B5          14    /* 257.3mm x 182.63mm */
#define PRI_PS_18x24       15    /* C+ size; 18" x 24" */
#define PRI_PS_TOP         16    /* Use paper in top tray */
#define PRI_PS_BOTTOM      17    /* Use paper in bottom tray */
#define PRI_PS_THIRD       18    /* Use paper in third tray */
#define PRI_PS_DONT_CARE   19    /* any paper size */
#define PRI_PS_HALF_LETTER 20    /* 5.5" x 8.5" */
```

+++ PRI - Print Services +++

```
#define PRI_PS_BEST_AVAIL      21    /* size which matches image size best */
#define PRI_PS_10x14          22    /* 10" x 14" */
#define PRI_PS_GLOBAL_DEFAULT 23    /* the predefined default for the printer */
#define PRI_PS_EXECUTIVE      24    /* 7.5" x 10.5" */
#define PRI_MAX_PAPER_SIZE    PRI_PS_EXECUTIVE
```

```
/*
 * The priority given to requests
 */
/*
 * For printers, requests are processed in order by priority, highest first,
 * and lowest last. All requests of the same priority are processed in
 * first in/first out order.
 */
/*
 * For FAX machines, request are processed mostly in order by priority,
 * except that busy redials can change the order. The first attempt at
 * sending a FAX request will be in priority order, but if the attempt fails
 * due to a busy error, then other requests will be processed before the
 * failed attempt will be retried.
 */
/*
 * Both FAX machines and printers use a pipelined architecture to speed up
 * operation, so that one document is being fetched at the same time another
 * is being decompressed and yet another is being printed/transmitted.
 * Once a document enters the pipeline other higher priority jobs will not
 * jump ahead of it, so priorities will not have an effect if only a small
 * number of pages are queued to a printer or fax.
 */

```

```
typedef unsigned short PRI_priority_typ;
```

```
#define PRI_HOLD_PRIORITY      0    /* Job suspended if priority is this */
#define PRI_MIN_PRIORITY      0
#define PRI_DEFAULT_PRIORITY  4
#define PRI_MAX_PRIORITY      9
```

```
/*
 * Scaling options:
 * These are the possible values that a "scaling" print_option can assume.
 */

```

```
typedef unsigned short PRI_scaling_typ;
```

```
#define PRI_SCALE_NORMAL      0    /* Scales image until the vertical
 * dimension of the image roughly matches
 * the vertical dimension of the paper.
 * If this causes the horizontal dimension
 * to grow too large, the right margin will
 * be clipped a configured amount. This
```


+++ PRI - Print Services +++

option does not preserve the aspect ratio. However, this is probably the most popular option since it is very and it preserves data on the bottom of the image, where clipping is most objectionable. */

```
#define PRI_SCALE_CLIPBOTH 1 /* Scales the image until it approximately
                             matches the paper size. If the scaled
                             image is larger than the paper, the
                             right and bottom margins will be
                             clipped a configured amount. The
                             aspect ratio is preserved. */
#define PRI_SCALE_EXACT 2 /* The image is scaled to fit the paper as
                             closely as possible without losing any
                             data. Despite the use of "exact" in the
                             name, this option does not guarantee to
                             fit even one edge precisely. This
                             option is the slowest scaling
                             operation. */
#define PRI_SCALE_APPROX 3 /* The image is scaled so that it occupies
                             as much as the paper size as possible
                             without clipping any data. If the
                             image exceeds the paper size by any
                             amount, the image will be scaled down by
                             an entire 2x. Successive 2x reductions
                             are performed until the image fits. The
                             aspect ration is preserved. */
#define PRI_SCALE_ORIGINAL 4 /* This option causes no change in size to
                                be made. Note, however, than whenever
                                "scale original" is specified, any
                                paper size option requested is ignored,
                                and the paper size chosen is the
                                smallest paper upon which the image can
                                be placed without clipping the image.
                                Since most scanners collect data edge
                                to edge but most printers can't print
                                edge to edge, this option usually causes
                                the request to be printed on a paper
                                size which is larger than the paper
                                originally scanned. */
#define PRI_SCALE_CENTER 5 /* Puts the image in the center of the
                                paper. This may cause clipping on both
                                edges, or addition of white space on
                                both edges. */
#define PRI_SCALE_ENHANCED_EXACT 6 /* This option is an improvement over
                                       PRI_SCALE_EXACT. This option keeps the
                                       size within 5% of the original. This
                                       option is also quite slow. */
```

+++ PRI - Print Services +++

```
#define PRI_MAX_SCALE          PRI_SCALE_ENHANCED_EXACT

/*****
/* Orientation options:                                     */
/* These are the possible values that a print_option of "orientation"
/* can assume.                                             */
*****/

typedef unsigned short PRI_orientation_typ;

#define PRI_ORIENT_DEFAULT    0 /* Aligns the long edge of the image with the
long edge of the paper. */
#define PRI_ORIENT_LANDSCAPE  1 /* Causes the image to be rotated (if
necessary) so that the long edge of the
image runs horizontally on the paper,
regardless of the orientation of the paper.
The output paper must be at least one size
larger than the stored image. */
#define PRI_ORIENT_PORTRAIT   2 /* Causes the image to be rotated (if
necessary) so that the long edge of the
image runs vertically on the paper,
regardless of scanning orientation. */
#define PRI_ORIENT_NO_ROTATE  3 /* Causes the image to be printed in whatever
orientation it is stored, regardless of the
orientation of the paper. */

#define PRI_MAX_ORIENT        PRI_ORIENT_NO_ROTATE

/*****
/* Print_direction_options                                 */
/* These values represent the two print directions available.
*****/

typedef unsigned short PRI_print_direction_typ;

#define PRI_PD_FORWARD        0
#define PRI_PD_BACKWARD      1

#define PRI_MAX_PD            PRI_PD_BACKWARD

/*****
/* fax_mode_options                                       */
/* These values represent the valid fax mode(resolution) options.
*****/

typedef unsigned short PRI_fax_mode_typ;
```

+++ PRI - Print Services +++

```
#define PRI_FM_COARSE          0 /* Normal resolution (204 dpi x 98 dpi)
                               Coarse by default */
#define PRI_FM_FINE           1 /* Fine resolution (204 dpi x 196 dpi) */
#define PRI_FM_300_DPI        2 /* Detail resolution (300 X 300 dpi ) */
#define PRI_FM_400_DPI        3 /* Super resolution (400 x 400 dpi) */
#define PRI_FM_FNDATA         4 /* FileNet image data type. */
#define PRI_FM_OTHERDATA      5 /* Other data type */

#define PRI_MAX_FM            PRI_FM_OTHERDATA
#define PRI_DEFAULT_FM        PRI_FM_COARSE

/*****
/* Printer_type values.                                     */
/* These values represent the various types of printers available. */
*****/

typedef unsigned short PRI_printer_type_typ;

#define PRI_PT_UNKNOWN        0 /* Device type is unknown */
#define PRI_PT_PRINTER        1 /* Device is a printer */
#define PRI_PT_FAX            2 /* Device is a facsimile machine */

#define PRI_MAX_PT            PRI_PT_FAX

/*****
/* Overlay types.                                         */
/* Please note that only pure-text overlay is currently supported */
*****/

typedef unsigned short PRI_overlay_typ;

#define PRI_NO_OVERLAY        0 /* no overlays */
#define PRI_ONE_OVERLAY       1 /* 1st page is overlaid */
#define PRI_MULTI_OVERLAY     2 /* all pages overlaid */

#define PRI_MAX_OVERLAY       PRI_MULTI_OVERLAY

/*****
/* Eject tray type                                       */
/* This value indicates which output tray the printed pages should be */
/* ejected into. An eject tray number may be specified (1..n), or */
/* PRI_DEFAULT_EJECT_TRAY may be specified for the default tray */
*****/

typedef unsigned short PRI_eject_tray_typ;

#define PRI_DEFAULT_EJECT_TRAY 0
#define PRI_ET_DEFAULT         0 /* Default Eject Tray */
```

+++ PRI - Print Services +++

```
#define PRI_ET_UPPER          1      /* Upper */
#define PRI_ET_UPPER_OFFSET  2      /* Upper Offset */
#define PRI_ET_LOWER         3      /* Lower */
#define PRI_ET_LOWER_OFFSET  4      /* Lower Offset */
#define PRI_ET_SIDE          5      /* Side */
#define PRI_ET_BIN1          6      /* Bin 1 */
#define PRI_ET_BIN2          7      /* Bin 2 */
#define PRI_ET_BIN3          8      /* Bin 3 */
#define PRI_ET_BIN4          9      /* Bin 4 */
#define PRI_ET_BIN5         10     /* Bin 5 */
#define PRI_ET_BIN6         11     /* Bin 6 */
#define PRI_ET_BIN7         12     /* Bin 7 */
#define PRI_ET_BIN8         13     /* Bin 8 */
#define PRI_ET_BIN9         14     /* Bin 9 */
#define PRI_ET_BIN10        15     /* Bin 10 */
#define PRI_MAX_EJECT_TRAY  PRI_ET_BIN10

/*****
/* Service name ids
/* There is a one to one mapping between a service name and an id
*****/

typedef unsigned short      PRI_service_id_typ;

#define PRI_NO_SERVICE_ID   0xffff

/*****
/* Maximum sizes
*****/

#define PRI_MAX_FORM_NAME_LEN  32
#define PRI_MAX_NOTE_LEN      42
#define PRI_MAX_PHONE_NUM_LEN  40
#define PRI_MAX_HEADLINE_LEN  98
#define PRI_MAX_DOCS_PER_REQ  1000
#define PRI_MAX_PHONE_EXT_LEN  48
#define PRI_MAX_MEMO_LEN      376

/*****
/* Print Options:
/* These defines enumerate the possible types of print options in the
/* PRI_print_option_typ record.  The option_type field in that structure
/* determines which of the union's members is appropriate.  Note that not
/* every printer supports all options listed, so check if your printer
/* supports an option before trying to use it.  Also note that FAX machines
/* support options which printers don't, and vice versa, and which
/* type of device is applicable to an option is indicated by the presence
/* PRINTER, FAX, or PRINTER & FAX on the lines listed below.
*****/
```

+++ PRI - Print Services +++

/*****

typedef unsigned short PRI_option_typ;

```
#define PRI_OPTION_PAPER_SIZE      0 /* PRINTER & FAX** */
#define PRI_OPTION_PRIORITY        1 /* PRINTER & FAX */
#define PRI_OPTION_PRINTER_NAME    2 /* PRINTER & FAX (printer or fax name) */
#define PRI_OPTION_COLLATE        3 /* PRINTER only */
#define PRI_OPTION_SECURITY        4 /* PRINTER & FAX */
#define PRI_OPTION_COPIES         5 /* PRINTER only */
#define PRI_OPTION_PRINT_TIME      6 /* PRINTER & FAX */
#define PRI_OPTION_STAPLE         7 /* PRINTER only */
#define PRI_OPTION_TWO_SIDED      8 /* PRINTER only */
#define PRI_OPTION_FORM_NAME      9 /* PRINTER only */
#define PRI_OPTION_NOTE          10 /* PRINTER only */
#define PRI_OPTION_ANNOTATIONS    11 /* PRINTER & FAX */
#define PRI_OPTION_REQUEST_HEADER 12 /* PRINTER & FAX */
#define PRI_OPTION_DOC_HEADERS    13 /* PRINTER only */
#define PRI_OPTION_SCALING        14 /* PRINTER only */
#define PRI_OPTION_ORIENTATION    15 /* PRINTER only */
#define PRI_OPTION_OVERLAY       16 /* PRINTER only */
#define PRI_OPTION_PHONE_NUM     17 /* FAX only */
#define PRI_OPTION_HEADLINE      18 /* FAX only */
#define PRI_OPTION_FAX_MODE      19 /* FAX only */
#define PRI_OPTION_PAGE_FOOTNOTE  20 /* FAX only */
#define PRI_OPTION_TIME_FOOTNOTE  21 /* FAX only */
#define PRI_OPTION_EJECT_TRAY    22 /* PRINTER only */
#define PRI_OPTION_PHONE_EXT     23 /* FAX only */
#define PRI_OPTION_MEMO          24 /* FAX only */
#define PRI_OPTION_COVER_DOC     25 /* FAX only */
#define PRI_OPTION_COVER_SSN     26 /* FAX only */
```

```
#define PRI_MAX_OPTION             PRI_OPTION_COVER_SSN
```

```
/* **Note that some fax machine types will ignore the paper size option */
```

```
*****
/* PRI_print_option_typ                                     */
/* This typedef contains a union of all the print options defined */
/* in the protocol spec. The option_type field determines which */
/* member of the union it is. Union members that are more than 12 bytes */
/* are pointers, to keep the size of the structure reasonable. */
*****
```

```
typedef struct PRI_print_option_struct {
```

```
    PRI_option_typ    option_type; /* see "Print Option" defines */ (pg303.3)
    union {
        PRI_paper_size_typ    paper_size; /* Paper to print request on. (pg297.26)
```

+++ PRI - Print Services +++

```

PRI_priority_typ      priority;          /* Priority of request.          (pg298.29)
                        Default=PRI_PS_GLOBAL_DEFAULT */
ASE_service_name_typ* printer_name_p; /* Printer to print request on. (pg486.28)
                        If not specified, defaults to
                        first printer found which has
                        the attributes required for the
                        request submitted. */
bool                  collate;          /* If want pages collated.      (pg491.22)
                        Default=FALSE. */
SEC_access_restrictions security;      /* Security for update/cancel   (pg385.2)
                        of this print request. Default
                        is read by anyone, and modify
                        or delete only by user who
                        submitted request (or SysAdmin)*/
unsigned short        copies;          /* #of copies to print.        (pg489.31)
                        Default = 1. */
ASE_time_typ          print_time;      /* Start print time. Used to   (pg489.31)
                        delay print requests.
                        Default is to start as soon as
                        a printer is available. */
bool                  staple;          /* If want output stapled.      (pg491.22)
                        Default is FALSE. */
bool                  two_sided;       /* If want two sided printing.  (pg491.22)
                        Default is FALSE. */
char*                 form_name_p;     /* maxlen=PRI_MAX_FORM_NAME_LEN.
                        Name of form to overlay pages
                        of request. Supported on FDOS
                        print servers only. Note: This
                        same feature can be achived with
                        the overlay option (see below).*/
char*                 note_p;         /* maxlen=PRI_MAX_NOTE_LEN.
                        A message which is printed on
                        request header page. */
bool                  annotations;     /* If want annotations printed. (pg491.22)
                        Note that this flag is ignored
                        if the PRI_doc_specifier_typ
                        record has "svc_type" equal to
                        "PRI_ST_CACHE_SERVICE".
                        Default is FALSE. */
bool                  request_header;  /* If want a header on this request.(pg491.22)
                        Default is FALSE. */
bool                  doc_headers;     /* If want header on each doc in (pg491.22)
                        this request. Default is FALSE*/
PRI_scaling_typ       scaling;         /* Scaling option to use. Default (pg298.41)
                        is PRI_SCALE_NORMAL. */
PRI_orientation_typ  orientation;     /* Orientation to use.          (pg300.10)
                        Default=PRI_ORIENT_DEFAULT. */

```

+++ PRI - Print Services +++

```

PRI_overlay_typ      overlay;      /* If first page of request is an      (pg301.30)
                                overlay page to be placed on top
                                of next or all subsequent pages.
                                Default=PRI_NO_OVERLAY. */
char*                phone_num_p; /* phone number to send request to,
                                maxlen=PRI_MAX_PHONE_NUM_LEN
                                Note: FAX only option */
char*                phone_ext_p; /* additional phone numbers for
                                routing purposes.
                                maxlen=PRI_MAX_PHONE_EXT_LEN
                                Note: FAX only option */
char*                headline_p; /* specifies the To...From...
                                message at top of each page.
                                maxlen=PRI_MAX_HEADLINE_LEN
                                Note: FAX only option */
char*                memo_p;      /* specifies additional information
                                related to the fax message.
                                maxlen=PRI_MAX_MEMO_LEN
                                Note: FAX only option */
PRI_fax_mode_typ    fax_mode;     /* resolution.                                (pg300.47)
                                Default=PRI_FM_COARSE.
                                Note: FAX only option */
bool                page_footnote; /* prints the page number at the      (pg491.22)
                                bottom of each page (NOTE: not
                                implemented!).
                                Note: FAX only option */
bool                time_footnote; /* prints the transmission date      (pg491.22)
                                and time at the bottom of each
                                page (NOTE: not implemented!).
                                Note: FAX only option */
PRI_eject_tray_typ  eject_tray;   /* Output paper tray to put          (pg301.45)
                                printed result in.
                                Note: non-FAX option */
ASE_doc_id_typ      cover_doc;    /* Cover page document used          (pg483.37)
                                by the FAX. It will have
                                special characters to handle
                                cover page info. The document
                                must available on the local
                                Print Server. FAX only option
                                */
ASE_ssn_typ         cover_ssn;    /* Cover page ssn. FAX only */      (pg483.47)
} opt;
} PRI_print_option_typ;

/*****
/* PRI_printer_status_typ
/* This typedef is used to represent the status of a printer. It has a
/* *next pointer to build a linked list of PRI_printer_statuses, one
*/

```

+++ PRI - Print Services +++

```

/* for each physical printer controlled by the service. */
/*****

typedef struct PRI_printer_status {
    unsigned short      magic;          /* magic number          */
    ASE_service_name_typ name;          /* print server's name  */ (pg486.28)
    PRI_printer_op_status_typ oper_status; /* status values        */ (pg297.1)
    PRI_print_direction_typ print_direction; /* direction to print in */ (pg300.35)
    unsigned short      requests_queued; /* requests_queued is the
                                        /* requests queued for this*/
                                        /* print server only. For */
                                        /* the entire service, see */
                                        /* PRI_service_status_typ */

    unsigned short      pages_queued;   /* pages_queued is the
                                        /* pages assigned to this */
                                        /* print server only      */

    unsigned short      num_paper_sizes; /* #paper_sizes         */
    PRI_paper_size_typ  *paper_sizes;   /* paper_sizes points to an*/ (pg297.26)
                                        /* array of paper sizes   */
                                        /* currently loaded on    */
                                        /* printer.               */

} PRI_printer_status_typ;

/*****
/* PRI_doc_specifier_typ */
/* This typedef is used to specify the doc_id, first_page, and last_page */
/* of a document to print, and the type (doc or cache) and name of the */
/* service from which the document can be gotten. */
/*****

typedef unsigned short PRI_service_type_typ;

#define PRI_ST_DOC_SERVICE      1      /* svc_name is a document service */
#define PRI_ST_CACHE_SERVICE   2      /* svc_name is a cache service */

#define PRI_MAX_SERVICE_TYPE   PRI_ST_CACHE_SERVICE

typedef struct PRI_doc_specifier_struct {
    ASE_page_range_typ  doc;           /* Doc_id, first_page and last_page. */ (pg484.42)
                                        /* Last_page > num pages in doc will */
                                        /* print to end of document.         */
                                        /* Last_page = 0 will print to end   */
                                        /* of document.                      */
                                        /* First_page = 0 or 1 will start    */
                                        /* print with first page of document */
                                        /* First_page > num pages in doc will*/
                                        /* generate an error.               */

    PRI_service_type_typ svc_type;     /* see "Service Type" defines above */ (pg306.31)
}

```


+++ PRI - Print Services +++

```

ASE_service_name_typ  svc_name;      /* Name of a cache or doc service. */ (pg486.28)
/* If svc_type is doc service, */
/* svc_name may be all NULL strings */
/* to use the doc service local to */
/* the print service. */
ASE_ssn_typ           ssn;           /* If this is a cache service, then */ (pg483.47)
/* ssn is the ssn of the object(s) */
/* in the cache, or ASE_LOCAL_SSN */
bool                  delete_after; /* If true, delete this cache object */ (pg491.22)
/* after printing. Note that this */
/* flag must be TRUE if svc_type is */
/* a cache and svc_name is a print */
/* cache name. */
} PRI_doc_specifier_typ;

/*****
/* PRI_service_status_typ */
/* This typedef is the basis for the service status. Its major */
/* component is the head of a linked list of PRI_printer_statuses. */
*****/

typedef struct PRI_service_status_struct {
    unsigned short    magic;          /* magic number */
    unsigned short    requests_queued; /* requests_queued is the total */
/* number of print requests */
/* pending for this service */
    unsigned short    num_printers;   /* the length of the */
/* printer_statuses array */
    PRI_printer_status_typ *printer_statuses; /* printer_statuses points to an */ (pg306.22)
/* array of printer statuses, */
/* one for each print server */
/* under control of this service*/
} PRI_service_status_typ;

/*****
/* PRI_printer_attr_typ */
/* This typedef describes the capabilities of a given printer. */
*****/

typedef struct PRI_printer_attr {
    unsigned short    magic;          /* magic number */
    ASE_service_name_typ    name;      /* name of printer server */ (pg486.28)
    PRI_printer_type_typ    printer_type; /* see printer type defines. */ (pg301.17)
    ASE_service_name_typ    security;   /* security needed to print on */ (pg486.28)
/* this printer. */
    unsigned short    pages_per_min;   /* speed of this printer */
    bool              two_sided;      /* true iff printer can duplex */ (pg491.22)
    bool              staple;         /* true iff printer can staple */ (pg491.22)
}

```

+++ PRI - Print Services +++

```

unsigned short      trays;          /* 1, 2, or 3          */
unsigned short      num_paper_sizes; /* length of paper_sizes array */
PRI_paper_size_typ  *paper_sizes;   /* points to array of sizes */ (pg297.26)
/* configured. Note that not */
/* all of these may be currently*/
/* loaded (see */
/* PRI_service_status_typ). */

} PRI_printer_attr_typ;

/*****
/* PRI_printer_attr2_typ */
/* This typedef describes the capabilities of a given printer. */
/* The PRI_printer_attr2_typ information can be retrieved by a */
/* PRI_get_printer_attrs2. */
*****/

typedef struct PRI_printer_attr2_typ {
    unsigned short      magic;
    struct PRI_printer_attr2_typ  *next_p; /* Points to the next element */ (pg308.43)
/* in the linked list. */
    ASE_service_name_typ  name; /* name of printer server */ (pg486.28)
    PRI_printer_type_typ  printer_type; /* see printer type defines. */ (pg301.17)
    ASE_service_name_typ  security; /* security needed to print on */ (pg486.28)
/* this printer. */
    unsigned short      pages_per_min; /* speed of this printer */
    bool                two_sided; /* true iff printer can duplex */ (pg491.22)
    bool                staple; /* true iff printer can staple */ (pg491.22)
    unsigned short      trays; /* 1, 2, or 3 */
    unsigned short      num_paper_sizes; /* length of paper_sizes array */
    PRI_paper_size_typ  *paper_sizes; /* points to array of sizes */ (pg297.26)
/* configured. Note that not */
/* all of these may be currently*/
/* loaded (see */
/* PRI_service_status_typ). */
    unsigned short      num_eject_trays; /* length of eject_trays array */
    PRI_eject_tray_typ  *eject_trays; /* points to array of eject */ (pg301.45)
/* tray configured */
    unsigned short      num_fax_modes; /* length of fax_modes array */
    PRI_fax_mode_typ    *fax_modes; /* points to array of fax */ (pg300.47)
/* mode configured */
} PRI_printer_attr2_typ;

/*****
/* PRI_request_desc_typ */
/* This typedef describes the publicly visible attributes of a print */
*****/

```

+++ PRI - Print Services +++

```

/* request. */
/*****/

typedef struct PRI_request_desc_struct {
    ASE_request_id_typ    request_id;    /* request id */           (pg486.15)
    ASE_time_typ          submit_time;    /* submit time */         (pg489.31)
    PRI_request_status_typ request_status; /* status */              (pg296.28)
    bool                  fax_request;    /* TRUE if fax request */ (pg491.22)
    unsigned short        unused;         /* not used */
    unsigned long          total_pages;    /* total #pages in request */
    ASE_time_typ          done_time;      /* time request finished */ (pg489.31)
    ASE_time_typ          print_time;     (pg489.31)
    PRI_priority_typ      priority;       (pg298.29)
    PRI_paper_size_typ    paper_size;     (pg297.26)
    unsigned short        copies;
    bool                  collate;        (pg491.22)
    bool                  staple;         (pg491.22)
    bool                  two_sided;      (pg491.22)
    bool                  annotate;        (pg491.22)
    bool                  req_header;     (pg491.22)
    bool                  doc_headers;    (pg491.22)
    PRI_scaling_typ       scaling;        (pg298.41)
    PRI_orientation_typ   orientation;    (pg300.10)
    PRI_fax_mode_typ      fax_mode;       /* FAX only */           (pg300.47)
    bool                  page_footnote;  /* FAX only */           (pg491.22)
    bool                  time_footnote;  /* FAX only */           (pg491.22)
    ASE_service_name_typ  printer;        (pg486.28)
    ASE_service_name_typ  user;           /* user who submitted request */ (pg486.28)
    char                  form_name [PRI_MAX_FORM_NAME_LEN+1];
    char                  note          [PRI_MAX_NOTE_LEN+1];
    char                  phone_num [PRI_MAX_PHONE_NUM_LEN+1]; /* FAX only */
    char                  headline [PRI_MAX_HEADLINE_LEN+1]; /* FAX only */
} PRI_request_desc_typ;

```

```

/*****/
/* PRI_sub_status_typ */
/* This typedef describes the publicly visible sub status of print request. */
/*****/
typedef struct PRI_sub_status_typ
{
    unsigned long    sub_status_prg;
    unsigned long    pct_complete;
} PRI_sub_status_typ;

```

```

#define PRI_SUBSTATUS_UNINITIALIZED 0
#define PRI_SUBSTATUS_WAITING_CONV 1
#define PRI_SUBSTATUS_CONVERSION 2

```

+++ PRI - Print Services +++

```

#define PRI_SUBSTATUS_SPOOLING      3
#define PRI_SUBSTATUS_TRANSMIT      4
#define PRI_SUBSTATUS_RETRY         5
#define PRI_SUBSTATUS_PRINTING      6
#define PRI_MAX_SUBSTATUS           PRI_SUBSTATUS_PRINTING

/*****
/* PRI_request_desc2_typ:
/* This typedef describes the new publicly visible attributes of a print
/* request.  These descriptions can be retrieved through PRI_find_requests2.*/
*****/

typedef struct PRI_request_desc2_typ {
    unsigned short      magic;
    struct PRI_request_desc2_typ*  next_p;
    ASE_request_id_typ  request_id;    /* request id */
    ASE_time_typ        submit_time;   /* submit time */
    PRI_request_status_typ  request_status; /* status */
    bool                fax_request;   /* TRUE if fax request */
    unsigned long       total_pages;   /* total #pages in request */
    ASE_time_typ        done_time;     /* time request finished */
    ASE_time_typ        print_time;    /*
    PRI_priority_typ    priority;
    PRI_paper_size_typ  paper_size;
    unsigned short      copies;
    bool                collate;
    bool                staple;
    bool                two_sided;
    bool                annotate;
    bool                req_header;
    bool                doc_headers;
    PRI_scaling_typ     scaling;
    PRI_orientation_typ orientation;
    PRI_fax_mode_typ    fax_mode;      /* FAX only */
    bool                page_footnote; /* FAX only */
    bool                time_footnote; /* FAX only */
    ASE_service_name_typ printer;
    ASE_service_name_typ user;         /* user who submitted request */
    char                form_name [PRI_MAX_FORM_NAME_LEN+1];
    char                note       [PRI_MAX_NOTE_LEN+1];
    char                phone_num [PRI_MAX_PHONE_NUM_LEN+1]; /* FAX only */
    char                headline  [PRI_MAX_HEADLINE_LEN+1]; /* FAX only */
    char                phone_ext [PRI_MAX_PHONE_EXT_LEN+1]; /* FAX only */
    char                memo      [PRI_MAX_MEMO_LEN+1];      /* FAX only */
    PRI_eject_tray_typ  eject_tray;   /* Output paper tray to put
    PRI_overlay_typ     overlay;      /* If first page of request is an
                                     printed result in.  Non-FAX*/
                                     overlay page to be placed on top

```

+++ PRI - Print Services +++

```

of next or all subsequent pages.
Default=PRI_NO_OVERLAY. */
SEC_access_restrictions security; /* Security for update/cancel (pg385.2)
of this print request. Default
is read by anyone, and modify
or delete only by user who
submitted request (or SysAdmin)*/
PRI_sub_status_typ sub_status; /* Specifies the progress of (pg309.44)
the request when the request
status is printing. */
ASE_doc_id_typ cover_doc; /* FAX cover document */ (pg483.37)
ASE_ssn_typ cover_ssn; /* FAX cover document's ssn */ (pg483.47)
unsigned short num_docs; /* Number of Associated documents
with the requests. This is
optionally returned. num_docs
count of the elements in
doc_array. */
PRI_doc_specifier_typ *doc_array; /* An array of documents associated (pg307.14)
with the requests. The length
of doc_array is returned in the
num_docs field above. */
} PRI_request_desc2_typ;

/*****
/* PRI_position_typ: */
/* This structure is used by PRI_find_requests2 to track the cursor */
/* position of returned requests list. */
*****/

typedef unsigned long PRI_position_typ[4];

/*****
/* PRI_type_typ: */
/* These values are used to specify future field types. This would allow */
/* for additional data transmission for some RPCs without breaking previous*/
/* implementations. This is used in PRI_get_printer_attrs2 and */
/* PRI_find_requests2. */
*****/

typedef unsigned short PRI_type_typ;
#define PRI_TYPE_UNKNOWN 0

/*****
/* PRI_class_typ: */
/* These values are used to specify future field classes. This would allow*/
*****/
```

+++ PRI - Print Services +++

```
/* for additional data transmission for some RPCs without breaking previous*/
/* implementations. This is used in PRI_get_printer_attrs2 and */
/* PRI_find_requests2. */
/*****/
```

```
typedef unsigned short PRI_class_typ;
#define PRI_CLASS_UNKNOWN 0
#define PRI_CLASS_SHORT 1
#define PRI_CLASS_LONG 2
#define PRI_CLASS_STRING 3
#define PRI_CLASS_ARRAY 4
```

```
/* *****
/* PRI_request_filter_typ */
/* *****
```

```
typedef struct {
    ASE_service_name_typ    user;                (pg486.28)
    PRI_printer_type_typ    queue_type;          (pg301.17)
    PRI_priority_typ        priority;            (pg298.29)
    PRI_request_status_typ  request_status;      (pg296.28)
    ASE_service_name_typ    printer;            (pg486.28)
    ASE_request_id_typ      request_id;          (pg486.15)
} PRI_request_filter_typ;
```

```
/* page record */
```

```
typedef unsigned short PRI_doc_spec_id_typ;
typedef struct {
    ASE_request_id_typ      request_id;          /* print request id */ (pg486.15)
    PRI_doc_spec_id_typ    doc_spec_id;        /* index into array of
                                                PRI_doc_specifier_typ */ (pg312.30)
    ASE_page_number_typ    page_num;          /* page #*/ (pg483.41)
} PRI_page_rec_typ;
```

```
/* ***** ERROR DEFINITIONS *****
```

```
/* Error function defines: */
```

```
#define PRI_XIN_PROTOCOL 0 /* Defined in this Courier protocol */
#define PRI_XNOT_IN_PROTOCOL 1 /* Errors not defined there */

#define PRI_zother_error 5
```

```
/* Error number defines: */
```

+++ PRI - Print Services +++

```
/*$D err_PRI */

#define PRI_pro_err(x)  err_encode (err_PRI, PRI_XIN_PROTOCOL, (x))
#define PRI_dev_err(x)  err_encode (err_PRI, PRI_XNOT_IN_PROTOCOL, (x))

/* Errors defined in the protocol */

#define PRI_err_invalid_session_handle      PRI_pro_err (1)
/*$M PRI given invalid session handle; session may have timed out.
Look at log for further information. */

#define PRI_err_no_permission               PRI_pro_err (2)
/*$M Client does not have valid access for the document or request. */

#define PRI_err_invalid_option              PRI_pro_err (3)
/*$M The Print Service cannot satisfy some of the specified options. */

#define PRI_err_no_such_request             PRI_pro_err (4)
/*$M The specified request ID could not be found. */

#define PRI_err_other_error                 PRI_pro_err (PRI_zother_error)
/*$M PRI 'other' error -- not in protocol */

#define PRI_err_no_such_service             PRI_pro_err (6)
/*$M The given service name is not in Clearinghouse. */

#define PRI_err_invalid_page_range          PRI_pro_err (7)
/*$M Some or all of the pages specified are not in the document. */

#define PRI_err_invalid_parameter           PRI_pro_err (8)
/*$M One or more parameters of the RPC were invalid. */

#define PRI_err_invalid_staple              PRI_pro_err (9)
/*$M Selected printer can't staple. */

#define PRI_err_invalid_two_sided           PRI_pro_err (10)
/*$M Selected printer can't print two-sided. */

#define PRI_err_invalid_priority            PRI_pro_err (11)
/*$M Selected priority is out-of-range. */

#define PRI_err_invalid_fax_collate         PRI_pro_err (12)
/*$M Invalid option for fax; can't collate. */

#define PRI_err_invalid_fax_form            PRI_pro_err (13)
/*$M Invalid option for fax; won't accept a form name. */

#define PRI_err_invalid_pri_form            PRI_pro_err (14)
```

+++ PRI - Print Services +++

```
/*$M Selected form-name size is invalid. */

#define PRI_err_invalid_fax_copies          PRI_pro_err (15)
/*$M Invalid option for fax; won't accept copies. */

#define PRI_err_invalid_pri_copies          PRI_pro_err (16)
/*$M Selected number of copies is out-of-range. */

#define PRI_err_invalid_note                PRI_pro_err (17)
/*$M Selected note option size is out-of-range. */

#define PRI_err_invalid_fax_scaling         PRI_pro_err (20)
/*$M Invalid option for fax; won't accept scaling */

#define PRI_err_invalid_pri_scaling         PRI_pro_err (21)
/*$M Selected scaling value is too large. */

#define PRI_err_invalid_fax_orientation     PRI_pro_err (22)
/*$M Invalid option for fax; won't accept orientation. */

#define PRI_err_invalid_pri_orientation     PRI_pro_err (23)
/*$M Selected orientation value is too large. */

#define PRI_err_invalid_fax_overlay        PRI_pro_err (24)
/*$M Invalid option for fax; won't accept overlay. */

#define PRI_err_invalid_pri_overlay        PRI_pro_err (25)
/*$M Invalid overlay option. */

#define PRI_err_invalid_pri_phone          PRI_pro_err (26)
/*$M Phone number not valid for non-fax printer */

#define PRI_err_invalid_fax_phone          PRI_pro_err (27)
/*$M Phone number string has an invalid length */

#define PRI_err_invalid_pri_headline        PRI_pro_err (28)
/*$M Invalid option for printer; won't accept headline. */

#define PRI_err_invalid_fax_headline        PRI_pro_err (29)
/*$M Fax headline size of out-of-range. */

#define PRI_err_invalid_pri_mode           PRI_pro_err (30)
/*$M Invalid option for printer; won't accept fax-mode. */

#define PRI_err_invalid_fax_mode           PRI_pro_err (31)
/*$M Selected fax mode option is invalid */

#define PRI_err_invalid_pri_footnote        PRI_pro_err (32)
```


+++ PRI - Print Services +++

```
/*$M Invalid option for printer; won't accept footnotes. */

#define PRI_err_invalid_status          PRI_pro_err (33)
/*$M Print request is in invalid state for operation requested. */

#define PRI_err_invalid_fax_eject_tray  PRI_pro_err (34)
/*$M Invalid option for fax; won't accept eject tray. */

#define PRI_err_print_data_too_short    PRI_pro_err (35)
/*$M Can't print less than 5 bytes with PRI_print_data */

#define PRI_err_invalid_fax_staple      PRI_pro_err (36)
/*$M Invalid option for fax; won't accept staple option */

#define PRI_err_invalid_fax_two_sided   PRI_pro_err (37)
/*$M Invalid option for fax; won't accept two sided option */

#define PRI_err_invalid_fax_doc_headers PRI_pro_err (38)
/*$M Invalid option for fax; won't accept doc headers option */

#define PRI_err_invalid_fax_note        PRI_pro_err (39)
/*$M Invalid option for fax; won't accept note option */

#define PRI_err_invalid_page_ordering   PRI_pro_err (40)
/*$M Invalid first and last page; The last page is less
the first page. A print request was submitted with
last_page < first_page of the document. This is unacceptable. */

#define PRI_err_invalid_pri_memo        PRI_pro_err (41)
/*$M Invalid option for printer; won't accept memo. */

#define PRI_err_invalid_fax_memo        PRI_pro_err (42)
/*$M Fax memo size of out-of-range. */

#define PRI_err_invalid_pri_phone_ext   PRI_pro_err (43)
/*$M Extended phone number not valid for non-fax printer */

#define PRI_err_invalid_fax_phone_ext   PRI_pro_err (44)
/*$M Extended phone number string has an invalid length */

#define PRI_err_invalid_pri_cover_doc   PRI_pro_err (45)
/*$M Invalid option for printer; won't accept cover page doc */

#define PRI_err_invalid_security_setting PRI_pro_err (46)
/*$M Invalid security setting. WAL clients must have remote
security setting */

#define PRI_err_bulk_trans_unexpected   PRI_pro_err (47)
```

+++ PRI - Print Services +++

```
/*$M Bulk data received more bytes than expected. This is a
sanity check. */
```

```
/* Errors not defined in the protocol */
```

```
#define PRI_err_bad_version PRI_dev_err (1)
/*$M Bad abstract link version when calling PRI. */
```

```
#define PRI_err_internal_rpc_error PRI_dev_err (2)
/*$M Internal rpc error occurred in PRI. */
```

```
#define PRI_err_no_notify_waiting PRI_dev_err (6)
/*$M No notification is pending. */
```

```
#define PRI_err_request_cancelled PRI_dev_err (7)
/*$M The given request has been cancelled. */
```

```
#define PRI_err_no_such_printer PRI_dev_err (8)
/*$M The specified printer is not known to this service. */
```

```
#define PRI_err_invalid_pointer PRI_dev_err (9)
/*$M The client tried to free an area using an invalid pointer. */
```

```
#define PRI_err_notify_timeout PRI_dev_err (10)
/*$M The asynchronous notify timeout has expired. */
```

```
#define PRI_err_empty_object PRI_dev_err (11)
/*$M Attempt to print 0 bytes with print data call. */
```

```
#define PRI_err_bad_migrate PRI_dev_err (12)
/*$M Bad migrate from optical disk.
This is an internal status sent from PRI_worker to the completion
status queue to handle bad optical disk migrates. */
```

```
#define PRI_err_lock_busy PRI_dev_err (13)
/*$M Can't get critical section lock (lock busy). */
```

```
#define PRI_err_bad_file PRI_dev_err (14)
/*$M Bad magnetic disk file (see syslog for errno) */
```

```
#define PRI_err_dest_redirected PRI_dev_err (15)
/*$M Can't chain printer redirection.
Can't redirect to a printer which is already redirected.
Also, if a printer X is redirected to a printer Y, printer Y can't be
redirected to a printer Z without first redirecting X to Z. */
```

```
#define PRI_err_no_printer_by_that_name PRI_dev_err (16)
/*$M No printer (or fax device) exists with the name specified. */
```

+++ PRI - Print Services +++

```
#define PRI_err_invalid_paper_size          PRI_dev_err (17)
/*$M No printers (or fax devices) handle the paper size specified. */

#define PRI_err_too_many_pages              PRI_dev_err (18)
/*$M Too many pages in request.
The limit is 64k documents or 64k * 32 pages, which nobody should ever hit. */

#define PRI_err_del_ageable                  PRI_dev_err (19)
/*$M Can't set delete after flag on object in ageable cache.
Only the system decides when to delete an object from an ageable cache. */

#define PRI_err_no_printers_for_options      PRI_dev_err (20)
/*$M No printers (or fax devices) can satisfy all options specified. */

#define PRI_err_invalid_num_pages           PRI_dev_err (21)
/*$M Invalid #of pages in request.
A request with no overlay option must have at least one page, and
a request with the overlay option must have at least two pages. */

#define PRI_err_migrate_timeout              PRI_dev_err (22)
/*$M Print services timed out while trying to retrieve a document. */

#define PRI_err_not_in_pri_cache             PRI_dev_err (23)
/*$M Can't put cache object to print in this cache.
When submitting an object in a cache to be printed (as opposed to a
document in a document service), the cache cannot be one of the
print caches specified in the print services configuration file.
This restriction is enforced to prevent objects from filling up
the caches used for document retrieval. */

#define PRI_err_duplicate_request            PRI_dev_err (24)
/*$M Duplicates error encountered on print_request table */

#define PRI_err_duplicate_option             PRI_dev_err (25)
/*$M Duplicates error encountered on print_option table */

#define PRI_err_duplicate_doc                PRI_dev_err (26)
/*$M Duplicates error encountered on print_doc table */

#define PRI_err_no_such_doc                  PRI_dev_err (27)
/*$M No such record exists in the print_docs table */

#define PRI_err_multiple_notify              PRI_dev_err (28)
/*$M Only one notify can be pending per PRI logon handle */

#define PRI_err_duplicate_service            PRI_dev_err (29)
/*$M Duplicates error encountered on print_service table */
```

+++ PRI - Print Services +++

```
#define PRI_err_invalid_NCH_name          PRI_dev_err (30)
/*$M Invalid 3 part Clearhouse name */

#define PRI_err_no_such_option            PRI_dev_err (31)
/*$M A print_options record was not found for the given request_id.
This error is probably due to an inconsistency in the MKF database,
where the print_options table is missing a record which is present
in the print_requests table */

#define PRI_err_must_delete_pri_cache     PRI_dev_err (32)
/*$M Delete after must be TRUE for this cache.
When printing objects which are default print services cache, the
delete_after flag must be TRUE. */

#define PRI_err_too_many_docs              PRI_dev_err (33)
/*$M Too many documents/objects in print request. */

#define PRI_err_too_many_options          PRI_dev_err (34)
/*$M Too many options on print request. */

#define PRI_err_invalid_service_type      PRI_dev_err (35)
/*$M Service type must be either cache service or document service. */

#define PRI_err_old_request                PRI_dev_err (36)
/*$M Print Services II cannot handle Print Services I requests.
All print requests generated by Print Services I must be deleted from
the database before Print Services II is started up. */

#define PRI_err_extra_papersize           PRI_dev_err (37)
/*$M Paper size configured on printer not configured on server.
Both the Print Server (has printer attached) and the Print Services
server (has print services running, normally an osar server) have
mag disk files indicating the printer configuration, and these files are
inconsistent, because a paper size configured on the print server is
not configured on the Print Services box. This paper size will not be
available to clients of Print Services. */

#define PRI_err_too_many_papersizes       PRI_dev_err (38)
/*$M Too many paper sizes configured on printer.
Both the Print Server (has printer attached) and the Print Services
server (has print services running, normally an osar server) have
mag disk files indicating the printer configuration, and these files are
inconsistent, because paper sizes configured on the print server are
not configured on the Print Services box. These paper sizes will not be
available to clients of Print Services. */

#define PRI_err_bad_cache                  PRI_dev_err (40)
```

+++ PRI - Print Services +++

```
/*$M Could not find the local cache.
Print services made an attempt to identify the local page cache, but
could not do so through NCH. */

#define PRI_err_bad_doc                PRI_dev_err (41)
/*$M Could not find the doc service.
Print services made an attempt to identify the document service
through NCH, but did not do so successfully. */

#define PRI_err_bad_ims                PRI_dev_err (42)
/*$M Could not determine the IMS info from NCH. */

#define PRI_err_invalid_printer_num    PRI_dev_err (43)
/*$M Invalid Printer Number. This should not happen */

#define PRI_err_invalid_eject_tray     PRI_dev_err(44)
/*$M Invalid Eject Tray. This is not a valid selection.
The option selected was not configured for this printer */

#define PRI_err_unexpected_exit        PRI_dev_err(45)
/*$M Unexpected exit from internal print routine. This
is just a sanity check which should never occur. If it
does occur, look at the log file to determine which function. */

#define PRI_trace_log_tuple            PRI_dev_err(46)
/*$M Not an error. This entry to the log was for tracing purposes only. */

#define PRI_WorkFlo_Printer_or_FAX_error PRI_dev_err(47)
/*$M WorkFlo Printer or FAX error, examine the Printer/FAX monitor for more info */
#define PRI_err_cache_timeout         PRI_dev_err(48)
/*$M Cache object not found.
Object from cache, and not optical cannot be found, do not attempt to print. */

#define PRI_err_notify_strlen          PRI_dev_err(49)
/*$M Notify string length mismatch in print request. */

#define PRI_err_address_to_string      PRI_dev_err(50)
/*$M IP address to string conversion error. */

#define PRI_err_string_to_address      PRI_dev_err(51)
/*$M String to IP address conversion error. */

#endif
```

25. PRI SUBROUTINES

+++ PRI - Print Services +++

25.1. PRI_cancel_request

error_typ

PRI_cancel_request (session_num, request_id, status_p)

ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)

ASE_request_id_typ request_id; /* IN: request to cancel */ (pg486.15)

PRI_request_status_typ* status_p; /* OUT: request status prior to cancellation */ (pg296.28)

/* Cancels a not-yet-completed print request. */

25.2. PRI_close_connection

error_typ

PRI_close_connection (session_num)

ASE_session_number_typ session_num; (pg483.24)

/* This entry point closes a connection previously opened by
PRI_open_connection. */

25.3. PRI_find_requests

error_typ

PRI_find_requests (session_num, filter_p, first_request_id, request_descs,
max_requests, num_requests_p, done_p)

ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)

PRI_request_filter_typ* filter_p; /* IN: A pointer to a structure containing values used to screen out unwanted requests for this call. The fields are: (pg312.26)

- user - Only requests for the user name specified here are returned. If all three strings of this field are null then all users are applicable
- queue_type - PRI_PT_UNKNOWN = all queues
PRI_PT_PRINTER = only printers
PRI_PT_FAX = only faxes
- priority - Only requests with the priority specified here are chosen. If this is 10 then all priorities are applicable.

+++ PRI - Print Services +++

request_status - Only requests with the status specified here are chosen. If this is a 0 then all statuses are applicable.
printer - Only requests for this printer are chosen. If all 3 strings of this field are null then all printers are applicable. */

ASE_request_id_typ first_request_id; /* IN: returns requests with (pg486.15)
id >= to this */
PRI_request_desc_typ request_descs []; /* OUT: ptr to array of request (pg309.33)
descriptors which will
hold results of query.*/
unsigned short max_requests; /* IN: max requests to
return */
unsigned short* num_requests_p; /* OUT: # requests returned */
bool* done_p; /* OUT: TRUE if no more (pg491.22)
requests for query */

/* Returns a list of pending requests into a client-supplied structure.
The search can be restricted to requests by a given user, and the
caller can specify the request_id at which the search is to begin.
The size of the returned list is limited to the size of the
client-supplied array.
If the caller does not have adequate permission to view a request,
the request_desc for that request will have only the request_id field
set, all other fields will be 0. */

25.4. PRI_find_requests2

error_typ

PRI_find_requests2 (session_num, filter_p,
start_position, get_docids,
request_descs_pp, num_requests_p, done_p)
ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
PRI_request_filter_typ* filter_p; /* IN: A pointer to a (pg312.26)
structure containing values
used to screen out unwanted
requests for this call.

The fields are:

user - Only requests for the user name specified here are returned. If all three strings of this field are null then all users are applicable

+++ PRI - Print Services +++

```

queue_type      - PRI_PT_UNKNOWN = all queues
                  PRI_PT_PRINTER = only printers
                  PRI_PT_FAX = only faxes
priority        - Only requests with the priority
                  specified here are chosen.  If
                  this is 10 then all priorities
                  are applicable.
request_status  - Only requests with the status
                  specified here are chosen.  If
                  this is a 0 then all statuses
                  are applicable.
printer         - Only requests for this printer
                  are chosen.  If all 3 strings
                  of this field are null then all
                  printers are applicable.
request_id      - Request request_id. Only
                  one request is at most returned.
                */

```

```

PRI_position_typ  start_position; /* IN/OUT: An opaque          (pg311.31)
                  position handle managed
                  by PS II. The array of
                  4 longs words
                  should be set to
                  zero the first time.
                */

```

```

bool             get_docids; /* IN: returns all associated (pg491.22)
                  docids with each
                  request */

```

```

PRI_request_desc2_typ **request_descs_pp; /* OUT: ptr to ptr (pg311.22)
                  of request descriptors
                  which will hold results
                  of query.*/

```

```

unsigned short*  num_requests_p; /* OUT: # requests returned */
bool*           done_p; /* OUT: TRUE if no more (pg491.22)
                  requests for query */

```

```

/* Returns a list of pending requests into ptr to ptr of linked list of request.
The search can be restricted to requests by a given user, and the
caller can specify the request_id at which the search is to begin.
The size of the returned list is limited to max_requests as
indicated by the client. To traverse the linked list use the
next_p field in the PRI_request_desc2_typ structure.
If the caller does not have adequate permission to view a request,
the request_desc for that request will have only the request_id field
set, all other fields will be 0.

```

If a filter value for the printer field is non-null, requests associated with that one printer will be returned in order.

+++ PRI - Print Services +++

This ordering is based on what will be printed first. If multiple calls to `PRI_find_requests2` are required to get a complete list, `start_position` should be used to keep track of the last returned requests. This is an array of 4 long words which should be initialized to zero for initial searches. For subsequent calls past the `start_position` array back. If `request_id` is specified in the filter only, at most only one request will be return.

If the input parameter, `getdocids` is set to `TRUE`, any associated document numbers will be returned with each request using the `num_docs` and `doc_array` fields in the `request_descs_pp` structure. The `num_docs` field is a count of the number of documents and the `doc_array` field is an array of the associated documents specifiers (`PRI_doc_specifier_typ`).

The `request_descs_pp` uses private heap space and needs to be returned with a `PRI_free_find_requests2` call.

*/

25.5. `PRI_free_find_requests2`

error_typ

```
PRI_free_find_requests2 (session_num, request_descs_pp, num_requests)
  ASE_session_number_typ session_num;          /* IN: session number */          (pg483.24)
  PRI_request_desc2_typ  **request_descs_pp; /* IN: ptr to ptr a linked list (pg311.22)
                                                of request descriptors
                                                which will hold results
                                                of query.*/
  unsigned short         num_requests;        /* IN: # requests */
```

/* Frees a request_desc-allocated structure gotten from `PRI_find_requests2` */

25.6. `PRI_free_printer_attrs`

error_typ

```
PRI_free_printer_attrs (session_num, num_printers, attrs_p_p)
  ASE_session_number_typ session_num;          /* IN: session number */          (pg483.24)
  unsigned short         num_printers;        /* IN: # of printers */
  PRI_printer_attr_typ** attrs_p_p;          /* IN: ptr to ptr to array of
                                                printer attrs */          (pg308.8)
```

25.7. `PRI_free_printer_attrs2`

+++ PRI - Print Services +++

error_typ

```
PRI_free_printer_attrs2 (session_num, num_printers, attrs_p_p)
  ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
  unsigned short num_printers; /* IN: # of printers */
  PRI_printer_attr2_typ** attrs_p_p; /* IN: ptr to ptr to linked list of (pg308.43)
                                     printer attrs */
```

/* Frees a service-allocated structure gotten from PRI_get_printer_attrs2 */

25.8. PRI_free_printer_status

error_typ

```
PRI_free_printer_status (session_num, status_p_p)
  ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
  PRI_printer_status_typ** status_p_p; /* IN: ptr to ptr to status */ (pg306.22)
```

/* Frees a service-allocated structure gotten from PRI_get_printer_status */

25.9. PRI_free_service_status

error_typ

```
PRI_free_service_status (session_num, status_p_p)
  ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
  PRI_service_status_typ** status_p_p; /* IN: ptr to ptr to status */ (pg307.33)
```

/* Frees a service-allocated structure gotten from PRI_get_service_status */

25.10. PRI_get_cache_name

error_typ

```
PRI_get_cache_name (session_num, cache_name_p)
  ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
  ASE_service_name_typ* cache_name_p; /* OUT: cache name */ (pg486.28)
```

/* Returns the name of the application print cache to be used in conjunction with this print service. This cache is to be used if the application does not have any other readily available cache. Any application which needs to print files or dynamically created objects may place them in this cache prior to calling PRI_print_docs. */

25.11. PRI_get_printer_attrs

error_typ

```
PRI_get_printer_attrs (session_num, num_printers_p, attrs_p_p)
```

+++ PRI - Print Services +++

```
ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
unsigned short* num_printers_p; /* OUT: #elements in attrs_p_p */
PRI_printer_attr_typ** attrs_p_p; /* OUT: ptr to ptr to array of (pg308.8)
printer attributes */
```

/* Returns a service-allocated array of PRI_printer_attr_typs, each of which describes the configuration and capabilities of a printer controlled by this print service. The array is allocated in the calling process' heap, and must be freed by calling PRI_free_printer_attrs(). Also returns the number of elements in the array. */

25.12. PRI_get_printer_attrs2

error_typ

```
PRI_get_printer_attrs2 (session_num, num_printers_p, attrs_p_p)
ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
unsigned short* num_printers_p; /* OUT: #elements in attrs_p_p */
PRI_printer_attr2_typ** attrs_p_p; /* OUT: ptr to ptr to linked list of (pg308.43)
printer attributes */
```

/* Returns a service-allocated linked list of PRI_printer_attr2_typ, each of which describes the configuration and capabilities of a printer controlled by this print service. The linked list is allocated in the calling process' heap, and must be freed by calling PRI_free_printer_attrs2(). Also returns the number of elements in the linked list. */

25.13. PRI_get_printer_status

error_typ

```
PRI_get_printer_status (session_num, printer, status_p_p)
ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
ASE_service_name_typ printer; /* IN: print server's name */ (pg486.28)
PRI_printer_status_typ** status_p_p; /* OUT: the address of a (pg306.22)
pointer to a service
allocated structure
describing the
specified printer's
current state */
```

/* Returns a service-allocated structure describing the service's current state. This structure is allocated in the calling process' private heap, and must be freed by calling PRI_free_printer_status. */

+++ PRI - Print Services +++

25.14. PRI_get_service_status

error_typ

PRI_get_service_status (session_num, status_p_p)

ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
PRI_service_status_typ** status_p_p; /* OUT: the address of a (pg307.33)
pointer to a service
allocated structure
describing service's
current state */

/* Returns a service-allocated structure describing the service's
current state. This structure is allocated in the calling process'
private heap, and must be freed by calling PRI_free_service_status. */

25.15. PRI_is_printed

error_typ

PRI_is_printed (session_num, request_id, timeout, done_p)

ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
ASE_request_id_typ request_id; /* IN: request id of request */ (pg486.15)
int long timeout; /* IN: #seconds to wait,
-1=>forever */
bool* done_p; /* OUT: TRUE if done, FALSE if (pg491.22)
not. If TRUE, no more
calls to PRI_is_printed
may be done for this
request_id. */

/* Waits (with timeout) for the request to be printed. The request waited
for is the last one submitted with notify_type = ASE_NOTIFY_ASYNCHRONOUS. */

25.16. PRI_logoff

error_typ

PRI_logoff(session_num)

ASE_session_number_typ session_num; (pg483.24)

/* This routine logs the client off from the Print Service, and
destroys the context for the given session. This should be done
when the client is through with the service in order to free server
resources. */

25.17. PRI_logon

+++ PRI - Print Services +++

error_typ

```
PRI_logon (service_name_p, leave_open, session_num_p, timeout_p)
  ASE_service_name_typ*  service_name_p; /* IN:  service to logon to */      (pg486.28)
  bool                   leave_open;    /* IN:  leave connection open after (pg491.22)
                                     the logon */
  ASE_session_number_typ* session_num_p; /* OUT: ptr to handle */      (pg483.24)
  unsigned short*        timeout_p;     /* OUT: obsolete--may be NULL */
```

/* This entry point logs a client on to a named Print Service.
This must be done before doing any work with the service. */

25.18. PRI_modify_request

error_typ

```
PRI_modify_request (session_num, request_id, num_options, options)
  ASE_session_number_typ session_num;    /* session number */      (pg483.24)
  ASE_request_id_typ     request_id;     /* request to be modified */ (pg486.15)
  unsigned short         num_options;     /* number of options[] */
  PRI_print_option_typ  options[];       /* array of print options */ (pg305.43)
```

/* Applies the given print options to a request. Options are applied
additively, i.e. existing options are not modified unless the
given print option list specifies a change.

If the request is already printing, it can't be modified unless the
printer, paper_size, or overlay option are changed, or the priority is
set to zero (which suspends the request). For these changes, the system
will automatically cancel and resubmit the request if processing on the
request has already started at the time of the modification request.
Note that the cancel/resubmit will potentially change the order in which
the request is printed relative to the other requests.

*/

25.19. PRI_open_connection

error_typ

```
PRI_open_connection(session_num)
  ASE_session_number_typ session_num;    (pg483.24)
```

/* This entry point opens a Courier connection for the given session with
Print Services. Multiple calls may be done on the same connection
(reducing overhead), but connections should not be left open for long
periods (e.g. while waiting for user input). A connection must not
be open already for this session. */

+++ PRI - Print Services +++

25.20. PRI_print_data

error_typ

```
PRI_print_data (session_num, data_len, data_p, fax_request,
                num_options, options, notify_type, request_id_p)
  ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
  unsigned               data_len;   /* IN: length of the data */
  char*                  data_p;     /* IN: pointer to the data */
  bool                   fax_request; /* IN: if this is a FAX request */ (pg491.22)
  unsigned short         num_options; /* IN: #PRI_print_option_typs in
                                     options[] */
  PRI_print_option_typ  options []; /* IN: array of print options */ (pg305.43)
  ASE_notify_option_typ notify_type; /* IN: must be either: (pg486.10)
                                     ASE_NOTIFY_NONE - call
                                     PRI find requests for
                                     status info.
                                     ASE_NOTIFY_ASYNCHRONOUS -
                                     call PRI_is_done to
                                     find if done.
                                     ASE_NOTIFY_SYNCHRONOUS -
                                     call will block until
                                     printed. */
  ASE_request_id_typ*   request_id_p; /* OUT: request id given to job */ (pg486.15)
```

/* Prints the given text or image data like it was a document. The caller must provide an array of print options for the request, and the type of notification requested. A request_id is returned.

The data in data_p must be in a format recognized by the destination printer (Print Services does not add any additional header or other information to the data input).

Only one request may have an asynchronous notify outstanding. Therefore if the notify_type is ASE_NOTIFY_ASYNCHRONOUS, the client must call PRI_is_printed or PRI_cancel_request before the next call which submits a request with a notify_type of ASE_NOTIFY_ASYNCHRONOUS.

*/

25.21. PRI_print_docs

error_typ

```
PRI_print_docs (session_num, num_docs, docs, fax_request,
                num_options, options, notify_type, request_id_p)
  ASE_session_number_typ session_num; /* IN: session number */ (pg483.24)
  unsigned short         num_docs;   /* IN: # ASE_page_range_typs in
                                     docs[] */
  PRI_doc_specifier_typ docs[];     /* IN: a list of document (pg307.14)
```

+++ PRI - Print Services +++

```

specifiers, each of which
contains a doc_id, page
range, and the type and
name of the service from
which they can be fetched.
If svc_name is a null field
then a default will be
used. */
bool          fax_request;    /* IN:  if this is a FAX request */ (pg491.22)
unsigned short num_options;  /* IN:  # PRI_print_option_typs in
options[] */
PRI_print_option_typ options[]; /* IN:  array of print options */ (pg305.43)
ASE_notify_option_typ notify_type; /* IN:  must be either: (pg486.10)
ASE_NOTIFY_NONE - call
PRI find requests for
status info.
ASE_NOTIFY_ASYNCHRONOUS -
call PRI_is_done to
find if done.
ASE_NOTIFY_SYNCHRONOUS -
call will block until
printed. */
ASE_request_id_typ* request_id_p; /* OUT: request id assigned to this (pg486.15)
print request */

/* This entry point prints a set of pages, specified in a list of
document page ranges.

When an element in the "docs" array specifies an object in a cache,
the cache must not be one specified in the print service configuration
file, because these caches are for document retrieval only.  Additionally,
if the cache is the default print cache specified in the NCH print service
description, the "delete_after" flag must be set to prevent overflow
of this cache.

When an element in the "docs" array specifies an object in a cache
the object is printed directly from the specified cache.  Therefore this
object must remain in the cache until the print is successful.

Only one request may have an asynchronous notify outstanding.  Therefore
if the notify_type is ASE_NOTIFY_ASYNCHRONOUS, the client must call
PRI_is_printed or PRI_cancel_request before the next call which
submits a request with a notify_type of ASE_NOTIFY_ASYNCHRONOUS.
*/

```

+++ PRS - Parser +++

26. PRS DECLARATIONS

```
/* ===== errors ===== */
#ifndef PC
#define err_PRS          err_INX
#else
#ifndef PCWS_h
#include "pcws.h"
#endif
#endif

#define PRS_err(_z)      err_encode(err_PRS,0,_z)

#define PRS_znomem      1000
#define PRS_err_nomem   PRS_err(PRS_znomem)
/*$M out of memory */

#define PRS_ztokensize  1001
#define PRS_err_tokensize PRS_err(PRS_ztokensize)
/*$M maximum token size exceeded */

#define PRS_znconspoflo 1002
#define PRS_err_nconspoflo PRS_err(PRS_znconspoflo)
/*$M numeric constant pool overflow */

#define PRS_zstgpoflo   1003
#define PRS_err_stgpoflo PRS_err(PRS_zstgpoflo)
/*$M string constant pool overflow */

#define PRS_zsymtaboflo 1004
#define PRS_err_symtaboflo PRS_err(PRS_zsymtaboflo)
/*$M symbol table overflow */

#define PRS_zbugsimp    1005
#define PRS_err_bugsimp PRS_err(PRS_zbugsimp)
/*$M bug in simple condition parser */

#define PRS_zrangl     1006
#define PRS_err_rangl PRS_err(PRS_zrangl)
/*$M unacceptable first operator of range */

#define PRS_zrangl_2   1007
#define PRS_err_rangl_2 PRS_err(PRS_zrangl_2)
/*$M syntax error in first part of range definition */

#define PRS_zrang3     1008
```


+++ PRS - Parser +++

```
#define PRS_err_rang3          PRS_err(PRS_zrang3)
/*$M unacceptable second operator of range */

#define PRS_zrang3_2          1009
#define PRS_err_rang3_2      PRS_err(PRS_zrang3_2)
/*$M syntax error in second part of range definition */

#define PRS_zrang4            1010
#define PRS_err_rang4        PRS_err(PRS_zrang4)
/*$M bug in range parser */

#define PRS_zmustbeint        1011
#define PRS_err_mustbeint    PRS_err(PRS_zmustbeint)
/*$M integer expected */

#define PRS_zmustbeposint     1012
#define PRS_err_mustbeposint PRS_err(PRS_zmustbeposint)
/*$M positive integer expected */

#define PRS_zkwdsexp          1013
#define PRS_err_kwdsexp      PRS_err(PRS_zkwdsexp)
/*$M KEYWORDS expected */

#define PRS_zlparenexp        1014
#define PRS_err_lparenexp    PRS_err(PRS_zlparenexp)
/*$M left parenthesis expected */

#define PRS_z2manykwds        1015
#define PRS_err_2manykwds    PRS_err(PRS_z2manykwds)
/*$M more keywords than allowed by integer */

#define PRS_zstgconsexp       1016
#define PRS_err_stgconsexp   PRS_err(PRS_zstgconsexp)
/*$M string constant expected */

#define PRS_zilglkwlist       1017
#define PRS_err_ilglkwlist   PRS_err(PRS_zilglkwlist)
/*$M comma or right parenthesis expected */

#define PRS_zormoofexp        1018
#define PRS_err_ormoofexp    PRS_err(PRS_zormoofexp)
/*$M OF expected */

#define PRS_zilglwhere        1019
#define PRS_err_ilglwhere    PRS_err(PRS_zilglwhere)
/*$M operand expected */

#define PRS_zexprttypemis     1020
```

+++ PRS - Parser +++

```
#define PRS_err_exprtypemis PRS_err(PRS_zexprtypemis)
/*$M where condition, expression type mismatch */

#define PRS_zexpropmis 1021
#define PRS_err_expropmis PRS_err(PRS_zexpropmis)
/*$M where condition expression and operator incompatible */

#define PRS_zbadwhere 1022
#define PRS_err_badwhere PRS_err(PRS_zbadwhere)
/*$M illegal where condition - unexpected end */

#define PRS_zkcons 1023
#define PRS_err_kcons PRS_err(PRS_zkcons)
/*$M illegal key constant */

#define PRS_zilglskeycon 1024
#define PRS_err_ilglskeycon PRS_err(PRS_zilglskeycon)
/*$M illegal key constant, unparenthesized */

#define PRS_ziltlkconslist 1025
#define PRS_err_iltlkconslist PRS_err(PRS_ziltlkconslist)
/*$M illegal key constant; comma or right paren expected */

#define PRS_zfindexpected 1026
#define PRS_err_findexpected PRS_err(PRS_zfindexpected)
/*$M FIND expected */

#define PRS_znottableid 1027
#define PRS_err_nottableid PRS_err(PRS_znottableid)
/*$M identifier not a table id */

#define PRS_ztableidexp 1028
#define PRS_err_tableidexp PRS_err(PRS_ztableidexp)
/*$M table id expected */

#define PRS_zviaexpected 1029
#define PRS_err_viaexpected PRS_err(PRS_zviaexpected)
/*$M VIA expected */

#define PRS_zkeyidexpected 1030
#define PRS_err_keyidexpected PRS_err(PRS_zkeyidexpected)
/*$M key identifier expected */

#define PRS_zidenexpected 1031
#define PRS_err_idenexpected PRS_err(PRS_zidenexpected)
/*$M identifier expected */

#define PRS_zkeyopexp 1032
```

+++ PRS - Parser +++

```
#define PRS_err_keyopexp      PRS_err(PRS_zkeyopexp)
/*$M key operator expected          */

#define PRS_zkeycondexp      1033
#define PRS_err_keycondexp   PRS_err(PRS_zkeycondexp)
/*$M key condition expected        */

#define PRS_zrangeexp        1034
#define PRS_err_rangeexp     PRS_err(PRS_zrangeexp)
/*$M range expected                */

#define PRS_zrang2           1035
#define PRS_err_rang2        PRS_err(PRS_zrang2)
/*$M key value expected in range condition */

#define PRS_zilglkconlist    1036
#define PRS_err_ilglkconlist PRS_err(PRS_zilglkconlist)
/*$M illegal key constant list      */

#define PRS_zstackoflo       1037
#define PRS_err_stackoflo   PRS_err(PRS_zstackoflo)
/*$M parse stack overflow           */

#define PRS_zilglqparam      1038
#define PRS_err_ilglqparam  PRS_err(PRS_zilglqparam)
/*$M illegal syntax for query parameter */

#define PRS_zilglfpconst     1039
#define PRS_err_ilglfpconst PRS_err(PRS_zilglfpconst)
/*$M illegal syntax for floating point constant */

#define PRS_zilglhexconst    1040
#define PRS_err_ilglhexconst PRS_err(PRS_zilglhexconst)
/*$M illegal hex constant           */

#define PRS_zni_qparam       1041
#define PRS_err_ni_qparam   PRS_err(PRS_zni_qparam)
/*$M query parameters not implemented */

#define PRS_zoparyoflo       1042
#define PRS_err_oparyoflo   PRS_err(PRS_zoparyoflo)
/*$M internal op array overflow     */

#define PRS_zni_arithop      1043
#define PRS_err_ni_arithop  PRS_err(PRS_zni_arithop)
/*$M arithmetic operators not implemented */

#define PRS_zcaryoflo        1044
```

+++ PRS - Parser +++

```
#define PRS_err_caryoflo      PRS_err(PRS_zcaryoflo)
/*$M internal constants array overflow      */

#define PRS_zunkconstyp      1045
#define PRS_err_unkconstyp   PRS_err(PRS_zunkconstyp)
/*$M bug: alloc cons: unknown constant type */

#define PRS_zmisrtquote      1046
#define PRS_err_misrtquote   PRS_err(PRS_zmisrtquote)
/*$M missing right string quote            */

#define PRS_zdealloccon      1047
#define PRS_err_dealloccon   PRS_err(PRS_zdealloccon)
/*$M bug: deallocate constant, index < 0   */

#define PRS_zilgldateop      1048
#define PRS_err_ilgldateop   PRS_err(PRS_zilgldateop)
/*$M illegal operator for date column      */

#define PRS_zilgltimeop      1049
#define PRS_err_ilgltimeop   PRS_err(PRS_zilgltimeop)
/*$M illegal operator for time column      */

#define PRS_znoncondatestg   1050
#define PRS_err_noncondatestg PRS_err(PRS_znoncondatestg)
/*$M date string not a constant            */

#define PRS_znoncontimstg    1051
#define PRS_err_noncontimstg PRS_err(PRS_znoncontimstg)
/*$M time string not a constant            */

#define PRS_znonconnumstg    1052
#define PRS_err_nonconnumstg PRS_err(PRS_znonconnumstg)
/*$M numeric string not a constant         */

#define PRS_zfpfail          1053
#define PRS_err_fpfail       PRS_err(PRS_zfpfail)
/*$M bad error from FP abstract            */

#define PRS_zni_unop         1054
#define PRS_err_ni_unop      PRS_err(PRS_zni_unop)
/*$M unary operators not implemented       */

#define PRS_znotdbcol        1055
#define PRS_err_notdbcol     PRS_err(PRS_znotdbcol)
/*$M identifier in where exp not db col id */

#define PRS_zmenuis          1056
```

+++ PRS - Parser +++

```
#define PRS_err_menumis      PRS_err(PRS_zmenumis)
/*$M expression operand mismatch for menuchoice type */

#define PRS_zilglmenuop     1057
#define PRS_err_ilglmenuop  PRS_err(PRS_zilglmenuop)
/*$M illegal operator on menuchoice type */

#define PRS_zkeycoflo       1058
#define PRS_err_keycoflo    PRS_err(PRS_zkeycoflo)
/*$M too many elements in key constant */

#define PRS_zkeywlcflo      1059
#define PRS_err_keywlcflo   PRS_err(PRS_zkeywlcflo)
/*$M too many keyword list constants */

#define PRS_zni_mpkeys      1060
#define PRS_err_ni_mpkeys   PRS_err(PRS_zni_mpkeys)
/*$M multi-part keys not implemented */

#define PRS_zpqlen          1061
#define PRS_err_pqlen       PRS_err(PRS_zpqlen)
/*$M length of parsed query area too small */

#define PRS_zkeyconstsz     1062
#define PRS_err_keyconstsz PRS_err(PRS_zkeyconstsz)
/*$M more items in key constant than in key */

#define PRS_zmenulen        1063
#define PRS_err_menulen     PRS_err(PRS_zmenulen)
/*$M length of menu item > 1 char */

#define PRS_zkeyconstype    1064
#define PRS_err_keyconstype PRS_err(PRS_zkeyconstype)
/*$M illegal key constant type */

#define PRS_zni_logop       1065
#define PRS_err_ni_logop    PRS_err(PRS_zni_logop)
/*$M this logical operator not implemented */

#define PRS_zpoolx          1066
#define PRS_err_poolx       PRS_err(PRS_zpoolx)
/*$M bug: illegal value for poolx */

#define PRS_zcoltype        1067
#define PRS_err_coltype     PRS_err(PRS_zcoltype)
/*$M bug: illegal db column type */

#define PRS_zexptype        1068
```

+++ PRS - Parser +++

```
#define PRS_err_exptype          PRS_err(PRS_zexptype)
/*$M bug: illegal expression type          */

#define PRS_zwhat                1069
#define PRS_err_what            PRS_err(PRS_zwhat)
/*$M bug: illegal what param to main      */

#define PRS_zMAXRESWD           1079
#define PRS_err_MAXRESWD       PRS_err(PRS_zMAXRESWD)
/*$M bug: MAXRESWD too small              */

#define PRS_zni_kwlst           1080
#define PRS_err_ni_kwlst       PRS_err(PRS_zni_kwlst)
/*$M bug: keyword lists not implemented   */

#define PRS_zkcon                1081
#define PRS_err_kcon           PRS_err(PRS_zkcon)
/*$M bug: illegal key condition type      */

#define PRS_znkeyitems          1083
#define PRS_err_nkeyitems      PRS_err(PRS_znkeyitems)
/*$M bug: num of key items wrong          */

#define PRS_zendarea            1084
#define PRS_err_endarea        PRS_err(PRS_zendarea)
/*$M bug: off end of area                 */

#define PRS_zkconop             1085
#define PRS_err_kconop         PRS_err(PRS_zkconop)
/*$M bug: illegal key compare operator    */

#define PRS_zidstate            1086
#define PRS_err_idstate        PRS_err(PRS_zidstate)
/*$M bug: illegal id state for scanner    */

#define PRS_zkeyeofexp          1087
#define PRS_err_keyeofexp      PRS_err(PRS_zkeyeofexp)
/*$M extra stuff at end of key condition  */

#define PRS_zconstsize          1088
#define PRS_err_constsize      PRS_err(PRS_zconstsize)
/*$M bug: incomplete implementation of larger integer */

#define PRS_zdclnamemis         1089
#define PRS_err_dclnamemis     PRS_err(PRS_zdclnamemis)
/*$M F_DOCCLASSNAME mismatch: not stg const */

#define PRS_zilgldclnameop     1090
```

+++ PRS - Parser +++

```
#define PRS_err_ilgldclnameop PRS_err(PRS_zilgldclnameop)
/*$M F_DOCCLASSNAME illegal operator */

#define PRS_zilgldef 1092
#define PRS_err_ilgldef PRS_err(PRS_zilgldef)
/*$M Illegal use of DEFINED */

#define PRS_zmenunmnotfound 1093
#define PRS_err_menunmnotfound PRS_err(PRS_zmenunmnotfound)
/*$M menu item name not found */

#define PRS_zlikeexp 1094
#define PRS_err_likeexp PRS_err(PRS_zlikeexp)
/*$M LIKE expected after NOT */

#define PRS_zunkop 1095
#define PRS_err_unkop PRS_err(PRS_zunkop)
/*$M bug: output unknown op */

#define PRS_zilgltype 1096
#define PRS_err_ilgltype PRS_err(PRS_zilgltype)
/*$M bug: illegal type left operand of binary op */

#define PRS_zdoctypemis 1097
#define PRS_err_doctypemis PRS_err(PRS_zdoctypemis)
/*$M F_DOCTYPE compared to expression of inappropriate type */

#define PRS_zilgldoctypeop 1098
#define PRS_err_ilgldoctypeop PRS_err(PRS_zilgldoctypeop)
/*$M F_DOCTYPE compared to expression using operator other than = or != */

#define PRS_zdoctypebug1 1099
#define PRS_err_doctypebug1 PRS_err(PRS_zdoctypebug1)
/*$M F_DOCTYPE implementation bug -- zero len constant string */

#define PRS_zdoctypstglen 1100
#define PRS_err_doctypstglen PRS_err(PRS_zdoctypstglen)
/*$M F_DOCTYPE compared against a string that is too long */

#define PRS_zilgldoctypepstgval 1101
#define PRS_err_ilgldoctypepstgval PRS_err(PRS_zilgldoctypepstgval)
/*$M F_DOCTYPE compared against illegal string value */

#define PRS_zneqstg1 1102
#define PRS_err_neqstg1 PRS_err(PRS_zneqstg1)
/*$M F_DOCTYPE != 'string' requires string length = 1 */

#define PRS_zdoctypekludge 1103
```

+++ PRS - Parser +++

```
#define PRS_err_doctypekludge PRS_err(PRS_zdoctypekludge)
/*$M F_DOCTYPE = 'abcde' internal kludge only */
```

27. PRS SUBROUTINES

27.1. PRS_free_query

```
void
PRS_free_query (query_p)
    INX_query_typ      *query_p; /* IN: ptr to query area. */           (pg251.4)

/* Returns the memory allocated for a query area */
```

27.2. PRS_parse

```
error_typ
PRS_parse(session, query_pp, arealen_p, keycond_p, filter_p, parselen_p)
    ASE_session_number_typ session; /* IN: session handle returned from   (pg483.24)
                                     IMS_logon (not INX_logon) */
    INX_query_typ      **query_pp; /* IN/OUT: ptr to ptr to query area.   (pg251.4)
                                     Initially passed in as zero,
                                     and returned as a pointer to
                                     an area containing the query
                                     structure. Subsequent calls
                                     may pass in the prior value
                                     returned for *query_pp, and then
                                     this area will be reused and
                                     resized if need be. Note:
                                     PRS_free_query(*query_pp) should
                                     be called to return the memory
                                     when done with it. */
    short              *arealen_p; /* IN/OUT: passed in as length of query
                                     area. Initially passed in as
                                     0 when *query_pp is also 0.
                                     Returned as new length of query
                                     area (query area will get
                                     resized if need be). */
    char               *keycond_p; /* IN: key condition string */
    char               *filter_p; /* IN: filter condition string */
    short              *parselen_p; /* OUT: ptr to number of characters
                                     parsed. Set if error returned
                                     to help caller determine error
                                     but caller must determine if
                                     error was in filter or key. */
```


+++ PRS - Parser +++

/* This routine takes ascii strings which indicate a query condition and converts them to the query structure required as an input to the INX_find_DIRS routine. The INX_find_DIRS routine can then be used to find all documents which satisfy the key condition string AND the filter condition string.

When INX_find_DIRS is called with the query structure returned by this routine, it will inspect each record specified in the key condition, and then only return this record if it satisfies the given filter condition. To optimize performance when using this routine, the key condition should be an inverted field with as few matches as possible. For example, if two inverted fields exist for each document, and one is a person's name, and the other is the city he lives in, the persons name should be in the key condition, and the city should be in the filter condition.

If this routine is called multiple times in a row, the memory area that *query_pp points to can be reused. The first call will allocate the area and return the length of it. The second and subsequent calls can be passed in the area and length allocated on the prior call, and they will resize the area larger if need be, and continue to return the pointer and length of the memory area. Note that any call may change the value of *query_pp if the area gets resized.

The key condition string and filter condition string use the following operators with the restrictions documented in the CONSTRAINTS section. If an operator is not listed in the constraints section, the operator can be used in either a key or a filter condition.

AND	logical and
OR	logical or
NOT	logical not
<	less than
<=	less than or equal to
=	equal to
>	greater than
>=	greater than or equal to
!=	not equal to
+	addition
-	subtraction
*	multiplication
/	division
IN RANGE	a range is specified with the syntax "IN RANGE <op> <constant> <op> <constant>", where <op> is <, <=, >, or >=, and constant is a numeric or string constant.
LIKE	Similar to. The pattern on the right hand side of the LIKE operator must be a single quoted character string

+++ PRS - Parser +++

which can contain '?' to indicate match any character,
and '*' to match any sequence of characters.

DEFINED(x) The defined function returns non-zero if the given column
name "x" is non-null, and zero if the value is null.

CONSTRAINTS

1. DEFINED and NOT DEFINED can only be in filter condition.
2. != can only be used in filter condition.
3. != and = are the only operators that can be used with F_DOCTYPE.
4. + at the beginning of an FP number can only be used in filter
condition.
5. LIKE and NOT LIKE can only be in filter condition for a string index.
6. LIKE and NOT LIKE cannot be used in a key OR filter condition as a
date index.
7. IN RANGE can only be used in key condition.

The operands are numeric constants, string constants, column names, and
key names.

Numbers are in the format "+ddd.dddE+eee" where "ddd" stands for 0 or more
numeric digits, "+" stands for "+" or "-" ("+" is optional), "eee" stands
for 0 to three digits of the exponent, and "." is a decimal point
(optional), and "E" is the beginning of the exponent (optional).

String constants are surrounded by single quotes, and an embedded single
quote within a string is represented by two consecutive single quotes.

Column names are found by calling `INX_get_all_index_descs` or
`INX_get_index_desc`, and using the "name" field in the
`INX_index_desc_typ` record returned.

Key names are found by calling `INX_get_all_key_descs` or
`INX_get_key_desc`, and using the "name" field in the
`INX_key_desc_typ` record returned.

The F_DOCTYPE column is a numeric value which can be compared against the
symbols IMAGE, FORM, MIXED, TEXT, and OTHER to test for the
various image types.

Columns which are of type `INX_VT_DATE` may be compared against date
strings which conform to the default date mask. The date strings

+++ PRS - Parser +++

must be surrounded by single quotes. Date type columns can also be compared against a number which is the integer format of the date (see the description of ASE_date_typ in file AS_externals.h).

Columns which are of type INX_VT_MENU may be compared against the "item_names" in the INX_menu_item_typ record surrounded by single quotes, or by a single letter also surrounded by quotes which is one of the "value" fields of the INX_menu_item_typ record. Note that you should not define item names which are single characters because doing so will make the syntax supported by this routine ambiguous. In other words when creating Menu items from say WFD, using the "Build a Menu" option do not use 1 character 'Text to Display' item names because this routine will interpret the 1 character menu item name as the return code. And then using a query against this item name as a filter condition would give a 'No doc's found' results. But, if there really is the need to create 1 character menu item names then the return code of these item names should match the 1 character item name. This way this routine will work correctly for 1 character menu item names.

Columns which are of type INX_VT_MENU cannot be compared using the less than or greater than operators. Only '=' and '!=' are allowed.

The key string must contain just one key name and one condition. Note that key names must be used, and a non-inverted column name may not be used. Examples of key strings are:

```
F_DOCNUMBER = 100000
F_DOCNUMBER IN RANGE >= 100000 <= 200000
```

The filter string is limited to no more than 85 operands and operators. Parenthesis may be also used in the filter to override precedence. Examples of filter conditions are as follows:

```
F_PAGES > 1 AND (F_ENTRYDATE > '10/15/92' OR F_DOCTYPE = MIXED)
F_PAGES > 1 AND (F_ENTRYDATE > 7000 OR F_DOCTYPE = IMAGE)
```

Note that F_PAGES with a NULL value actually represents a single page document. Therefore, the query result of 'F_PAGES IS NULL' will also include single-page documents.

*/

28. SEC DECLARATIONS

```
/*****\  
*                               SEC                               *\  
\*****/
```

1. Introduction

This document details the functional specification for the revised Security Services. The new SEC will utilize MKF for the storage of information whereas SCT used SKF. Additionally, the existing SEC server software will also be rewritten to support the new SEC design. This design contains many new terms which are defined in the glossary at the end of this document.

In order to use MKF efficiently, SEC client and server stubs will need to be generated (SCT relied upon SKF for remote operations). To increase performance over repetitive MKF database requests, SEC will offer a client and server caching mechanism. Security objects will be placed in the client/server cache on an as-needed basis.

In addition to changing the security service's internal functionality, a set of new SEC interface calls has been designed. These new calls primarily satisfy the system administration software. This will allow a system administrator to view the security database and log-files as well as to control the security environment of the system. SEC will also offer a new set of underlying configurable and non-configurable options discussed in this document.

In order to migrate from the SCT storage method to the SEC storage method, a conversion utility will be offered. The key conceptual difference between SCT and SEC is the distinction between users and groups. In SCT, a user/group differentiation is not made. As a result, the conversion utility will need to decide which SCT entities are users and which are groups. The impact upon the customer would be that they will need to review their entity hierarchies using SEC to verify that the conversion will be performed as desired.

SEC will also provide a tool to debug the security service. This tool is intended to be used by your service representative.

2. General Requirements

SEC is a super-set of the existing FileNet security system. The goals of this service are:

- o Provide WAL-style access to the security data base.

+++ SEC - Security Services +++

- o Create an "open" design so that modules may be replaced with minimal code change to accommodate future security requirements.
- o Make SEC more robust by providing a greater variety of data base search routines, and by providing enhanced exception logging.
- o Close existing security holes inherent to SCT.
- o Provide a design which can move towards compatibility with other operating system environments.
- o Provide the ability to enforce concurrent license agreements.

3. SEC Components

The SEC architecture will include five major items:

- o a tool (and interface) to allow the system administrator to view and create reports based on the contents of the security (MKF) data base (Xsec_admin);
- o a tool to convert from SCT's SKF data base to SEC's MKF data base; the tool will allow the system administrator to view the old security SKF data base as well as show what changes will occur when the conversion tool is run;
- o a programmer interface by which applications and services may determine access restrictions;
- o the capability to view and debug key elements of this service;

4. SEC Configuration

SEC will allow the customer to configure the security service based on options which range from information logging to restricting access. Default values will be used for the configuration variables until altered by the customer through the system administration software. Most of these defaults may be altered by the customer.

The SEC_init program (discussed later) will initially create a default set of records within the MKF database. It is not expected, however, that SEC_init will be run manually. This program is typically started by the SEC1 shared library only if it detects that the security tables are empty. As well, SEC_init is launched from SEC_convert in order to add security objects which may not have existed in the old security services.

4.1 System Level Configuration

The system level configuration, also known as the system defaults, may only be controlled by the SysAdmin user. All of these settings will be applied to every user, except in the case of overrides (discussed below).

- o terminal security
- o no function definition ok
- o time usage restrictions
- o maximum logon sessions allowed per user

+++ SEC - Security Services +++

- o password restrictions
- o security logging
- o allow override

4.1.1 Terminal security

The terminal security attribute may be set at the system level. This attribute may be overridden by any object (user, group or device). See the section on Device Controls for a functional overview.

4.1.2 No function definition ok

Function security within the FileNet environment is driven by the FileNet set of applications. Typically, in order to restrict users or groups from performing various application options the calling program will request that security services validate the user's access to a specific function. This validation process merely checks for an intersection between the user's transitive closure and the specified function's list of members. Security services will inform the application if access will be granted depending upon the result of this comparison and depending upon the value of the function definitions field.

The attribute which governs function definitions may be set "on" or "off". When this attribute is "on" it implies that function access will be granted, even though the function does not exist in the security data base. If a function does exist in the security data base, then access will be checked. A more restrictive approach to function security would be to set this attribute to "off" requiring that all functions be defined. As well, by making the wildcard "(ANYONE)" a member of a function, all users are then given access to that function.

4.1.3 Time usage restrictions

Time-usage restrictions govern the time frames in which a user, group or device may be used on a given system. An indepth discussion of time usage restrictions is presented in the section entitled "Logon Controls".

4.1.4 Maximum logon sessions

The maximum number of logon sessions allowed per user may be set on a system-wide level. This attribute will control how many concurrent logons a single user may have at a given time. This may also be set on a per user basis.

4.1.5 Password restrictions

A variety of password restrictions exist which may be set only at the system level. These include:

+++ SEC - Security Services +++

- o password special characters
- o password minimum lengths
- o password days until change required
- o password grace period for password change
- o password failed attempts until account is suspended
- o password time in which failed attempts may occur to cause suspension

Each of these attributes is discussed in detail in the "Password Controls" section of this document.

4.1.6 Security logging

Logging may be set at the system or at the object level. The set of logging attributes are discussed in detail under the "Reporting Controls" section of this document.

4.1.7 Allow override

The allow override attribute allows the SysAdmin user to determine if other administrators should have the ability to override the system defaults. If this value is turned "on", then non-SysAdmin administrators may set these values on a per-object basis. If the allow override attribute is "off", then only SysAdmin may effect overrides on a per-object basis. The values which may be overridden are time usage restrictions, security logging, terminal security and maximum logon sessions.

4.2 Object Level Configuration

A security object is an entity which affects the security of the system. Objects may be users, groups or devices. Each object may be configured in a variety of ways. The configuration and behavior of an object may vary depending upon its class.

- o object name
- o object id
- o object class
- o device class
- o administrative attributes
- o device security
- o user language
- o primary group
- o administrative group
- o session group
- o time usage restrictions
- o expiration time
- o security logging
- o system override
- o maximum logon sessions

+++ SEC - Security Services +++

- o object comments
- o password

4.2.1 Object name

Each object should have a unique name. An object name may be as long as 80 characters and as short as 1 character in length. Object names may be changed at any time by the system administrator without affecting security. Object names consist of three parts: the name (up to 40 characters), the domain (up to 20 characters) and the organization (up to 20 characters).

4.2.2 Object id

When a security object is created by the system administrator, security services automatically assigns a numeric "id" to that object. This id may be used in access permissions for documents or cache objects. The id is not modifiable by the system administrator.

4.2.3 Object class

The object class determines the type of object. Object classes may be users, groups, devices or system (reserved). The class of an object will determine its behavior within the security domain. Once a class is assigned to an object, the class of that object may not be altered.

4.2.4 Device class

This attribute defines the type of device that this object represents. Examples of device classes are terminals and printers.

4.2.5 User language

This field is the language for a given user. This language string is accessed by a variety of FileNet applications for the sake of display purposes. This field has no bearing upon PC applications.

4.2.6 Primary group

A primary group is like any other group within security services. The primary group represents the default set of security restrictions which will be applied by the various applications when creating a FileNet object (eg. annotation, Folder, etc). It is not necessary to assign a primary group for user objects. Assigning a primary group to a user automatically makes that user a member of that group. Changing a user's primary group will have the effect of removing membership to the previous primary group. The primary group does not apply to device objects. A group object is its own primary group. A group's primary

group may not be changed. When a group is deleted from the security database, all users with that primary group will have their primary group changed to (NONE).

4.2.7 Administrative group

An administrative group is like any other group within security services. The administrative group is a stamp related to the system administrator which created the object. The stamp is the administrator's administrative group. In a sense, it is an ownership stamp used to determine if a given system administrator can modify this object. Modification of an object is determined through comparing the transitive closure of the administrator's administrative group to the administrative group of the target object. When a group object is deleted from the security database, all users with that administrative group will have their administrative group changed to (NONE). The assignment of an administrative group to a user does not make that user a member of that group.

4.2.8 Session group

The session group may control logon access to the security service. A session group is like any other group within security services. A session group may only be assigned if the system defaults setting allows overrides. The SysAdmin user may bypass this rule, however. If a session group is assigned to a user, then the override setting within the session group will be evaluated. The values which the override setting affects relative to the session group are terminal security, security logging and logon times. If a user object has its override value set to "on", then the user's override settings will supersede those of the session group. When a group object is deleted from the security database, all users with that session group will have their session group changed to (NONE). The assignment of a session group to a user does not make that user a member of that group.

4.2.9 Time usage restrictions

The time usage restrictions may apply to all objects (ie. users, groups and devices). Time usage limits are imposed upon logon and during the length of a security session. The use of time restrictions in a group object will only have an impact if that group object has been assigned as a user's session group. Time usage restrictions are discussed in detail in the "Logon Controls" section.

4.2.10 Expiration time

The expiration time of an object is the time at which the object will no longer be usable. Objects may be set to expire on a specific

+++ SEC - Security Services +++

date/time or objects may be set to not expire at all.

If the object is a user, the user will be unable to logon or use the system beyond the set expiration time. If the object is a device, that device will become unusable beyond the set expiration time. If the object is an assigned session group, then users with that session group will not be able to logon. This last characteristic allows a system administrator to cancel the logon capabilities for groups of users through a single object.

4.2.11 Security logging

Security logging may be set on a per object basis. Whether information is logged is a factor of the user, session group and device object.

4.2.12 System override

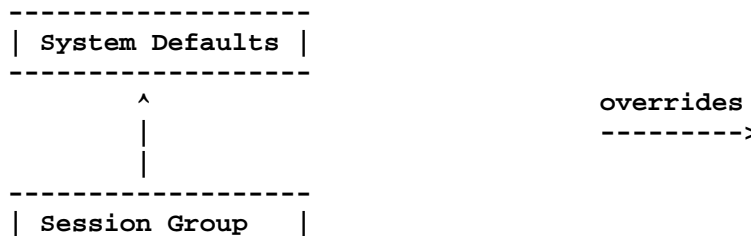
The system override attribute causes the security service to use the object's attributes versus the system attributes in certain instances. The attributes which the system override may alter are:

- o device security
- o time usage restrictions
- o security logging
- o maximum concurrent sessions per user

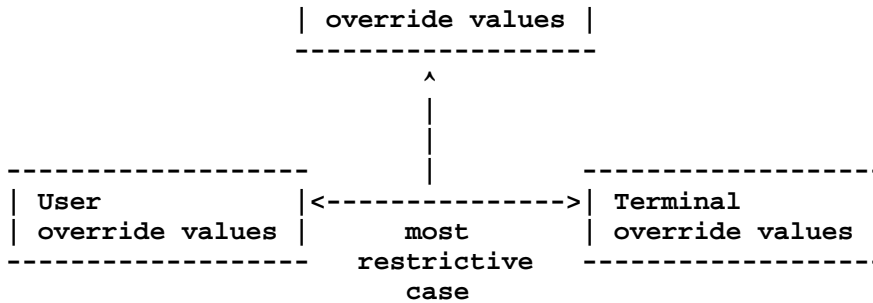
All of the above attributes apply to users, session groups and devices except for the "maximum concurrent sessions", which only applies to users. A set of rules will be invoked for the user, device and primary group objects relative to the override value:

- o a user object's settings always supersedes the session group settings and the system defaults settings for the aforementioned attributes
- o a device's settings, when compared to a user or a session group will invoke the most restrictive case for the logged-on user; like the user object, a device's settings always supersede those of the system

Figure 4-1: Override Hierarchy



+++ SEC - Security Services +++



The use of system overrides may be as simple or as complex as an administrator may desire. At the simplest level, override values are not set for any objects, hence the system default settings are used. The ability to apply system overrides is controlled by the SysAdmin user. See the description of the "allow overrides" field for system defaults. At the next level, users may be assigned a session group which has the system override value turned "on". The result would be that a user's logon session would be governed by the settings within the session group. This allows an administrator to group session capabilities for collections of users without having to manually set these values for each user object.

A user may have their system override value turned "on". In this case, the user's settings take precedence over both the session group's settings and the system default settings. An administrator who explicitly sets a user's override settings may not want to assign a session group to that user since the session group's values will be ignored. However, if an assigned session group were to be expired, then a user would not be able to logon irregardless of that user's system override settings. In this case, the assignment of a session group to a user depends upon the administrator's desired goals.

A device may have its system override value turned "on". In this case, the device's settings will take precedence for a user's logon session. This assumes that neither the user nor the session group have the system override value turned "on".

In the event that the session group, user and device have the override value turned "on", security services will need to evaluate which settings to apply for this logon session. First, the user's settings and the session group's settings will be resolved. Since the user has the system override value "on", this will be the settings which security services will use to compare to the device's settings. When comparing these settings to the device, in this case, the most restrictive settings between these two objects will be enforced.

4.2.13 Maximum concurrent sessions

This attribute controls the maximum number of concurrent logon sessions which a user may have at a given time on a given system. When the maximum is reached, subsequent logons will result in the appropriate error. This attribute is useful to a system administrator in managing the number of logons, preventing any single user from impacting the concurrent license agreement.

4.2.14 Object comments

Each object is accompanied by a comment field. The comment field is a free-form field which may be used to further describe the object. An example of a comment for the device object "printer1" may be: "ricoh, 2nd floor, rm 232".

4.2.15 Password

Passwords only apply to user objects. The password is used upon logon to the system. The password is bounded by the system password attributes.

4.3 Object Templates

Object templates exist for the convenience of the system administrator. A template is an actual object within the security data base. It allows the system administrator to specify default attributes for an object upon creation. At the very least, the creation of an object requires that an object name and object class are specified. The remainder of the object will be set to the corresponding template values. A template exists for users, groups and devices. The values in these templates are configurable by the SysAdmin user.

5. Security Features

5.1 Logon/Logoff Controls

- o A user, group or device may be limited on a time-of-day and/or a day-of-week basis.
- o A user, group or device may have an accompanying expiration date.
- o A user may be denied logon access if that user already has a pre-defined number of current logon sessions to a given system.
- o A group will not be allowed to logon.
- o Multiple logons from the same endpoint to the same system using the same terminal name could result in erratic behavior.
- o Account suspension may occur after a configured number of failed password attempts have occurred over a configured period of time.
- o Rebooting the security service will not log users off due to an

automatic relogon feature.

5.1.1.1 Time usage restrictions

A user may be restricted based upon the time of day and/or the day of the week. The time of day may be specified by providing starting and ending times which are a combination of the hour and minutes. The day of the week time frame may be expressed using starting and ending days. For example: start hour/min=08:00, end hour/min=17:30; start day=Monday, end day=Friday. This user may logon and work beginning 8:00am through 5:30pm on each weekday. Once the 5:30pm time has been reached that user should not be able to perform any more work with the servers on the system on which that user is defined. Local workstation tasks may still be acceptable, however, since the imposed time restriction is stored and checked on the security server, not the workstation.

This logon/time usage restriction may be imposed on a per user basis or on a system-wide basis. In addition, "wild cards" are permitted for the start and end values. A "wild card" will force the start value to the minimum and the end value to the maximum for that attribute (eg. a "wild card" for starting hour will force the hour to be 00 and a "wild card" for ending hour will force the hour to be 23). If all time restriction attributes are set to a "wild card", then no time usage restriction will be imposed.

Time usage restrictions may also be imposed upon devices and groups in addition to users. Each user may have a session group. In order to logon, a user must use a terminal. Each of these objects may be set to override the system wide settings for time usage. Only those objects which have the system override attribute set may have their usage time evaluated. If none of these three objects has their system override attribute set, then the system wide time usage restriction will apply. For a detailed explanation on override hierarchy see the section describing 'system overrides'. An example follows.

User "fred" is set to override the system wide settings and has this set of time usage attributes: start time=08:00am, end time=05:30pm, start day=Monday, end day=Thursday. User "fred" then tries to access a terminal which is also set to override the system wide settings and has this set of time usage attributes: start time=05:00am, end time=01:00pm, start day=Wednesday, end day=Friday. As a result, "fred" will inherit the following time usage attributes: start time=08:00am, end time=01:00pm, start day=Wednesday, end day=Thursday. If the current time falls outside of this time range, then "fred" will not be able to use the system from this terminal. If user "fred's" session group had the system override value set (in this case), then it would be ignored since a user's and terminal's settings always supersede the session group's settings.

5.1.2 Expiration dates

Each user, group and device may be set to expire at a specific point in time. Through the use of expiration dates, security objects may be made unusable without actually deleting the object.

Expiration times are imposed upon logon and during a session. Upon logon, if either a user, that user's session group, or the terminal's expiration time is less than the current time, then a logon will be denied. If a successful logon has occurred, then the most restrictive expiration time will be imposed during that user's session. If the expiration time occurs during the user's session, that user will no longer be able to use the system onto which that user had previously successfully accessed. Time usage restrictions (discussed in the previous subsection) may also apply.

Users, groups and devices may also be set to never expire. If an expiration is set, however, the ability of a user to logon may be affected depending upon the object which has expired. If a group is expired, then every user who has that group set as their session group will not be able to logon. If a terminal has expired, then no user may logon from that device. Printer/fax devices may not be expired. If a user is expired, then that user will not be able to logon.

5.1.3 Maximum number of sessions

A user may be restricted from additional logons to a system if that user's maximum number of sessions has been exceeded. The maximum number of sessions per user may be set on a system wide basis or on a per user basis depending if that user has the system override attribute set. If a user is restricted from logging onto a system due to an excessive number of logon sessions, that user may access the system by logging-off another active session executed by that user.

5.1.4 No group logons

In the older versions of security services, no distinction could be made between a user and a group. All of these "entities" could logon and have members. In the new security services, a clear distinction is made between a user and a group. Groups may not logon; only users may logon. For more details regarding the distinctions between the old and new security services, see the section on data base conversion.

5.1.5 Multiple logons from the same end-point

Although the FileNet set of applications (both UNIX-based and DOS-based) will control multiple logons from the same end-point to a given security service, a user should be aware of the behavior caused

by this action. Once a user logs onto a security service from a PC, an in-memory record of that session is recorded on the server. If the PC loses the connection, or if the user powers-off the PC without performing a graceful logoff, that in-memory logon session will remain on the server. The security service will assume that that user is still logged-on until one of two actions are performed: a user logs onto the security service again from the same endpoint (overwriting the previous logon from that end-point); or the system administrator clears that in-memory logon instance using the provided set of tools. Note that if a second application logs onto the same system from the same PC, the first application will lose its logon instance.

5.1.6 Failed password account expiration

A system administrator may configure a system to expire a user account if a configured number of failed password/logon attempts has been achieved on that account within a configured period of time. Once the user's account has been suspended, all subsequent retry attempts will fail until a system administrator reactivates that user's account.

5.1.7 Automatic relogon

When the FileNet software is recycled, the in-memory record of logged-on user sessions disappears. However, as the previously logged-on clients access the security service (directly or indirectly), then an automatic relogon will occur. This automatic relogon authenticates the user in a manner identical to a normal logon. Due to this feature, rebooting a server which contains security services, will not cause disruption to the server or client software.

Due to the automatic relogon feature, unexpected behavior may result when the FileNet software is recycled. If a system administrator logs-off a user through the provided tools, that user will receive errors whenever an attempt is made to access the various services. This is due to the fact that this user has had their session terminated. However, once the software is recycled, this user will be automatically relogged-on when access to the services is attempted. This occurs because the security service can no longer distinguish between a logged-off account and a normal logged-on account.

5.2 Password Controls

- o A user may change their password at any time.
- o A cleartext password will not reside in memory, on disk or on the network.
- o Password lengths may range from 0 characters to 8 characters and are configurable by the system administrator.
- o Password special character requirements may be imposed by the

system administrator.

- o Passwords may be set to expire, requiring the user to change their password after a given time interval.

5.2.1 Changing a password

A user may change their own password at any time using the provided set of tools. A user may not change any other user password, unless the appropriate system administration attributes are set. The password which the user selects must abide by other system wide restrictions. When a user changes their password, they must logoff then logon again, otherwise various access failures could occur. The user may change their password through the use of SEC_tool or the Security Administration application.

Passwords only apply to users, not groups or devices.

5.2.2 Passwords in cleartext

No password will reside in clear-text for an extended period of time in memory or in the security data base. The only point at which encrypted passwords may be decrypted, such as when a remote user performs a logon, will be on the server on which the security service resides. In this case, the user typed-in password will be encrypted using a proprietary algorithm then shipped across the network to the security service. The security service will then decrypt the received password, then re-encrypt this password using the standard UNIX "crypt" function (the DES algorithm). The encrypted passwords are then compared on the server for authentication of the user.

5.2.3 Password length restrictions

The system administrator may set a minimum required password length of 0 through 8. The maximum length of a password is 8 characters. All newly created/changed passwords must adhere to the set restriction. If the restriction changes, it does not affect existing passwords in the security data base that may not adhere to this rule. Typically, the longer the password, the more secure the password may be from a brute force attack.

5.2.4 Password special character requirement

The system administrator may require that all users create passwords which contain a special character. A special character may be any typed character other than a-z or A-Z (eg. ";"). This causes a certain level of ambiguity, making a password more difficult to guess.

5.2.5 Password expiration

The system administrator may set all user's passwords to time-out if not changed after a configured period of time. The administrator will set the time until password renewal is required (eg. 30 days) from the time of last change, and the administrator will set a grace period in which the user may receive warnings that the renewal date is approaching. If the user does not change their password by the renewal date, then that user's account will be automatically suspended upon the next use of that account. As well, when the renewal period is changed, the calculated password expiration time will be changed to the new value for each object.

Setting the password expiration value also has an effect upon the use of a newly created user object. If the password expiration value is set, then it is required that the user change their password for a first logon account or a newly unexpired account. If the password to this newly created account is not changed prior to the next logon, then the user object will be automatically expired. When the user changes their password at this point, the user must logoff then logon again for their session to acquire the new password. Otherwise, various security failures may occur for that user throughout the system. The act of warning the user that the "grace period" has been entered is strictly a function of the application and not the security service, per se. The security service does provide the capability to display when the password will expire and when the grace period has been entered.

5.3 Device Controls

- o A user may not access a terminal/device based on membership.
- o A user may not access a terminal based on time usage restrictions.
- o A system administrator may add and delete devices.

5.3.1 Terminal/device membership access

If the terminal security attribute is set by the system administrator, then access to that terminal will be determined based upon the membership intersection of the user and the membership intersection of the terminal. Like a user object, a terminal object may be a member of many groups. In determining terminal access, the security service will identify the user's membership transitive closure and the terminal's membership transitive closure. The transitive closure refers to the extended membership of an object. If the user belongs to group "A" and group "A" belongs to group "B", then the user's membership transitive closure is A and B.

Similar to other security attributes, implementation of terminal security may be on a system wide level, a per user level, a group level or a device level. If either the user, group or terminal have the system

+++ SEC - Security Services +++

override attribute set, then terminal security will be implemented.

For example, the system wide terminal security attribute is turned "off", and the group "accounting" has the system override attribute and the terminal security attribute turned "on". As a result, terminal security will be imposed on any user who has their session group set to "accounting". The user "fred", whose session group is "accounting", attempts to logon to the security service from a terminal. Neither "fred" nor the terminal have the override or device security attribute turned "on". User "fred" is a member of the groups "accounting", "payroll" and "personnel". The terminal is a member of the groups "accounting" and "payroll". When "fred" successfully logs onto the security service he will only be able to access documents which are the intersection of his transitive closure and the terminal's transitive closure. In this simple example, "fred" may only manipulate those objects which have the security attributes of "accounting" and "payroll" (accounting and payroll are not members of other groups). If no intersection existed in this case, "fred" would not be able to logon.

Security services does not enforce this type of security on devices other than terminals. Although an administrator may add printer and fax devices and control the membership of these devices, security services' enforcement ends at this point. The implementation of printer/fax security through the membership mechanism is enforced/configured through print services by querying the security service.

5.3.2 Terminal time usage restrictions

As discussed previously, time usage restrictions may be applied per system, group user or device. See section 4.1.1 and 4.1.2 for a detailed explanation.

5.3.3 Adding devices

A system administrator may add, delete or update the attributes of a device. In the case of terminals, these devices will be automatically created upon logon if the device does not exist in the security data base. The automatically created terminal device will inherit the attributes from the device object template (discussed later). If the terminal is a PC, the automatically created terminal object will be prefixed with the string "PC" followed by the network address (eg. PC@130.90.2.3). If the terminal device is an X-station or a dumb-terminal, then the object name will be prefixed with the string "WS###" followed by the network address (eg. WS001@130.90.2.3). Since X-stations all reside local to the security service, each X-station will be identified by the same name. Hence, in a pure X environment or dumb terminal environment, terminal security is supported in a limited fashion.

A system administrator may create a terminal device, but must create the terminal object name in the appropriate format. A terminal name consists of a string name (eg. "terminal1") plus a device address, either TCP/IP or XNS. A system administrator created name may then appear as "terminal1@135.0.21.2". This is referred to as the endpoint name. Each time that a user logs onto a system from "terminal1", the application software must inform the security service that the device is "terminal1". The security service will then determine the device's address to manufacture the complete endpoint name stored in the data base.

The previous descriptions of device names only apply to terminals (PCs, X-stations, etc.) and not to other types of devices, like printers. A printer device name is not subject to the naming convention restrictions imposed upon terminals. As well, printer device objects are not created automatically. A valid printer name may be "ricoh123". Device names, like user and group names, must be unique.

5.4 Reserved User/Group Objects

A set of reserved objects will accompany the security data base upon initialization. These objects provide special capabilities with respect to administrating the security service and the set of FileNet images.

- o SysAdmin is a reserved user and has unlimited read, write, and execute/append capabilities of both the security data base and the FileNet set of images.
- o SysAdminG is a reserved group which is a member of all groups.
- o AuditG is a reserved group which allows read-all access privileges to all of the documents on the system.
- o (NONE) is a reserved group which indicates that no-one has access to a particular object.
- o (ANYONE) is a reserved group which indicates that everyone has access to a particular object.
- o ServiceProcess is a reserved system object which is used to restrict inter-domain access.

5.4.1 SysAdmin user

When the security data base is first created, many default user, device and group objects are created. SysAdmin is the initial account with which a customer will manage the security data base. The SysAdmin user is given the highest privileges of all users and can access any document, modify any object or create any membership that it desires. This user derives its capabilities to modify the security data base by virtue of being the SysAdmin user. It is also able to access any document through its membership to the SysAdminG group (which is considered a

member of all groups).

This account should typically be used to develop the initial group hierarchies and objects within the security database. This is also the only account which is allowed to set a user's administrative attributes, update the system-wide defaults, update the object templates, add a function name and change an object's admin group (discussed later). The SysAdmin user is also not subject to restrictions imposed upon other users such as time usage restrictions or device security restrictions.

SysAdmin possesses all of the administrative attributes: supervisor, principal, group and password. These attributes are defined later in greater detail. It is generally expected that once a customer has set-up their system, they may not need the SysAdmin account for day-to-day administrative duties.

5.4.2 SysAdminG group

The SysAdminG group is a reserved group. By definition, SysAdminG is the member of all groups. By making an object a member of the SysAdminG group, that object will get read, write and execute/append access to all documents on the system. Membership in the SysAdminG group does not imply that that user has access to the security administrative functions. Modification of the security database is controlled through the administrative attributes.

5.4.3 AuditG group

The AuditG group is a reserved group. Any member of the AuditG group will get read-all access to all documents on the system.

5.4.4 (NONE)

(NONE) may be applied to access restrictions. This will indicate that no-one has the specified access to a document, folder, etc. The only users which may access an object with access restrictions of (NONE) are members of the SysAdminG group.

(NONE) may also be used as a user's primary and session group. This will effectively have no effect upon the user object, except in the instance in which application software sets access restrictions to the user's primary group.

5.4.5 (ANYONE)

(ANYONE) may also be applied to access restrictions. This will indicate that everyone has the specified access to a document, folder,

etc.

(ANYONE) may also be used as a user's primary group. This will effectively have no effect upon the user object, except in the instance in which application software sets the access restrictions to a user's primary group.

5.4.6 ServiceProcess

ServiceProcess is a reserved system object which is used to restrict inter-domain access. As well, ServiceProcess is the identity which FileNet servers may assume to acquire information not normally accessible by a normal user.

Inter-domain access may be secured through changing either the ServiceProcess name (the default is ServiceProcess:System:System) or by changing the ServiceProcess password. The only user which may change the ServiceProcess name and password is the SysAdmin user.

5.5 Administrative Attributes

The partitioning of administrative groups allows a customer to provide a level of security whereby no single system administrator has unchecked capabilities. An administrator may manage security objects which have an administrative group which lies within the transitive closure (extended membership) of the administrator's administrative group. The ability to perform a specific action is also bounded by an administrator's administrative attributes. An administrator may not create or modify another user object which has admin attributes set.

- o The administrative group stamp determines which objects may be modified and deleted by an administrator.
- o The Supervisor attribute grants permission to update/delete security objects.
- o The Principal attribute allows the creation of security objects, but not the update of those objects.
- o A Group administrator will only be able to make objects members of group objects.
- o A Password administrator may only change user's passwords.

5.5.1 Administrative group

An admin group is an identified group object. Whenever an administrator creates an object, that object will be stamped with the administrator's admin group. Subsequently, any administrator may modify this object only if the object's admin group lies within the transitive closure of the administrator's admin group. This relationship (transitive closure) is determined through the membership of the group object with which an administrator was labelled.

+++ SEC - Security Services +++

When a group is deleted, a check is performed to determine if a security object has the deleted group as its admin group. When a match is found, the admin group of the object is changed to (NONE). As a result, the act of deleting a group may not only affect access to documents, but an administrator's ability to access security objects. Once the admin group is set to (NONE), only the SysAdmin user may modify these objects. As well, only the SysAdmin user may change the admin group of an object.

5.5.2 Supervisor attribute

The supervisor administrative class can perform the following activities relative to the objects within the security database:

- o update security objects
- o delete security objects
- o logoff users

The set of objects which a supervisor may manage are users, groups and devices. The object which the supervisor is manipulating must have its admin group within the transitive closure of the supervisor's admin group. Through this restriction it is possible to create a system in which multiple system administrators exist, each system administrator manipulating only those objects within its hierarchy.

5.5.3 Principal attribute

An administrator with the principal attribute may:

- o add security objects
- o logoff users

When a principal creates a security object (user, group or device), that object will be stamped with the administrator's admin group. This indicates which administrators may manipulate this object, as previously described.

Although a principal may create a security object, it may not activate that object for use. An object is activated through setting the expiration time to the "no expiration" value or to a time beyond the current date/time. This is a function of the supervisor administrator. Since activation is considered to be an update process, a principal may not set this field and hence cannot make an object usable on the system.

A principal may not set a user's primary group, except in the instance in which the principal creates a group. This is harmless, since by definition, a group's primary group may only be itself. For user and device objects, it is not required that a primary group be provided.

If no primary group is specified for these objects, then the primary group will default to (NONE).

5.5.4 Group attribute

An administrator with the group attribute set may:

- o add members to groups
- o delete members from groups
- o add members to functions
- o delete members from functions

Limitations exist, however, relative to the membership process. A group administrator may add members to a group object if:

- o the group's expiration time is beyond the current time
- o the group's admin group is within the group administrator's admin group transitive closure

Although a group administrator may not make an object a member of an expired group, the group administrator may make an expired group a member of another active group. The advantage of this last rule allows a supervisor to review the transitive closure of the newly created object before anything may be made a member of it.

5.5.5 Password attribute

An administrator with the password attribute may add/change the password of any user whose admin group stamp intersects the transitive closure of the password administrator's admin group stamp. A password administrator may not change the passwords of any user which has any administrative attribute set. If a password administrator does not possess the supervisor attribute, then the target user's object should be inactivated, or expired, by the appropriate supervisor administrator before the password administrator may effect a password change.

5.6 Reporting Controls

Security services provides the capability to monitor various security related activities within a domain. The reporting may be set on a system, user, group or terminal basis. The security logs are kept on a daily basis for a period of 28 days. Once the 29th day arrives, the oldest security log file will be overwritten. The security log file is only accessible by the Unix fnsu user. These logs may be viewed through the security administration application.

- o Failed logons
- o Successful logons/logoffs
- o Changes to the security data base

5.6.1 Failed logons

+++ SEC - Security Services +++

A logon may fail for a variety of reasons. The set of errors which may occur consist of: a bad password, a non-existent user, an expired user/primary group/terminal object, an invalid logon time, too many user sessions, or too many concurrent sessions.

5.6.2 Successful logons/logoffs

When a logon succeeds, the action will be logged in the security log file depending upon whether the system, user, primary group or terminal has this logging attribute turned-on. As well, a corresponding successful logoff will be recorded.

5.6.3 Security database updates

Security data base updates relate to changes made to membership, functions, objects or other system attributes.

5.7 Function Controls

Security services only allows the SysAdmin administrator to save and delete function names within the security database. A group administrator may freely add members to functions, however. Each function name is representative of a task or menu option within an application. The appropriate function names and behavior may be identified through the user documentation. At this point, security services will not validate the function name. Whatever function name is input will be added to the security data base, verbatim.

Access to a function is validated by comparing the set of function members to the user's membership transitive closure. If an intersection exists, then function access is granted.

The behavior of functions may be controlled through the system level attribute, "no function definition ok". If this attribute is turned "on", then the lack of a function definition will not result in a restriction. However, if this attribute is turned "off", then a function definition is required in order to evaluate a membership intersection. In this case, if the function is not defined, then access is always denied.

6. SEC Tools

The set of SEC tools provided allows the system administrator and your service representative to view and manipulate the SEC data base using a command-line format similar to that used by MKF_tool and CSM_tool. In addition, conversion utilities will be provided so that the customer may convert from the old SKF format to the new MKF format. Depending

+++ SEC - Security Services +++

upon how the customer has configured their old SCT system, some degree of effort may be required to convert. The set of tools provided consist of:

1. SEC conversion tools
2. SEC debug tools
3. SEC data base initialization tools
4. SEC logon tool

6.1 SEC Conversion

SEC_convert will read the existing SKF data base and transfer the elements to the MKF data base format. One of the key changes in the new SEC relates to the fact that a distinction has been made between users and groups. SCT allowed a system administrator to make a user a member of another user. By doing this, the definition of ownership and accountability became blurred. The conversion routine will effectively identify all groups to which no other user belongs and assume that this SCT group is an SEC user. Likewise, those entities which have members will be forked into both a user and a group; the old entity id will be associated with the group.

SEC_convert will also provide reports to show the system administrator the entity relationships in SCT and how these will map to the new SEC data structures. This will allow the system administrator the opportunity to review the elements of the old and new database before making conversion decisions.

Since the user-group definition in SCT is not definitive, the only method of identifying whether an entity is a user or a group would be through membership. The resultant groups and users may be renamed later through the administration software.

In the event that an entity contains itself as its own primary group, the conversion process will change the primary group to (NONE). Once the conversion is complete, a report will be generated showing the results and problems which occurred. For example, changing the user/group name in conversion may exceed the allowable name length. Functions, terminals, and preferences will also be mapped to the new SEC format. Function tables will essentially remain the same, except for the class definition. Since the SEC preferences is a superset of the SCT preferences, these may be easily mapped along with SEC defaults to the new system preference list.

6.2 SEC tool

SEC_tool will provide debugging capabilities for your service representative. The interface will be command interactive and will follow the convention set-forth by PRI_tool (II), MKF_tool, and CSM_tool. Viewing the SEC database will not be performed through

+++ SEC - Security Services +++

SEC_tool, but through MKF_tool. SEC_tool requires that a user be logged-on as a FileNet user prior to execution. This may be accomplished through the use of the fnlogon program, discussed later. In addition, a user may use SEC_tool without using the fnlogon program only if that user is logged-on as the Unix "root" or "fnsw" user. This will effectively give the user the same capabilities as the SysAdmin user.

SEC_tool will provide the ability to:

- o create file output from an SEC_tool session
- o identify who is logged-on
- o identify "who i am"
- o logoff users
- o export/import the security data base to/from an ascii file
- o perform name-to-id and id-to-name conversion
- o encode and decode access restrictions
- o examine concurrent licensing restrictions
- o debug the security service

6.2.1 Hardcopy output

Through the use of the "hardcopy" command in SEC_tool, it is possible to redirect all subsequent output to a named file. This output would represent all information which is displayed from SEC_tool until the hardcopy option is turned off.

6.2.2 Who

SEC_tool will return a list of users which the security service believes are currently logged-on. This list will also include other information regarding each logon session.

6.2.3 Whoami

The whoami command will provide basic information about the current active security session.

6.2.4 Logoff

This command will allow a user to logoff their own sessions. This is also useful if a security domain has reached its concurrent license limit. A system administrator who is logged-on as the UNIX "root" user or as the "fnsw" user may execute SEC_tool without a prior logon. The system administrator may then terminate any security session.

6.2.5 Export/Import

The export command allows a customer to export, or copy, the security data base into a designated file. The security data base will be

+++ SEC - Security Services +++

copied in an ascii, human readable format. The export file may then be used to replicate the security service across multiple domains through the use of the import command.

The import command allows a customer to import an "exported" security database. The import command allows a user to alter the domain names of all imported objects and to completely overwrite reserved objects such as "SysAdmin" and "AuditG".

In order for the export and import command to function, the executing user must be logged-on as the "SysAdmin" user, otherwise an error will be returned.

6.2.6 Name-to-id/id-to-name

Using this option, a user may query the SEC data base requesting that a given name be translated to an id, or that a given id be translated to an object name.

6.2.7 Encode/decode

Most SEC clients will store security access restrictions in an encoded (compressed) format in order to conserve memory and disk space. These encoded access restrictions are not in a human understandable format. As a result, support personnel may need to identify actual id's associated with access restrictions. The decode command will take an encoded access restriction and provide the ids associated with the read/write/execute-append permissions. The encode command will take object ids and translate these into an encoded format. These commands are strictly used for debugging purposes.

6.2.8 License

The license command will display basic information related to concurrent licensing. This information consists of: the number of users currently logged-on; the concurrent license limit; the number of logon rejects due to the concurrent license limit; the most number of users ever logged-on.

6.3 SEC data base initialization

The security service data base is automatically initialized upon system startup. This will occur if security services finds "all" of its security tables empty. Additionally, the SEC_init tool is provided so that a customer may manually initialize the security data base. This initialization tool will add the default objects and membership relationships in addition to the system wide defaults. The only object which will be given a usable password during initialization is the

+++ SEC - Security Services +++

"SysAdmin" user object. Its password will default to "SysAdmin". The set of default objects are:

- o SysAdminG
- o FieldServiceG
- o OperatorG
- o ServiceProcess
- o UserDefaults
- o GroupDefaults
- o DeviceDefaults
- o SysAdmin
- o FieldService
- o Operator
- o ServProcTerm@0.0.0.0
- o AuditG

This tool provides two options which will modify the security data base. The first option will search the security data base for each of the aforementioned reserved security objects. If the object does not exist, it will add that object to the data base. If the object does exist the tool will skip that object and proceed to the next reserved object.

The second option must be run as the Unix "root" user. This option overwrites each of the aforementioned security objects with the initial, hardcoded default values. This option is secured by "root" because the password of the SysAdmin user will revert to its initial value, "SysAdmin".

6.4 SEC Logon tool

The logon tool, called "fnlogon", allows a user to logon to the FileNet security service from the Unix shell. The user is prompted for a FileNet logon name, a corresponding FileNet logon password, the name of the security service and the name of an application to launch, if desired. If no security service name is specified, then fnlogon will connect the user to the local domain's security service. It is generally recommended that no service name be specified. This is because subsequent applications which a user may launch may not be designed for remote operation outside the local domain. Finally, the user may specify an application to launch or press carriage return. By pressing the return-key, the user's default shell will be launched as identified through their Unix logon in the /etc/passwd file. If the user types "exit" at this point, then their launched shell will be terminated and they will be automatically logged-off from the security service. From this shell, the user may start any FileNet application. If the FileNet application requires that the user be logged-on, then that application will automatically inherit the the logon information from the recently executed fnlogon program. This program will allow a user to by-pass

+++ SEC - Security Services +++

the use of Xapex and go directly to the desired application.

In addition, many FileNet tools require that an fnlogon be performed in order to execute. These tools may generally be identified through the "XXX_tool" naming convention.

7. Appendix B: Glossary of Terms

ACCESS RESTRICTIONS - An access restriction indicates if a security object may read, write, execute/append a document or data. The current Image Services method of employing access restrictions associates an object id with each of these three categories. Access is determined through comparing a requestor's transitive closure to the id associated with the access requested type (ie. read, write, execute/append).

ADMINISTRATIVE GROUP - An administrative group may be any group on the system. When a system administrator creates a security object, that security object will inherit the administrative group of the creator. Subsequently, only administrators who have that administrative group within the transitive closure of their administrative group may modify this newly created object. Administrative groups may only be changed by the SysAdmin user.

CACHING - Caching indicates that information is being stored in an area for fast retrieval. Information may be cached to hard-disk or memory. Security services will cache its information into shared memory to avoid repeated calls to the data base.

CLIENT - A client may be any process making a request of a service (eg. PC software, Doc Services).

CONCURRENT LICENSE - A concurrent license for logon will restrict the number of simultaneous FileNet logons at a given point in time.

control - A control is a method for restricting the behavior of software and individuals. Controls may offer a series of checks and balances so that a given environment may be regulated.

decrypt(ion) - This is a method used to take an encrypted message and create a plaintext, human readable message.

DES - This the the Data Encryption Standard algorithm. This is a published method of encrypting information and is required by many Federal Government agencies.

DEVICE - This is a physical hardware unit which exists within the domain of the security service. This equipment may consist of terminals or printers.

DOS - This is the operating system which is resident on some of the

+++ SEC - Security Services +++

Personal Computer clients used by the FileNet applications.

DUMB-TERMINAL - A dumb-terminal is a device which does not possess any processing capacity of its own. Dumb-terminals merely relay information provided by the applications which run on the central processor. Without the central processor, dumb-terminals are useless. An example of a "non" dumb-terminal would be a PC.

ENCRYPT(ION) - This is a method of putting messages into a secret form using words or symbols to represent the plaintext message. A "key" is typically used to encrypt and decrypt the message.
endpoint address - An endpoint address represents the network address of a device object in the new security services. The endpoint address may be either an ethernet or a TCP/IP formatted address. Relative to security services, the only object for which an endpoint address is relevant is a terminal device.

ENTITY - This term is used to refer to SCT users/groups since a distinction is not made in the old security services.
extended membership - See transitive closure.

GROUP - This is a security object which represents a collection of users, devices and other groups.

MKF - Multi-Keyed File (MKF) is the relational data base in which SEC will store its security tables. The security tables are placed in the permanent data base.

OBJECT - This is a security user, group or device.
password - A password is a secret set of characters which is used to identify a client and permit access to a system or service.
plaintext - A plaintext message is a human readable message. A plaintext (or cleartext) may be converted into a secret message, or cryptogram, by encrypting it.

PRIMARY GROUP - A primary group may be any group on the system. Declaring a group as an object's primary group causes that object to inherit some of the attributes of that group. This inheritance is applicable to logon restrictions and security logging.
proprietary algorithm - This indicates that a series of operations within the FileNet set of software are strictly inherent to that environment. These algorithms will remain a non-disclosed asset.

SCT - This is the original subsystem which provided the security service for FDOS and Unix servers. SCT indirectly provided a security service for PC clients. This subsystem also stored security information within an SKF (Single-Keyed File) data base; more specifically, the ERM_cust file.

+++ SEC - Security Services +++

SEC (old) - This was a conglomeration of remote security calls designed to communicate with a PC client. The old SEC server is layered on top of SCT.

SEC (new) - The new security services is identified with this acronym. This replaces both SCT and the old SEC.

SERVER/SERVICE - A server is the logical set of software which may provide information, or functionality for a requesting set of software (the client). For example, a security server may restrict access and provide user information; a print server will provide the capability to print and review printing items.

SESSION - This indicates (relative to this document) a single logon occurrence. If a user has multiple sessions, this implies that the user is logged-on multiple times.

SKF - This is a Single-Keyed File data base which is used by SCT and the old SEC for the storage of security information.

STUBS - A stub is that set of software which provides the client server communication capability. Typically, FileNet stubs may be identified through the naming convention XXXr and XXXs, where the "r" suffix represents the client code and the "s" suffix represents the server code.

SYSADMIN - This corresponds to the UNIX "root." system administrator - This is the individual at the customer site responsible for implementing and managing FileNet security.

TRANSITIVE CLOSURE - The transitive closure of an object represents all groups of which an object is a member in addition to the groups of which the groups are a member, and so forth. For example, user "A" is a direct member of group "B". Group "B" is a direct member of groups "C" and "D". Group "C" is a direct member of group "X". Therefore, the transitive closure of user "A's" membership is "B, C, D, X". In the event that a document has its read permission set to group "D", then user "A" will be granted read capability to that document by virtue of that user's transitive closure. See access restrictions.

UNIX - UNIX is the operating system on which most of the FileNet software products exist. There are many variants of the Unix operating system (eg. Hewlette-Packard UX (HPUX), IBM(R) AIX(R) and AT&T System V).

user - A user is a unit that has no members and can initiate a security session through logon.

WAL - In the context of this document, WAL (Workflo Application Libraries) represents a programmatic interface which allows access to

+++ SEC - Security Services +++

security data and functionality.

WILD CARD - A wild card is a value which may represent a complete range of values within the definition of a given environment. Security services uses wild card values on time restrictions. This indicates that any time value is valid.

X-STATION - An X-station is a terminal device which uses the "X" software user interface for the display of windows and other graphic/text oriented information. For this release, many of the administrative software applications operate in an X environment, including the security administration application.

```
/*-----*\
*****\
* File:  SEC.defs
*****
\*****/
```

```
#ifndef SEC_defs
```

```
#define SEC_defs
```

```
#include <FileNet.h>
```

```
#include <AS_externals.h>
```

```
/*-----*\
| Used for setting what is to be debugged                               |
\*-----*/
```

```
typedef unsigned long SEC_debug_options_typ;
```

```
/*-----*\
| Lengths of items contained in a field.  These length values apply to
| the actual string length of a value and not to the actual size of the
| container.
\*-----*/
```

```
#define SEC_EPWD_LENGTH          13
#define SEC_PWD_LENGTH           8
#define SEC_COMMENTS_LENGTH     79
#define SEC_ORGDOM_LENGTH       42
#define SEC_DICT_NAME_LENGTH    80
#define SEC_ORG_LENGTH          20
#define SEC_DOM_LENGTH          20
#define SEC_OBJ_LENGTH          40
#define SEC_FUN_LENGTH          80
#define SEC_FULL_OBJECT_LENGTH  83
#define SEC_TERM_LENGTH         16
#define SEC_UNIXPWD_LENGTH      13
#define SEC_ADDR_LENGTH         32
```


+++ SEC - Security Services +++

```
#define SEC_LANGUAGE_LENGTH          16
#define SEC_SCTPASSWORD_LENGTH      16
#define SEC_DBNAME_LENGTH           40
#define SEC_EDBPWD_LENGTH           64
#define SEC_DBPWD_LENGTH            40
#define SEC_maxnamelength           84
#define SEC_maxterminallength       17
#define SEC_maxpasswordlength       9
#define SEC_DB2_ACCT_NAME_LENGTH    79
#define SEC_EDB2PWD_LENGTH          64
#define SEC_DN_LENGTH               1023
/* SEC_CN_LENGTH was added for ER 749845 - Extensible Authentication */
#define SEC_CN_LENGTH                39

/*****
* STR 52183
*****/
#define SEC_ACTIVATED                0x0001
#define SEC_GROUP_OBJECT             0x0002

/*-----*\
| String definitions; all must be null terminated.
\*-----*/
typedef char SEC_password_typ [ SEC_PWD_LENGTH + 1 ];
typedef char SEC_name_typ [SEC_FULL_OBJECT_LENGTH + 1];
typedef char SEC_terminal_typ [ SEC_TERM_LENGTH + 1 ];
typedef char SEC_endpt_addr_typ [ SEC_ADDR_LENGTH + 1 ];
typedef char SEC_function_name_typ[SEC_FUN_LENGTH];
typedef char SEC_comments_typ [SEC_COMMENTS_LENGTH+1];
typedef char SEC_despwd_typ [ SEC_UNIXPWD_LENGTH + 1 ];
typedef char SEC_oldpassword_typ [ SEC_SCTPASSWORD_LENGTH ];
typedef char SEC_dbname_typ [ SEC_DBNAME_LENGTH + 1 ];
typedef char SEC_edbpwd_typ [ SEC_EDBPWD_LENGTH ];
typedef char SEC_dbpwd_typ [ SEC_DBPWD_LENGTH ];
typedef char SEC_db2_acct_typ [ SEC_DB2_ACCT_NAME_LENGTH + 1 ];
typedef char SEC_eDB2PwD_typ [ SEC_EDB2PWD_LENGTH ];
typedef char SEC_dn_typ [SEC_DN_LENGTH + 1]; /* Not available in ISTK */
typedef char SEC_cn_typ [SEC_CN_LENGTH + 1]; /* Not available in ISTK */

#define SEC_DB2_PRIMARY_PWD          0
#define SEC_DB2_SECONDARY_PWD        1

#define SEC_DB2_MOVE_SCND_PWD        0
#define SEC_DB2_CLEAR_SCND_PWD       1

#define SEC_DB2_MAGIC_NUM             0xFACE9731
#define SEC_DB2_PWD_MAX_LENGTH        32
#define SEC_DB2_ACCT_NAME_MAX_LENGTH  64
```

+++ SEC - Security Services +++

```
/*-----*\
| The ids for these languages apply to old FDOS related languages,
| and does not apply to any other platform.
/*-----*/
typedef unsigned char          SEC_language_id_typ;

#define SEC_ENGLISH_ID        49
#define SEC_GERMAN_ID         SEC_ENGLISH_ID+1
#define SEC_FRENCH_ID         SEC_ENGLISH_ID+2
#define SEC_ITALIAN_ID        SEC_ENGLISH_ID+3
#define SEC_SPANISH_ID        SEC_ENGLISH_ID+4
#define SEC_BRITISH_ID        SEC_ENGLISH_ID+5
#define SEC_PORTUGUESE_ID     SEC_ENGLISH_ID+6
#define SEC_DANISH_ID         SEC_ENGLISH_ID+7
#define SEC_NORWEGIAN_ID      SEC_ENGLISH_ID+8
#define SEC_SWEDISH_ID        SEC_ENGLISH_ID+9
#define SEC_SWISSGERMAN_ID    SEC_ENGLISH_ID+10
#define SEC_SWISSROMANCE_ID   SEC_ENGLISH_ID+11
#define SEC_DUTCH_ID          SEC_ENGLISH_ID+12
#define SEC_ARABIC_ID         SEC_ENGLISH_ID+13

/*-----*\
| This define exists only for the sake of nomenclature and 3.0.1
| WAL compatibility.
/*-----*/
#define SEC_get_language      SEC_get_language_name

/*-----*\
| These language names only apply to FDOS servers. For all other
| environments, a POSIX language string will be used.
/*-----*/
typedef char                  SEC_language_typ [SEC_LANGUAGE_LENGTH + 1];

#define SEC_ENGLISH_NAME     "English"
#define SEC_GERMAN_NAME      "German"
#define SEC_FRENCH_NAME      "French"
#define SEC_ITALIAN_NAME     "Italian"
#define SEC_SPANISH_NAME     "Spanish"
#define SEC_BRITISH_NAME     "British"
#define SEC_PORTUGUESE_NAME  "Portuguese"
#define SEC_DANISH_NAME      "Danish"
#define SEC_NORWEGIAN_NAME   "Norwegian"
#define SEC_SWEDISH_NAME     "Swedish"
#define SEC_SWISSGERMAN_NAME  "SwissGerman"
#define SEC_SWISSROMANCE_NAME "SwissRomance"
#define SEC_DUTCH_NAME       "Dutch"
#define SEC_ARABIC_NAME      "Arabic"
```

+++ SEC - Security Services +++

```
#define SEC_MAX_LANGUAGES      14

#define SEC_CONVERTED          "Conversion xref_id=" /* put in comments field*/

/*-----*\
| The time range structure represents from/to times at which an object
| may perform operations through the security service such as logging-
| on. As well, each time that a client makes a request of the
| security service, a check will be performed to determine if the
| client is operating within the accepted time range.
\*-----*/
#define SEC_WILDCARD_TIME      0xB00D

typedef struct SEC_time_range_typ
{
    long    start_min;        /* min=0, max=59 */
    long    start_hour;      /* min=0, max=23 */
    long    start_dweek;     /* min=0, max=6 **/* 0=Sunday */
    long    end_min;         /* min=0, max=59 */
    long    end_hour;        /* min=0, max=23 */
    long    end_dweek;       /* min=0; max=6 **/* 0=Sunday */
} SEC_time_range_typ;

/*-----*\
| This is a set of predefined id's. These are initially placed in the
| security data base by the SEC_init program or SEC_convert program.
| Some of these id's are carry-overs from previous FileNet security
| software and do not necessarily have special meaning in the new
| security services.
\*-----*/
typedef unsigned long          SEC_id_typ;

/*-----*\
| These ids are not found in the security data base. Some of the
| security services entry-points will accept these as valid ids ( eg.
| SEC_id_to_name() ). These may be found on access restrictions of
| documents, queues or other objects.
\*-----*/
#define SEC_NULL_GROUP_ID      0 /* can't do anything */
#define SEC_ANYONE_ID          1 /* anyone can access */

/*-----*\
| Any member of the SEC_SYSADMIN_GRP_ID will automatically get access
| too all documents irregardless of the access restrictions placed on
| the document. Making a user a member of this group should be used
| prudently due to the security risk.
\*-----*/
#define SEC_SYSADMIN_GRP_ID    2 /* mbrs get access to all docs */
```

+++ SEC - Security Services +++

```
/*-----*\
| These group-ids exist due to compatibility with older FileNet
| security services.
/*-----*/
#define SEC_FIELD_SVC_GRP_ID      3 /* left over from sct */
#define SEC_OPERATOR_GRP_ID      4 /* left over from sct */
/*-----*\
| The service process is used by the FileNet set of services and its
| use is generally internal. This object may be used to restrict
| access between different security domains. This may be accomplished
| by changing the service process password to a password other than
| that used by other domains' service process objects.
/*-----*/
#define SEC_SERVPROC_ID          5 /* local service id, internal */
/*-----*\
| This is an internal id and represents an unknown object.
/*-----*/
#define SEC_UNDEFINED_ID         6 /* unknown, internal */
/*-----*\
| These ids represent object templates. When objects are
| created, the object will be populated with the settings contained
| in this template unless otherwise specified.
/*-----*/
#define SEC_USER_DEFAULTS_ID     7 /* user object defaults id */
#define SEC_GROUP_DEFAULTS_ID   8 /* group object defaults id */
#define SEC_DEVICE_DEFAULTS_ID  9 /* device object defaults id */
/*-----*\
| This id is used as a key to access the system structure which contains
| various attributes for the specific security domain.
/*-----*/
#define SEC_SYSTEM_DEFAULTS_ID  10 /* system object defaults id */
/*-----*\
| The SysAdmin user is the equivalent of the UNIX root user. Not only
| does note SysAdmin user get complete access to all documents, but
| this user may modify the security database without restrictions.
/*-----*/
#define SEC_SYSADMIN_USR_ID      11 /* initial account */
/*-----*\
| These ids are carry-overs from older security services.
/*-----*/
#define SEC_FIELD_SVC_USR_ID     12 /* carry over */
#define SEC_OPERATOR_USR_ID     13 /* carry over */
#define SEC_SERVPROC_TERM_ID    14 /* service proc terminal */
/*-----*\
| Any member of the audit group will get read-all capabilities.
/*-----*/
#define SEC_AUDIT_GRP_ID        15 /* audit group gets readall */
#define SEC_LAST_RESERVED_ID   19
```

+++ SEC - Security Services +++

```

/*-----*\
| This set of characters is illegal to use for object names.
\*-----*/
#define SEC_BAD_CHARS          "'#[ ]!<>?* "
#define SEC_BAD_TERM_CHARS    "'#[ ]!<>?*: "
#define SEC_UNIFIED_LOGON_BAD_CHARS      ""

/*-----*\
| Predefined names.
\*-----*/
#define SEC_NULL_GROUP_NAME    "(NONE)"          /* not in db */
#define SEC_ANYONE_NAME        "(ANYONE)"        /* not in db */
#define SEC_SYSADMIN_GRP_NAME  "SysAdmin"        /* group name */
#define SEC_FIELD_SVC_GRP_NAME "FieldServiceG"   /* group name */
#define SEC_OPERATOR_GRP_NAME  "OperatorG"       /* group name */
#define SEC_SERVPROC_NAME      "ServiceProcess:System:System"
#define SEC_USER_DEFAULTS_NAME "UserDefaults"    /* sys name */
#define SEC_UNDEFINED_NAME     "(UNDEFINED)"     /* user name */
#define SEC_GROUP_DEFAULTS_NAME "GroupDefaults"  /* not in db */
#define SEC_DEVICE_DEFAULTS_NAME "DeviceDefaults" /* user name */
#define SEC_SYSTEM_DEFAULTS_NAME "SystemDefaults" /* user name */
#define SEC_SYSADMIN_USR_NAME   "SysAdmin"        /* sys name */
#define SEC_FIELD_SVC_USR_NAME  "FieldService"    /* user name */
#define SEC_OPERATOR_USR_NAME   "Operator"        /* user name */
#define SEC_SERVPROC_TERM_NAME  "ServProcTerm@0.0.0.0" /* device name */
#define SEC_AUDIT_GRP_NAME      "AuditG"          /* user name */

/*-----*\
| Object Classes
\*-----*/
typedef unsigned char          SEC_object_class_typ;

#define SEC_OBJCLASS_SYSTEM    1 /* generally internal to SEC */
#define SEC_OBJCLASS_GROUP     2 /* a group is something that has mbrs*/
#define SEC_OBJCLASS_USER      3 /* a user may logon */
#define SEC_OBJCLASS_DEVICE    4 /* a device is similar to a user but
                                may not logon */

#define SEC_MIN_OBJ_CLASS      SEC_OBJCLASS_SYSTEM
#define SEC_MAX_OBJ_CLASS      SEC_OBJCLASS_DEVICE

/*-----*\
| Device Classes
\*-----*/
typedef unsigned char          SEC_device_class_typ;

```

+++ SEC - Security Services +++

```
#define SEC_DEVCLASS_UNKNOWN 0 /* not a device */
#define SEC_DEVCLASS_TERMINAL 1 /* something from which a logon occurs*/
#define SEC_DEVCLASS_PRINTER 2 /* printer */
#define SEC_DEVCLASS_SCANNER 3 /* scanner, not yet implemented */
#define SEC_DEVCLASS_TAPE 4 /* tape drive, not yet implemented */
#define SEC_DEVCLASS_FAX 5 /* fax */
```

```
#define SEC_MIN_DEVICE_CLASS SEC_DEVCLASS_UNKNOWN
#define SEC_MAX_DEVICE_CLASS SEC_DEVCLASS_FAX
```

```
/*-----*\
| These conditional operators are used with the various entry-points
| which search the security services data base.
\*-----*/
```

```
typedef unsigned short SEC_conditional_typ;
```

```
#define SEC_EQL EQL
#define SEC_GEQ GEQ
#define SEC_GTR GTR
```

```
/*-----*\
| This set of access restriction types are used in the SEC_check_access()
| entry-point. FileNet access restrictions are read, write, execute/append.
\*-----*/
```

```
typedef unsigned short SEC_access_wanted_typ;
```

```
#define SEC_WANT_READ 1
#define SEC_WANT_WRITE 2
#define SEC_WANT_AX 4
```

```
/*-----*\
| Object Options Defines
\*-----*/
```

```
typedef short SEC_optobj_typ;
```

```
/* f = find, a = add, u = update */
/* u = user, g = group, d = dev */
```

```
/* action: object */
/* ----- */
#define SEC_OPTOBJ_OBJECT_NAME 1 /* f,a,u : u,g,d */
#define SEC_OPTOBJ_OBJECT_CLASS 2 /* f,a : u,g,d */
#define SEC_OPTOBJ_DEVICE_CLASS 3 /* f,a : d */
#define SEC_OPTOBJ_ADMIN_CLASS_SUPERV 4 /* f,a,u : u */
#define SEC_OPTOBJ_ADMIN_CLASS_PRINC 5 /* f,a,u : u */
#define SEC_OPTOBJ_ADMIN_CLASS_GROUP 6 /* f,a,u : u */
#define SEC_OPTOBJ_ADMIN_CLASS_PASSWORD 7 /* f,a,u : u */
#define SEC_OPTOBJ_ADMIN_CLASS_NOT 8 /* f : u */
```

+++ SEC - Security Services +++

```

#define SEC_OPTOBJ_DEVICE_SECURITY      9  /* f,a,u : u,g,d      */
#define SEC_OPTOBJ_LANGUAGE             10 /* f,a,u : u          */
#define SEC_OPTOBJ_PRIM_GROUP           11 /* f,a,u : u          */
#define SEC_OPTOBJ_ADMIN_GROUP          12 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_SESS_GROUP           13
#define SEC_OPTOBJ_START_MIN            14 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_START_HOUR           15 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_START_DWEEK          16 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_END_MIN              17 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_END_HOUR             18 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_END_DWEEK            19 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_EXPIRATION_TIME      20 /* a,u   : u,g,d     */
#define SEC_OPTOBJ_SUCCESS_WHERE        21 /* f     : u,g,d     */
#define SEC_OPTOBJ_FAILED_WHERE         22 /* f     : u,g,d     */
#define SEC_OPTOBJ_FAILED_ERR           23 /* f     : u,g,d     */
#define SEC_OPTOBJ_LOG_SUCCESS_LOGON    24 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_LOG_FAILED_LOGON     25 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_LOG_DB_UPDATES        26 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_SYSTEM_OVERRIDE       27 /* f,a,u : u,g,d     */
#define SEC_OPTOBJ_MAX_SESSIONS         28 /* f,a,u : u          */
#define SEC_OPTOBJ_COMMENTS             29 /* a,u   : u,g,d     */
#define SEC_OPTOBJ_PASSWORD             30 /* a,u   : u          */
#define SEC_OPTOBJ_DEFAULT_DBMAP_ID     31 /* a,u,f : u          */
#define SEC_OPTOBJ_CLASS_KEY            32 /* f */
#define SEC_MAXOBJ_OPTIONS              SEC_OPTOBJ_CLASS_KEY

```

```

/*-----*\
| An object may be inactivated by setting its expiration_time to a time
| value less than the current time.  If the expiration time is 0, then
| no expiration time exists for the object.  This applies to all
| security objects.
\*-----*/

```

```

#define SEC_NO_EXPIRATION                0
#define SEC_EXPIRED                      1

```

```

/*-----*\
| When an object is first added, its password expiration time is set
| to 0.  After the first logon, the password expiration will
| be changed to SEC_PWD_EXPIRED.  The next time that a user logs-on
| without changing their password, access will be denied.
|
| If the system level mandatory password change flag is set then the
| expiration time is set to SEC_PWD_RESET and the user will be required
| to change their password on next logon.
\*-----*/

```

```

#define SEC_PWD_EXPIRED                  1
#define SEC_PWD_RESET                    2

```

+++ SEC - Security Services +++

```

/*-----*\
| This is the union of options which may be specified by an SEC client
| for adding, searching or updating objects. Each pointer listed in
| the union must have its memory allocated by the client. The field
| types which each of these pointers represents is correlated below.
*-----*/
typedef struct SEC_object_opt_typ
{
    SEC_optobj_typ      option_type;      /* option for update */      (pg376.34)
    union
    {
        char            *object_name_p;    /* SEC_name_typ size */
        SEC_object_class_typ object_class;    (pg375.33)
        SEC_device_class_typ device_class;    (pg375.47)
        bool            admin_class_supervisor;    (pg491.22)
        bool            admin_class_principal;    (pg491.22)
        bool            admin_class_group;    (pg491.22)
        bool            admin_class_password;    (pg491.22)
        bool            admin_class_not;    (pg491.22)
        bool            device_security;    (pg491.22)
        char            *language_p;        /* SEC_language_typ size */
        SEC_id_typ      prim_group;    (pg373.32)
        SEC_id_typ      admin_group;    (pg373.32)
        SEC_id_typ      sess_group;    (pg373.32)
        long            start_min;
        long            start_hour;
        long            start_dweek;
        long            end_min;
        long            end_hour;
        long            end_dweek;
        ASE_time_typ    expiration_time;    (pg489.31)
        SEC_id_typ      success_where;    (pg373.32)
        SEC_id_typ      failed_where;    (pg373.32)
        error_typ      failed_err;    (pg493.26)
        bool            log_success_logon;    (pg491.22)
        bool            log_failed_logon;    (pg491.22)
        bool            log_db_updates;    (pg491.22)
        bool            system_override;    (pg491.22)
        unsigned short  max_sessions;
        char            *comments_p;        /* SEC_comments_typ size */
        char            *password_p;        /* SEC_password_typ size */
        SEC_id_typ      default_dbmap_id;    (pg373.32)
    } opt;
} SEC_object_opt_typ;

/*-----*\
| This structure is representative of all four types of security objects;

```


+++ SEC - Security Services +++

```

| system, group user and device.
/*-----*/
typedef struct SEC_object_desc_typ
{
    SEC_name_typ      object_name;      /* up to a three part name      */ (pg371.25)
    SEC_id_typ        object_id;        /* number identifying object    */ (pg373.32)
    unsigned char     object_class;     /* user, group, device         */
    SEC_device_class_typ device_class; /* printer, terminal, etc.     */ (pg375.47)
    bool              admin_class_supervisor; /* may update accounts      */ (pg491.22)
    bool              admin_class_principal; /* may create accounts       */ (pg491.22)
    bool              admin_class_group; /* may manage grp mbrs       */ (pg491.22)
    bool              admin_class_password; /* can change passwords      */ (pg491.22)
    bool              admin_class_not; /* no administrative caps     */ (pg491.22)
    bool              device_security; /* 1=on, 0= off               */ (pg491.22)
    SEC_language_typ language;          /* only applies to FDOS systems */ (pg372.33)
    SEC_id_typ        prim_group;       /* primary object group       */ (pg373.32)
    SEC_id_typ        admin_group;      /* admin group which created obj */ (pg373.32)
    SEC_id_typ        sess_group;       /* sessn grp ctling obj       */ (pg373.32)
    SEC_time_range_typ logon_times;     /* allowable logon times      */ (pg373.23)
    ASE_time_typ      expiration_time; /* UNIX time for account death */ (pg489.31)
    ASE_time_typ      creation_time; /* time created                */ (pg489.31)
    ASE_time_typ      success_log_time; /* last successful logon time  */ (pg489.31)
    SEC_id_typ        success_where; /* where successful logon      */ (pg373.32)
    ASE_time_typ      failed_log_time; /* last failed logon time     */ (pg489.31)
    SEC_id_typ        failed_where; /* last failed where          */ (pg373.32)
    error_typ         failed_err; /* failed where error tuple   */ (pg493.26)
    bool              log_success_logon; /* log a successful logon?    */ (pg491.22)
    bool              log_failed_logon; /* log a failed logon?       */ (pg491.22)
    bool              log_db_updates; /* log an SEC db update?     */ (pg491.22)
    bool              system_override; /* 0=use system, 1= use user  */ (pg491.22)
    unsigned short    max_sessions; /* max session for the user per sys */
    SEC_comments_typ comments; /* user description          */ (pg371.29)
    unsigned long     nbr_logons; /* total nbr times logged in  */
    ASE_ssn_typ      ssn; /* system ssn                */ (pg483.47)
    unsigned short    nbr_members; /* number of members in array */
    SEC_id_typ        *mbr_p; /* member of grps id list    */ (pg373.32)
    SEC_id_typ        default_dbmap_id; /* db user to map to        */ (pg373.32)
} SEC_object_desc_typ;

/*-----*\
| This represents a linked list of data base objects which may be returned
| by the SEC_find_object() entry-point.
/*-----*/
typedef struct SEC_object_list_typ
{
    struct SEC_object_list_typ *next_p; (pg379.48)
    SEC_object_desc_typ obj; (pg379.38)
} SEC_object_list_typ;

```

+++ SEC - Security Services +++

```
/*-----*\
| Object Option Range Limits
\*-----*/
#define SEC_MIN_MINUTES          0
#define SEC_MAX_MINUTES          59
#define SEC_MIN_HOURS            0
#define SEC_MAX_HOURS            23
#define SEC_MIN_DWEEK            0
#define SEC_MAX_DWEEK            6
#define SEC_MIN_SYS_OVERRIDE     0
#define SEC_MAX_SYS_OVERRIDE     1
#define SEC_MIN_MAX_SESSIONS     1
#define SEC_MAX_MAX_SESSIONS     10000
#define SEC_MIN_PWD_MIN_LENGTH   0
#define SEC_MAX_PWD_MIN_LENGTH   8
#define SEC_MIN_PWD_ATTEMPTS     0
#define SEC_MAX_PWD_ATTEMPTS     256
#define SEC_MIN_PWD_RENEWAL_DAYS 0
#define SEC_MAX_PWD_RENEWAL_DAYS 365
#define SEC_MIN_PWD_GRACE_PERIOD 0
#define SEC_MAX_PWD_GRACE_PERIOD 90
#define SEC_MIN_PWD_FAILURE_MINS 0
#define SEC_MAX_PWD_FAILURE_MINS 100
#define SEC_MIN_FUNCLASS         1
#define SEC_MAX_FUNCLASS         1000
#define SEC_MAX_PWD_LENGTH       8
#define SEC_MAX_TITLE_LENGTH     49
#define SEC_MAX_FORM_LENGTH      9
#define SEC_MAX_TMASK_LENGTH     49
#define SEC_MAX_DMASK_LENGTH     49
#define SEC_MAX_NMASK_LENGTH     19
#define SEC_MAX_MMASK_LENGTH     9
#define SEC_MAX_LANGUAGE_LENGTH  17
```

```
/*-----*\
| System Options Defines
\*-----*/
typedef short          SEC_optsys_typ;

#define SEC_OPTSYS_SYSTEM_ID      1
#define SEC_OPTSYS_DEVICE_SECURITY 2
#define SEC_OPTSYS_NO_FUNC_DEF_OK 3
#define SEC_OPTSYS_UPDATE_TIME   4
#define SEC_OPTSYS_START_MIN     5
#define SEC_OPTSYS_START_HOUR    6
#define SEC_OPTSYS_START_DWEEK   7
#define SEC_OPTSYS_END_MIN       8
```

+++ SEC - Security Services +++

```

#define SEC_OPTSYS_END_HOUR          9
#define SEC_OPTSYS_END_DWEEK        10
#define SEC_OPTSYS_MAX_SESSIONS     11
#define SEC_OPTSYS_PWD_SPEC_CHAR    12
#define SEC_OPTSYS_PWD_MIN_LEN      13
#define SEC_OPTSYS_PWD_RENEWAL_DAYS 14
#define SEC_OPTSYS_PWD_GRACE_PERIOD 15
#define SEC_OPTSYS_PWD_ATTEMPTS     16
#define SEC_OPTSYS_PWD_FAILURE_MINS 17
#define SEC_OPTSYS_LOG_SUCCESS_LOGON 18
#define SEC_OPTSYS_LOG_FAILED_LOGON 19
#define SEC_OPTSYS_LOG_DB_UPDATES   20
#define SEC_OPTSYS_LANGUAGE         21
#define SEC_OPTSYS_ALLOW_OVERRIDE   22
#define SEC_MAXSYS_OPTIONS           SEC_OPTSYS_ALLOW_OVERRIDE

/*-----*\
| The system options blanket the behavior of all security objects in
| a given security domain.
\*-----*/
typedef struct SEC_system_opt_typ
{
    SEC_optsys_typ      option_type;                (pg380.39)
    union
    {
        SEC_id_typ      system_id;                  /* SEC_SYSTEM_DEFAULTS_ID */ (pg373.32)
        bool            device_security;            (pg491.22)
        bool            no_func_def_ok;            (pg491.22)
        ASE_time_typ    update_time;               (pg489.31)
        long            start_min;
        long            start_hour;
        long            start_dweek;
        long            end_min;
        long            end_hour;
        long            end_dweek;
        unsigned short  max_sessions;
        unsigned short  pwd_special_char;
        unsigned short  pwd_min_len;
        unsigned short  pwd_renewal_days;
        unsigned short  pwd_grace_period;
        unsigned short  pwd_attempts;
        unsigned short  pwd_failure_mins; /* account suspended if pwd_attempts
                                           exceeded within pwd_failure_mins*/
        bool            log_success_logon;          (pg491.22)
        bool            log_failed_logon;          (pg491.22)
        bool            log_db_updates;            (pg491.22)
        char            *language_p;
        bool            allow_override;            (pg491.22)
    }
}

```

+++ SEC - Security Services +++

```

    bool          annot_security;      /* Not available in ISTK */    (pg491.22)
    bool          pwd_chg_upon_reset;  /* Not available in ISTK */    (pg491.22)
} opt;
} SEC_system_opt_typ;

```

```

/*-----*\
| This is the structure which represents the FileNet security services
| attributes for a given domain.
\*-----*/

```

```

typedef struct SEC_system_desc_typ
{
    SEC_id_typ          system_id;      /* equal to svc process obj id */    (pg373.32)
    ASE_ssn_typ         ssn;           /* system ssn */                    (pg483.47)
    bool                device_security; /* 0=off,1=on */                     (pg491.22)
    bool                no_func_def_ok; /* 0=no,1=yes */                     (pg491.22)
    ASE_time_typ        update_time;   /* stamp of time db last updted */  (pg489.31)
    SEC_time_range_typ  logon_times;   /* allowable logon times */         (pg373.23)
    unsigned short      max_sessions;  /* max nbr of user sessions */      *
    bool                pwd_special_char; /* special char required in pwd? */  (pg491.22)
    unsigned char       pwd_min_len;   /* minimum pwd length */            *
    unsigned short      pwd_renewal_days; /* time until pwd renewal, days */  *
    unsigned short      pwd_grace_period; /* time until sspnsn w/no pwd delta*/
    unsigned short      pwd_attempts;  /* attmpts until sessn is disabled */
    unsigned short      pwd_failure_mins; /* time period in mins for fail */
    bool                log_success_logon; /* logging attributes... */         (pg491.22)
    bool                log_failed_logon; /* logs a failed logon attempt */    (pg491.22)
    bool                log_db_updates; /*                               */    (pg491.22)
    SEC_language_typ    language;      /* system language */               (pg372.33)
    bool                allow_override; /*                               */    (pg491.22)
} SEC_system_desc_typ;

```

```

/*-----*\
| Information related to database name mapping.
\*-----*/

```

```

#define SEC_DBMAP_NOMAP_ID          0
#define SEC_DBMAP_FOPEN_ID          1
#define SEC_DBMAP_LAST_RESERVED_ID 100

#define SEC_DBMAP_NOMAP_NAME        "F_NOMAP"
#define SEC_DBMAP_FOPEN_NAME        "f_open"          /* STR# 36308 */
#define SEC_DBMAP_FOPEN_PWD         ""                /* STR# 36308, 36537 */

```

```

typedef short SEC_optdbinfo_typ;

#define SEC_OPTDB_NAME              1
#define SEC_OPTDB_PASSWORD          2
#define SEC_MAXDB_OPTIONS           SEC_OPTDB_PASSWORD

```

+++ SEC - Security Services +++

```
typedef struct SEC_dbinfo_opt_typ
{
    SEC_optdbinfo_typ          option_type;          (pg382.43)
    union
    {
        char                  *dbname_p;
        char                  *dbpwd_p;
    } opt;
} SEC_dbinfo_opt_typ;
```

```
typedef struct SEC_dbinfo_desc_typ
{
    SEC_dbname_typ            db_name;              (pg371.32)
    SEC_id_typ                db_pseudo_id;        (pg373.32)
    SEC_id_typ                db_creator_id;       (pg373.32)
} SEC_dbinfo_desc_typ;
```

```
typedef struct SEC_dbinfo_list_typ
{
    struct SEC_dbinfo_list_typ *next_p;          (pg383.22)
    SEC_dbinfo_desc_typ        dbinfo;          (pg383.16)
} SEC_dbinfo_list_typ;
```

```
/*-----*\
| This structure is a linked list of members with the associate class
| value.
/*-----*/
```

```
typedef struct SEC_member_list_typ
{
    struct SEC_member_list_typ *next_p;          (pg383.34)
    SEC_id_typ                member_id;         /* group member id */ (pg373.32)
    SEC_object_class_typ      member_class;     /* group member class */ (pg375.33)
} SEC_member_list_typ;
```

```
/*-----*\
|
/*-----*/
```

```
typedef struct SEC_function_list_typ
{
    struct SEC_function_list_typ *next_p;          (pg383.44)
    SEC_function_name_typ      function_name;     (pg371.28)
    SEC_id_typ                function_id;       (pg373.32)
} SEC_function_list_typ;
```

```
typedef struct SEC_function_members_typ
{
    struct SEC_function_members_typ *next_p;      (pg384.4)
```

+++ SEC - Security Services +++

```

SEC_function_name_typ      function_name;          (pg371.28)
unsigned short             nbr_members;
SEC_id_typ                 *members_p;          (pg373.32)
} SEC_function_members_typ;

```

```

typedef struct SEC_user_logon_list_typ
{
    struct SEC_user_logon_list_typ  *next_p;      (pg384.14)
    SEC_name_typ                    user_name;     (pg371.25)
    SEC_name_typ                    endpt_name;    (pg371.25)
    ASE_time_typ                    logon_time;    (pg489.31)
    ASE_time_typ                    expires_time;  (pg489.31)
    unsigned short                  nbr_sessions;
} SEC_user_logon_list_typ;

```

```

typedef unsigned long        SEC_leftoff_typ;

```

```

typedef struct SEC_admin_class_typ
{
    bool                            supervisor;    (pg491.22)
    bool                            principal;      (pg491.22)
    bool                            group;          (pg491.22)
    bool                            password;       (pg491.22)
    bool                            notadmin;      (pg491.22)
} SEC_admin_class_typ;

```

```

/*-----*\
| Logon statistics for a given user.
\*-----*/

```

```

typedef struct SEC_stats_desc_typ
{
    unsigned long                   nbr_logons;     /* total nbr times logged in */
    ASE_time_typ                    success_log_time; /* last successful logon time */ (pg489.31)
    SEC_name_typ                    success_where;  /* where successful logon */ (pg371.25)
    ASE_time_typ                    failed_log_time; /* last failed logon time */ (pg489.31)
    SEC_name_typ                    failed_where;  /* last failed where */ (pg371.25)
    error_typ                       failed_err;    /* last failed error */ (pg493.26)
    bool                            pwd_grace_period; /* has grace period started */ (pg491.22)
    ASE_time_typ                    pwd_expires_time; /* time pwd expires */ (pg489.31)
} SEC_stats_desc_typ;

```

```

/*-----*\
| Access restrictions for various objects (eg. documents)
\*-----*/

```

```

typedef struct SEC_access_restrictions_typ
{
    SEC_id_typ                      read_group;    (pg373.32)
    SEC_id_typ                      write_group;   (pg373.32)
}

```

+++ SEC - Security Services +++

(pg373.32)

```
SEC_id_typ          AX_group;
} SEC_access_restrictions_typ, SEC_access_restrictions;

/*-----*\
| Service who identifier
\*-----*/
typedef struct SEC_identification_typ
{
    char          identification [356];
} SEC_identification_typ;

/*-----*\
| Leftoff type on getting a logon list.
\*-----*/
typedef char          SEC_logon_leftoff_typ [16];

/*-----*\
| The following block of structures have been moved back here from SEC.h
| so that anyone using SEC_update_info can have access to these defines.
\*-----*/
typedef unsigned long          SEC_update_typ;

#define SEC_UPDATE_USER_PASSWORD          (unsigned short) 0
#define SEC_UPDATE_USER_LANGUAGE          (unsigned short) 1
#define SEC_UPDATE_USER_PRIMARY           (unsigned short) 2
#define SEC_UPDATE_USER_FORLOGON          (unsigned short) 3
#define SEC_UPDATE_SVCPRCS_NAME           (unsigned short) 4
#define SEC_UPDATE_SVCPRCS_PASSWORD       (unsigned short) 5
#define SEC_UPDATE_USER_NAME              (unsigned short) 6

/*****\
*****
*   OLD SEC DEFINES
*****
\*****/
#define SEC_MAX_OLD_FUNCTIONS 10
#define ACCESS_APPROVED(x) (sec_function_results_p [x] == ( char ) 1)
#define SEC_MAX_GROUP_ID 4294967295
#define SEC_INVALID_ID SEC_MAX_GROUP_ID

typedef struct SEC_func_list_typ
{
    char          *name_p;
} SEC_func_list_typ;

#define SEC_UNKNOWN_FUNC "unknown"
#define SEC_FUNC_LIST_GROUP_ADMIN "GroupAdmin"
```

+++ SEC - Security Services +++

```
#define SEC_FUNCLIST_SITE_ADMIN "SiteAdmin"
#define SEC_FUNCLIST_FUNCTION_ADMIN "FunctionAdmin"
#define SEC_FUNCLIST_SECURITY_ADMIN "SecurityAdmin"
#define SEC_FUNCLIST_TIME_ADMIN "TimeAdmin"
#define SEC_FUNCLIST_PARAM_ADMIN "ParamAdmin"
#define SEC_FUNCLIST_PASSWORD_ADMIN "PasswordAdmin"
#define SEC_FUNCLIST_SERVICE_ADMIN "ServiceAdmin"
```

```
#define SEC_MEMLIST_SUBSCRIPTION 1
#define SEC_MEMLIST_SUBSCRIBER 2
```

```
typedef unsigned short SEC_info_type_typ;
```

```
#define SEC_INFOTYPE_USER 1
#define SEC_INFOTYPE_USER_MEMBER 2
#define SEC_INFOTYPE_TERMINAL 3
#define SEC_INFOTYPE_TERMINAL_MEMBER 4
#define SEC_INFOTYPE_FUNCTION 5
#define SEC_INFOTYPE_FUNCTION_MEMBER 6
#define SEC_INFOTYPE_DEFAULTS 7
#define SEC_INFOTYPE_DATE_TIME 8
#define SEC_INFOTYPE_SERVICE_PROCESS 9
#define SEC_INFOTYPE_UPDATE_TIME 10
#define SEC_INFOTYPE_GROUP 11
```

```
#define SEC_GUID_REGISTRY_FORMAT_LENGTH 36
```

```
typedef char SEC_guid_typ [SEC_GUID_REGISTRY_FORMAT_LENGTH+1];
```

```
typedef unsigned long SEC_slu_accounting_typ;
```

```
#define SEC_SLU_NO_POOLING 0
#define SEC_SLU_ACCOUNTING_STRICT 1
#define SEC_SLU_ACCOUNTING_APPROX 2
```

```
typedef struct {
    SEC_name_typ name;
} SEC_name_struct_typ; (pg371.25)
```

```
typedef struct SEC_security_info_typ
{
    SEC_name_typ name; (pg371.25)
    SEC_id_typ id; (pg373.32)
    SEC_id_typ prim_group; (pg373.32)
    unsigned short language;
    bool for_login; (pg491.22)
} SEC_security_info_typ;
```

```
typedef struct {
    unsigned short field_type;
    union
```


+++ SEC - Security Services +++

```
{
    char          *name_p;
    char          *password;
    unsigned short language;
    bool          for_login;
} u;
} SEC_update_info_typ;

typedef unsigned short      SEC_memlist_type_typ;

typedef struct SEC_memlist_typ
{
    SEC_memlist_type_typ    list_type;
    SEC_id_typ              security_id;
    unsigned long           included_len;
    SEC_id_typ              included [150];
    unsigned long           excluded_len;
    SEC_id_typ              excluded [50];
} SEC_memlist_typ;

/* BFN 4.0.40 Release - SEC RM Config *****/
typedef struct SEC_rm_config_typ
{
    unsigned long           key_id;
    unsigned long           declare_level;
    unsigned long           rm_group_id;
    char                    rm_group_name[128];
    unsigned long           purge_level;
    unsigned long           scrub_level;
    unsigned long           log_level;
    unsigned long           lock_sysadming;
} SEC_rm_config_typ;

#define SEC_MAX_DOCS_TO_DECLARE      200
#define SEC_RM_SYSTEM_LEVEL_CONFIG_KEY 0xFFFFF00 /* Default key_id */
#define SEC_RM_DECLARE_PERM_READ_ONLY 1 /* Default record declare level */
*/
#define SEC_RM_DECLARE_PERM_ANNOT_ONLY 2
#define SEC_RM_DECLARE_PERM_NO_CHANGE 3
#define SEC_RM_DEFAULT_RM_GROUP_ID SEC_UNDEFINED_ID
#define SEC_RM_INVALID_RM_GROUP_ID 0
#define SEC_RM_DEFAULT_PURGE_LEVEL 0
#define SEC_RM_DEFAULT_SCRUB_LEVEL 0
#define SEC_RM_LOG_LEVEL_ERR_ONLY 0 /* Default log level */
#define SEC_RM_LOG_LEVEL_VERBOSE 1
#define SEC_RM_DEFAULT_SYSADMING_LOCK 0 /* Default = do NOT lock */

/* BFN 4.0.40 Release - SEC RM Config *****/
```

+++ SEC - Security Services +++

```
typedef struct SEC_security_defaults_typ
{
    unsigned short    terminal_security;
    unsigned short    undefined_functions;
    unsigned short    default_language;
    SEC_name_typ      noone_name;           (pg371.25)
    SEC_name_typ      anyone_name;        (pg371.25)
    ASE_domain_name_typ default_domain;    (pg486.33)
} SEC_security_defaults_typ;

typedef struct
{
    SEC_name_typ      name;                (pg371.25)
    SEC_id_typ        id;                 (pg373.32)
    SEC_id_typ        primary_group;      (pg373.32)
    unsigned short    language;
    bool              for_login;          (pg491.22)
} SEC_object_info_typ;

/*-----*\
| Structure used to store info on each process using (via SEC_logon)
| a set of credentials from the global credentials cache.
| Not used on systems where the credentials are stored in the environment
\*-----*/
typedef struct SEC_sess_proc_rec_typ {
    int                pid;
    char               proc_name[40];
    long               start_time;
    int                ref_count;
    struct SEC_proc_logon_domain_typ *domain_logons;
    struct SEC_sess_proc_rec_typ *next_p;   (pg388.32)
} SEC_sess_proc_rec_typ;

/* enum SEC_ACTION {SEC_SET=1, SEC_GET}; */
#define SEC_SET      1
#define SEC_GET      2

#define SEC_XIN_PROTOCOL      0
#define SEC_XNOT_IN_PROTOCOL  1
#define SEC_SCT_PROTOCOL     2

/*$D err_SEC */

#define SEC_pro_err(x) err_encode (err_SEC, SEC_XIN_PROTOCOL, (x))
#define SEC_sct_err(x) err_encode (err_SEC, SEC_SCT_PROTOCOL, (x))
#define SEC_dev_err(x) err_encode (err_SEC, SEC_XNOT_IN_PROTOCOL, (x))

#ifndef FN_ENTRY
```

+++ SEC - Security Services +++

```
#define FN_ENTRY
#endif
```

```
/*----- SEC_daemon Defines -----*/
```

```
/* retry to open the log file N times before giving up */
#define RETRY_OPEN_COUNT 60
/* allow LOCAL_TRAN to be logged into log file before the file must be closed*/
#define LOCAL_TRAN 100

#define SEC_DAEMON_IDLE_TIME 10000 /* wait time in 100th of microseconds
before closing any open files 60+40 secs
*/
```

```
/*-----*/
```

```
/*-----*\
| In Protocol Definitions.
\*-----*/
```

```
#define SEC_err_other_error SEC_pro_err (1)
/*$M SEC error other than in protocol definition. */
```

```
/* SCT_BadPassword */
#define SEC_err_invalid_password SEC_sct_err (2)
/*$M The password provided does not match that in the data base. */
```

```
/* SCT_NotMember */
#define SEC_err_not_member SEC_sct_err (4)
/*$M The requested object does not have a membership intersection. */
```

```
/* SCT_ReadDenied */
#define SEC_err_read_denied SEC_sct_err (5)
/*$M Read permission is denied. */
```

```
/* SCT_WriteDenied */
#define SEC_err_write_denied SEC_sct_err (6)
/*$M Write permission is denied. */
```

```
/* SCT_AppendOrExecuteDenied */
#define SEC_err_ax_denied SEC_sct_err (7)
/*$M Append/execute permission is denied. */
```

```
/* SCT_GroupNotFound */
#define SEC_err_object_not_found SEC_sct_err (8)
/*$M The user, group, or device object information could not be found. */
```

```
/* SCT_CannotDecodeAR */
```

+++ SEC - Security Services +++

```
#define SEC_err_cant_decode_AR          SEC_sct_err (10)
/*$M SEC could not decode the access restrictions. */

/* SCT_BadHandle */
#define SEC_err_no_logon                SEC_sct_err (11)
/*$M The requesting user is not logged onto the security service. */

/* SCT_BadLanguageName */
#define SEC_err_language_not_found     SEC_sct_err (12)
/*$M The specified language name/language id was not found. */

#define SEC_err_undefined_object       SEC_pro_err (13)
/*$M This object contains access restrictions which are not defined here.
The object you tried to access is owned by a group that is not defined on
the local system. To remove this error the groups in the access
restriction for this object must be defined in the local security data
base. */

#define SEC_err_invalid_service_logon   SEC_pro_err (14)
/*$M Service logon may not be done after a SEC_logon */

#define SEC_err_credential_not_in_environment SEC_pro_err (15)
/*$M The specified credential could not be found in the environment
during logoff. Most likely the source credential data has been
overwritten (as designed). */

#define SEC_err_invalid_session_nbr     SEC_pro_err (93)
/*$M An invalid session number was supplied. */

#define SEC_err_invalid_info_type       SEC_pro_err (94)
/*$M Invalid information type field. */

#define SEC_err_invalid_name            SEC_pro_err (95)
/*$M The name provided is not of a valid syntax. */

#define SEC_err_invalid_param           SEC_pro_err (96)
/*$M An invalid parameter was supplied. */

#define SEC_err_access_denied           SEC_pro_err (97)
/*$M Access to the security data base is refused. */

#define SEC_err_duplicate_logon         SEC_pro_err (98)
/*$M A duplicate logon was performed. */

#define SEC_err_max_sessions            SEC_pro_err (99)
/*$M The user has already reached the maximum allowable number of sessions. */

#define SEC_err_cant_remove_inheritance SEC_pro_err (100)
```

+++ SEC - Security Services +++

```
/*$M Cannot remove inheritance if logged on. */

#define SEC_err_invalid_nbr_options      SEC_pro_err (101)
/*$M Too many options were passed for this data structure. */

#define SEC_err_invalid_object_class     SEC_pro_err (102)
/*$M An invalid object class was specified. */

#define SEC_err_invalid_device_class     SEC_pro_err (103)
/*$M An invalid device class was specified. */

#define SEC_err_invalid_admin_class      SEC_pro_err (104)
/*$M An invalid administrative class was specified. */

#define SEC_err_invalid_time             SEC_pro_err (105)
/*$M The current time is not within the restricted time range. */

#define SEC_err_invalid_sess_override    SEC_pro_err (106)
/*$M An invalid session override value was specified. */

#define SEC_err_invalid_max_sessions     SEC_pro_err (107)
/*$M An invalid max-sessions value was passed to SEC. */

#define SEC_err_duplicate_object         SEC_pro_err (108)
/*$M A duplicate object already exists in the data base. */

#define SEC_err_invalid_object_name      SEC_pro_err (109)
/*$M The object name provided was in an incorrect format. */

#define SEC_err_invalid_comments         SEC_pro_err (110)
/*$M The comments length provided exceeded the maximum allowable length. */

#define SEC_err_invalid_object_filter     SEC_pro_err (111)
/*$M An invalid object filter was passed to SEC. */

#define SEC_err_invalid_system_filter     SEC_pro_err (112)
/*$M An invalid system filter was passed to SEC. */

#define SEC_err_invalid_log              SEC_pro_err (113)
/*$M An invalid log field was passed to SEC. */

#define SEC_err_invalid_dev_security     SEC_pro_err (114)
/*$M An invalid device security value was passed to SEC. */

#define SEC_err_invalid_func_def         SEC_pro_err (115)
/*$M An invalid no-function-definition-ok value was passed to SEC. */

#define SEC_err_invalid_pwd_spec_char    SEC_pro_err (116)
```

+++ SEC - Security Services +++

```
/*$M The special character designator provided is not valid. */

#define SEC_err_invalid_pwd_min_length SEC_pro_err (117)
/*$M The minimum enforced password length in the system defaults is out of range.*/

#define SEC_err_invalid_pwd_aging SEC_pro_err (118)
/*$M An invalid password aging value was passed to SEC. */

#define SEC_err_invalid_pwd_attempts SEC_pro_err (119)
/*$M An invalid number of password attempts was passed to SEC. */

#define SEC_err_invalid_pwd_suspending SEC_pro_err (120)
/*$M An invalid password-suspending time was passed to SEC. */

#define SEC_err_invalid_pwd_memory SEC_pro_err (121)
/*$M An invalid password-memory time limit was passed to SEC. */

#define SEC_err_invalid_key_num SEC_pro_err (122)
/*$M An incorrect key_num type was passed to SEC. */

#define SEC_err_invalid_member_deletion SEC_pro_err (123)
/*$M The member id and group id were identical. This is illegal. */

#define SEC_err_no_such_service SEC_pro_err (124)
/*$M The service specified does not exist. */

#define SEC_err_device_access_denied SEC_pro_err (125)
/*$M The device security prevents access. */

#define SEC_err_not_user SEC_pro_err (126)
/*$M The object is required to be of the user class. */

#define SEC_err_invalid_device_name SEC_pro_err (127)
/*$M The device name provided is in an incorrect format. */

#define SEC_err_bad_magic SEC_pro_err (128)
/*$M A bad magic number was discovered in memory. */

#define SEC_err_invalid_comments_search SEC_pro_err (129)
/*$M Searching for a comment is not a supported feature. */

#define SEC_err_user_expired SEC_pro_err (130)
/*$M The account has expired and is no longer valid. */

#define SEC_err_invalid_dn_length SEC_pro_err (131)
/*$M DN data exceeds maximum length allowed. */

#define SEC_err_bad_service SEC_pro_err (132)
```

+++ SEC - Security Services +++

```
/*$M A bad service name was provided. */

#define SEC_err_dn_size_too_short      SEC_pro_err (133)
/*$M DN buffer size too short. */

#define SEC_err_illegal_instruction    SEC_pro_err (136)
/*$M The function call executed is not legal. */

#define SEC_err_invalid_function_name  SEC_pro_err (138)
/*$M The function name provided was in an invalid format. */

#define SEC_err_invalid_logon_time     SEC_pro_err (139)
/*$M The calculated duration of this logon instance has been exceeded. */

#define SEC_err_device_expired         SEC_pro_err (140)
/*$M The device specified has exceeded its expiration date. */

#define SEC_err_group_expired          SEC_pro_err (141)
/*$M The group specified has exceeded its expiration date. */

#define SEC_err_invalid_pwd_renewal_days SEC_pro_err (142)
/*$M The number of renewal days is out of range. */

#define SEC_err_invalid_grace_period   SEC_pro_err (143)
/*$M The specified grace period is out of range. */

#define SEC_err_invalid_failure_mins   SEC_pro_err (144)
/*$M The specified number of failure minutes is out of range. */

#define SEC_err_invalid_search_param   SEC_pro_err (145)
/*$M The parameter specified is not a search filter. */

#define SEC_err_primary_not_found      SEC_pro_err (146)
/*$M The primary group specified was not found. */

#define SEC_err_invalid_primary        SEC_pro_err (147)
/*$M The primary group specified is not a group. */

#define SEC_err_invalid_conditional    SEC_pro_err (148)
/*$M The conditional operator specified is invalid. */

#define SEC_err_invalid_sysopt         SEC_pro_err (149)
/*$M An invalid system option was provided. */

#define SEC_err_invalid_objopt         SEC_pro_err (150)
/*$M An invalid object option was provided. */

#define SEC_err_system_not_found       SEC_pro_err (151)
```

+++ SEC - Security Services +++

```
/*$M The system defaults information could not be found. */

#define SEC_err_group_not_found          SEC_pro_err (152)
/*$M The group information could not be found. */

#define SEC_err_deleted_not_found        SEC_pro_err (153)
/*$M The deleted object information could not be found. */

#define SEC_err_duplicate_group          SEC_pro_err (155)
/*$M The group-member in the data base already exists. */

#define SEC_err_duplicate_deleted        SEC_pro_err (156)
/*$M The specified object has already been deleted. */

#define SEC_err_duplicate_system         SEC_pro_err (157)
/*$M The specified system structure already exists in the data base. */

#define SEC_err_invalid_system_options   SEC_pro_err (158)
/*$M The system option list provided is invalid. */

#define SEC_err_invalid_annot_security   SEC_pro_err (159)
/*$M An illegal value was provided for the annot_security field. */

#define SEC_err_invalid_pwd_chg_upon_reset SEC_pro_err (160)
/*$M An illegal value was provided for the pwd_chg_upon_reset field. */

#define SEC_err_invalid_language         SEC_pro_err (166)
/*$M The system language option is in an invalid format */

#define SEC_err_function_not_found       SEC_pro_err (167)
/*$M The specified function name was not found. */

#define SEC_err_funcmbr_not_found        SEC_pro_err (168)
/*$M The specified function member combination was not found. */

#define SEC_err_duplicate_function        SEC_pro_err (169)
/*$M The specified function name already exists. */

#define SEC_err_duplicate_funcmbr        SEC_pro_err (170)
/*$M The specified function member combination already exists. */

#define SEC_err_funcmbr_out_of_sync      SEC_pro_err (171)
/*$M The function ids or member ids are out-of-sync. */

#define SEC_err_invalid_min_range        SEC_pro_err (172)
/*$M The specified minute range is invalid. */

#define SEC_err_invalid_hour_range       SEC_pro_err (173)
```


+++ SEC - Security Services +++

```
/*$M The specified hour range is invalid. */

#define SEC_err_invalid_dweek_range      SEC_pro_err (174)
/*$M The specified days of week range is invalid. */

#define SEC_err_invalid_call             SEC_pro_err (175)
/*$M The executed entry-point is not supported in this state. */

#define SEC_err_invalid_handle_ssn       SEC_pro_err (176)
/*$M The ssn stored in the handle is invalid. */

#define SEC_err_pwd_attempts_exceeded    SEC_pro_err (177)
/*$M The number of allowable failed password attempts has been exceeded. */

#define SEC_err_password_expired         SEC_pro_err (178)
/*$M The password has expired. */

#define SEC_err_no_special_char          SEC_pro_err (179)
/*$M The specified password requires a special character. */

#define SEC_err_invalid_handle           SEC_pro_err (180)
/*$M The specified handle should be non-null. */

#define SEC_err_invalid_tag_type         SEC_pro_err (181)
/*$M The buffer tag definition is not of a recognized type. */

#define SEC_err_obj_update_denied        SEC_pro_err (182)
/*$M The object update is denied. */

#define SEC_err_member_add_denied        SEC_pro_err (183)
/*$M The member addition is denied. */

#define SEC_err_obj_delete_denied        SEC_pro_err (184)
/*$M The object delete is denied. */

#define SEC_err_terminate_logon_denied   SEC_pro_err (185)
/*$M The termination of a logon is denied due to inadequate permissions. */

#define SEC_err_pwd_update_denied        SEC_pro_err (186)
/*$M The password update is denied due to inadequate permissions. */

#define SEC_err_member_delete_denied     SEC_pro_err (187)
/*$M The deletion of the specified member from the group is denied. */

#define SEC_err_obj_add_denied           SEC_pro_err (188)
/*$M The addition of the specified object is denied. */

#define SEC_err_func_add_denied          SEC_pro_err (189)
```

+++ SEC - Security Services +++

```
/*$M The user does not have the permissions to add a function. */
#define SEC_err_func_delete_denied      SEC_pro_err (190)
/*$M The user does not have the permissions to delete a function. */

#define SEC_err_fmbr_add_denied         SEC_pro_err (191)
/*$M The user does not have the permissions to add a function member. */

#define SEC_err_fmbr_delete_denied      SEC_pro_err (192)
/*$M The user does not have the permissions to delete a function member. */

#define SEC_err_too_many_termids        SEC_pro_err (193)
/*$M The terminal identifier maximum has been reached for this server. */

#define SEC_err_invalid_termid         SEC_pro_err (194)
/*$M An invalid terminal identifier was detected. */

#define SEC_err_groups_out_of_sync      SEC_pro_err (195)
/*$M The member/group relationship in the groups table is out of sync. */

#define SEC_err_cannot_update_primary   SEC_pro_err (197)
/*$M Changing the primary group of a group is an illegal operation. */

#define SEC_err_maximum_logons_reached  SEC_pro_err (198)
/*$M The concurrent license limit has been reached. */

#define SEC_err_invalid_terminal_name   SEC_pro_err (199)
/*$M The terminal name provided is in an incorrect format. */

#define SEC_err_invalid_password_opt    SEC_pro_err (201)
/*$M The semantic use of the password option is invalid. */

#define SEC_err_invalid_admin_group_opt SEC_pro_err (202)
/*$M It is illegal to add or update the admin group. */

#define SEC_err_invalid_exp_time_opt    SEC_pro_err (203)
/*$M It is not allowable to search for an object by expiration time. */

#define SEC_err_invalid_success_opt     SEC_pro_err (204)
/*$M It is not allowable to add or update an object by success_where. */

#define SEC_err_invalid_failed_opt      SEC_pro_err (205)
/*$M It is not allowable to add or update an object by failed_where. */

#define SEC_err_invalid_error_opt       SEC_pro_err (206)
/*$M It is not allowable to add or update an object by its failure error. */

#define SEC_err_invalid_comments_opt    SEC_pro_err (207)
```

+++ SEC - Security Services +++

```
/*$M It is not allowable to search for an object by its comment field. */

#define SEC_err_invalid_admin_opt      SEC_pro_err (208)
/*$M It is illegal to set the admin class for a non-user/system object. */

#define SEC_err_invalid_language_opt   SEC_pro_err (209)
/*$M It is not allowable to add or update the language of a non-user/system. */

#define SEC_err_invalid_dev_class_opt  SEC_pro_err (210)
/*$M It is not allowable to set the device class for a non-device/system. */

#define SEC_err_invalid_obj_class_opt  SEC_pro_err (211)
/*$M It is not allowable to change the object class of an existing object. */

#define SEC_err_invalid_max_sess_opt   SEC_pro_err (212)
/*$M It is not allowable to set the maximum sessions for a device object. */

#define SEC_err_cannot_specify_primary SEC_pro_err (213)
/*$M It is illegal to specify a primary group for a group object. */

#define SEC_err_invalid_time_range     SEC_pro_err (214)
/*$M An invalid time combination was specified for the dweek/min/hour values. */

#define SEC_err_obj_read_denied        SEC_pro_err (215)
/*$M The user does not have the admin permission to read the object. */

#define SEC_err_sys_read_denied        SEC_pro_err (216)
/*$M The user does not have the admin permission to read the system defaults. */

#define SEC_err_mbr_read_denied        SEC_pro_err (217)
/*$M The user does not have the admin permission to read the member list. */

#define SEC_err_grp_read_denied        SEC_pro_err (218)
/*$M The user does not have the admin permission to read the group list.
The user must be of an administrative class to view groups. */

#define SEC_err_fun_read_denied        SEC_pro_err (219)
/*$M The user does not have the admin permission to read the function list.
The user must be of an administrative class to view function membership. */

#define SEC_err_fmbr_read_denied       SEC_pro_err (220)
/*$M The user does not have the admin permission to read the function members.
The user must be of an administrative class to view function membership. */

#define SEC_err_pwd_len_out_of_range   SEC_pro_err (221)
/*$M The length of the password provided is out of range.
A password should at least have a minimum length of that specified by the
system defaults and cannot have a length greater than 8 characters.*/
```

+++ SEC - Security Services +++

```
#define SEC_err_bad_filename          SEC_pro_err (222)
/*$M An operation on the specified file failed.
An error occurred attempting to open the specified file name. */

#define SEC_err_bad_import_version    SEC_pro_err (223)
/*$M The import version contained in the import file is not recognized.
The import routine expects the export file to contain a version
stamp of an explicit value. */

#define SEC_err_invalid_char_set      SEC_pro_err (224)
/*$M The export file has a different default character set than the import system.
It is required that an export/import be performed across systems which
contain the same default character set. It is not possible to convert
systems across different character sets due to encrypted password
incompatibilities. */

#define SEC_err_bad_import_format     SEC_pro_err (226)
/*$M The import file is in an incorrect format.
When the import routine attempts to parse the export file, it expects
the information to be of certain sizes and in certain locations within
the export file. If it encounters a conflict, then this error will
occur. It is possible that the export file is corrupt, or that it
has been manually edited. */

#define SEC_err_invalid_import_param  SEC_pro_err (227)
/*$M There were missing or improper import parameters. */

#define SEC_err_invalid_domain        SEC_pro_err (228)
/*$M The domain length may not exceed 20 characters in length. */

#define SEC_err_conflicting_obj_class SEC_pro_err (229)
/*$M The import object class conflicts with that of an existing object.
When the security service imports an object, it will look for objects
of the same name within the import data base. If those objects exist, then
it will compare the object class. If the object class is not the same,
then the import will not continue. This check exists to protect
conflicting membership (ie. a user cannot be made a member of a user). */

#define SEC_err_invalid_service_name  SEC_pro_err (230)
/*$M An invalid security service name was provided.
This error typically occurs when the security service expects to
receive the service name as a parameter, but has received nothing
instead. */

#define SEC_err_export_denied         SEC_pro_err (231)
/*$M A user who is not SysAdmin attempted to export the security data base.
For the sake of security, SEC requires that only the SysAdmin user may
```

+++ SEC - Security Services +++

perform this action. This prevents any user from being able to export a data base without being authenticated. */

```
#define SEC_err_import_denied          SEC_pro_err (232)
/*$M A user who is not SysAdmin attempted to import the security data base.
For the sake of security, SEC requires that only the SysAdmin user may
perform this action. This prevents any user from being able to import
a data base without being authenticated. */
```

```
#define SEC_err_stale_session          SEC_pro_err (234)
/*$M The session handle is stale. The security service was rebooted.
Typically, this error is seen only internal to the security service
so that the client can determine if it needs to relogin to the
security service after a server reboot. */
```

```
#define SEC_err_invalid_relogin       SEC_pro_err (235)
/*$M The relogin information provided is inaccurate.
Once the security service issues the SEC_err_stale_session, the client
SEC code will attempt to relogin to the security service. If the
information which is sent to the security service from the client
is incorrect, then this error will occur. */
```

```
#define SEC_err_admingrp_not_found    SEC_pro_err (236)
/*$M The provided admin group does not exist in the security data base.
The data base object table was searched for this object, however, it
was not able to be found. */
```

```
#define SEC_err_invalid_admingrp_opt  SEC_pro_err (237)
/*$M The admin group option may not have an id of 0, 1 or 6.
The id 0 equates to (NONE), the id 1 equates to (ANYONE) and the
id 6 equates to (UNDEFINED). */
```

```
#define SEC_err_invalid_admingrp      SEC_pro_err (238)
/*$M The provided admin group is not a group. */
```

```
#define SEC_err_invalid_epassword     SEC_pro_err (239)
/*$M The encrypted password in the data base consists of nulls. */
```

```
#define SEC_err_missing_license       SEC_pro_err (240)
/*$M The concurrent license is either expired or missing. */
```

```
#define SEC_err_map_prin_not_found    SEC_pro_err (241)
/*$M The Map Principal to DN information could not be found. */
```

```
#define SEC_err_duplicate_map_prin    SEC_pro_err (242)
/*$M The Map Principal to DN data already exists */
```

```
#define SEC_err_ce_dom_not_found      SEC_pro_err (243)
```

+++ SEC - Security Services +++

```
/*$M The CE Domain information could not be found.
Make sure SEC_map tool is run to map between IS and CE users. */

#define SEC_err_duplicate_ce_dom          SEC_pro_err (244)
/*$M The CE Domain data already exists */

#define SEC_err_batch_update_dn_by_id     SEC_pro_err (245)
/*$M Error occurred in SEC_update_dn_by_id(). See SysLog */

#define SEC_err_pid_not_found             SEC_pro_err (250)
/*$M The fnfork program could not find the pid in the term_id list. */

#define SEC_err_invalid_pc_admin          SEC_pro_err (251)
/*$M The administrator must be SysAdmin to use the PC security administration
utility. */

#define SEC_err_sessgrp_not_found         SEC_pro_err (252)
/*$M The specified session group does not exist in the security
data base. */

#define SEC_err_invalid_sessgrp          SEC_pro_err (253)
/*$M The session group supplied is not a group. */

#define SEC_err_invalid_allow_override   SEC_pro_err (254)
/*$M An illegal value was provided for the allow_override field. */

#define SEC_err_cannot_specify_sessgrp   SEC_pro_err (255)
/*$M It is illegal to add/update a session group for a group or
device object. */

#define SEC_err_namemap_not_found         SEC_pro_err (256)
/*$M The database name map information could not be found. */

#define SEC_err_dbinfo_not_found         SEC_pro_err (257)
/*$M The database name information could not be found. */

#define SEC_err_duplicate_namemap        SEC_pro_err (258)
/*$M The FileNet to database name map information already exists. */

#define SEC_err_duplicate_dbinfo         SEC_pro_err (259)
/*$M The database map name already exists. */

#define SEC_err_invalid_db_filter        SEC_pro_err (260)
/*$M The dbinfo table search filter is not recognized. */

#define SEC_err_invalid_dbpassword       SEC_pro_err (261)
/*$M The database password is in an incorrect format. */
```

+++ SEC - Security Services +++

```
#define SEC_err_invalid_dbname          SEC_pro_err (262)
/*$M The database name provided is in an incorrect format. */

#define SEC_err_db_delete_denied        SEC_pro_err (263)
/*$M The user does not have permission to delete a dbinfo record. */

#define SEC_err_db_update_denied        SEC_pro_err (264)
/*$M The user does not have permission to update a dbinfo record. */

#define SEC_err_db_add_denied           SEC_pro_err (265)
/*$M The user does not have permission to add a dbinfo record. */

#define SEC_err_dbmap_denied            SEC_pro_err (266)
/*$M The user does not have permission to map to a db user. */

#define SEC_err_invalid_dbopt           SEC_pro_err (267)
/*$M The option specified is not valid. */

#define SEC_err_cannot_specify_dbmap    SEC_pro_err (268)
/*$M It is illegal to specify a db name map for a non-user. */

#define SEC_err_invalid_pointer         SEC_pro_err (269)
/*$M The pointer passed is NULL. */

#define SEC_err_invalid_nbr_bytes       SEC_pro_err (270)
/*$M The number of bytes value is zero. */

#define SEC_err_nomap                   SEC_pro_err (271)
/*$M The requesting user is refused access to the native database. */

#define SEC_err_nomemory_for_terminal_listing SEC_pro_err (272)
/*$M Could not allocate global memory for terminal listing. */

#define SEC_err_process_create_time     SEC_pro_err (273)
/*$M Could not get process create time in SEC. */

#define SEC_err_get_process_access_token SEC_pro_err (274)
/*$M Could not get process access token. */

#define SEC_err_insufficient_token_buffer SEC_pro_err (275)
/*$M Could not get sufficient buffer for token stats */

#define SEC_err_in_GetTokenInfo         SEC_pro_err (276)
/*$M Got error in GetTokenInformation */

#define SEC_err_invalid_event_type      SEC_pro_err (277)
/*$M Invalid event type */
```

+++ SEC - Security Services +++

```
#define SEC_err_database_level_not_match      SEC_pro_err (278)
/*$M Import database level does not match current database level */

#define SEC_err_service_is_remote            SEC_pro_err (279)
/*$M Configuration error; service is remote */

#define SEC_err_make_log_directory           SEC_pro_err (280)
/*$M Could not make log directory */

#define SEC_err_create_README                SEC_pro_err (281)
/*$M Could not create README in log directory */

#define SEC_err_write_README                 SEC_pro_err (282)
/*$M Could not write README in log directory */

#define SEC_err_db2_not_found                 SEC_pro_err (283)
/*$M The Db2 information could not be found. */

#define SEC_err_duplicate_db2                SEC_pro_err (284)
/*$M The DB2 acct name already exists. */

#define SEC_err_db2_pwd_max_length_reached   SEC_pro_err (285)
/*$M DB2 password length reached maximum length. */

#define SEC_err_db2_acctname_max_length_reached SEC_pro_err (286)
/*$M DB2 acct name length reached maximum length. */

#define SEC_err_db2_time_out_of_range        SEC_pro_err (287)
/*$M The time between DB2 client and server is out of range */

#define SEC_err_rm_config_not_found           SEC_pro_err (288)
/*$M The RM Configuration information could not be found. */

#define SEC_err_duplicate_rm_config          SEC_pro_err (289)
/*$M The RM Configuration data already exists */

#define SEC_err_invalid_rm_config_data       SEC_pro_err (290)
/*$M The RM Configuration data is invalid */

#define SEC_err_rm_config_access              SEC_pro_err (291)
/*$M Must be a member of group fnadmin to modify the Record Management settings */

#define SEC_err_rm_config_group_not_found    SEC_pro_err (292)
/*$M A configured Record Management group was not found. It must be created using the
security RM tool */

#define SEC_err_ext_authent_init_library_failed SEC_pro_err (293)
/*$M The extensible authentication library initialize entry point has returned an error */
```


+++ SEC - Security Services +++

```
#define SEC_err_ext_authent_validation_failed    SEC_pro_err (294)
/*$M The extensible authentication library has rejected the user credentials */

#define SEC_err_ext_authent_invalid_parameters    SEC_pro_err (295)
/*$M The extensible authentication library was passed invalid parameters */

#define SEC_err_ext_authent_mode_undetermined    SEC_pro_err (296)
/*$M Failed to determine authentication mode. The system will default to standard
authentication */

#define SEC_err_signals_undefined    SEC_pro_err (297)
/*$M Failed to install signal handlers (undefined) */

#define SEC_err_incompatible_client    SEC_pro_err (298)
/*$M Image Services has detected out of date client software. Please upgrade
your client software before continuing. Check the Image Services Release
Notes for more information. */

#define SEC_err_debug_info_only    SEC_pro_err (300)
/*$M Message is for SEC information and/or debugging purposes only */

#define SEC_err_run_SEC_convert    SEC_pro_err (301)
/*$M Must be root or fnsu to execute SEC_convert */

#define SEC_err_convert_bad_character    SEC_pro_err (302)
/*$M A bad character was detected */

#define SEC_err_convert_duplicate_name    SEC_pro_err (303)
/*$M A duplicate name exists */

#define SEC_err_convert_name_exceed_39_characters    SEC_pro_err (304)
/*$M A name exceeding 39 characters was found */

#define SEC_err_run_SEC_init_on_NT    SEC_pro_err (305)
/*$M Must be a member of group fnadmin to rebuild the security data base */

#define SEC_err_run_SEC_init_on_UNIX    SEC_pro_err (306)
/*$M Must be root to rebuild the security data base */

#define SEC_err_run_SEC_init_on_MPE    SEC_pro_err (307)
/*$M Must be fnadmin to rebuild the security data base */

#define SEC_err_permission_denied_on_NT    SEC_pro_err (308)
/*$M User is not a member of group fnadmin, permission denied */

#define SEC_err_permission_denied_on_UNIX    SEC_pro_err (309)
/*$M User is not root, permission denied */

#define SEC_err_permission_denied_not_SysAdmin    SEC_pro_err (310)
```

+++ SEC - Security Services +++

```
/*$M User is not SysAdmin, permission denied */

#define SEC_err_putenv_failed          SEC_pro_err (311)
/*$M System command putenv failed */

#define SEC_err_invalid_option_in_fnfork_exec    SEC_pro_err (312)
/*$M Invalid option in fnfork/exec */

#define SEC_err_exec_failed           SEC_pro_err (313)
/*$M Exec() failed */

#define SEC_err_fork_failed           SEC_pro_err (314)
/*$M Fork() failed */

#define SEC_err_CreateProcess_failed   SEC_pro_err (315)
/*$M CreateProcess failed */

#define SEC_err_create_SEC_anl_success_file      SEC_pro_err (316)
/*$M Could not create SEC_anl success file */

#define SEC_err_zero_session           SEC_pro_err (317)
/*$M Zero session */

#define SEC_err_data_segment_alloc      SEC_pro_err (318)
/*$M Data segment allocation verification failed */

#define SEC_err_QMA_null_pointer        SEC_pro_err (319)
/*$M Unable to get valid logging queue */

#define SEC_err_functionality_not_supported      SEC_pro_err (320)
/*$M The functionality requested is not supported by present version of
   IMS server */

#define SEC_err_ext_validation_pwd_failed    SEC_pro_err (321)
/*$M Custom password validation failed. The specified password does not
   meet policy requirements. Please check the password format rules
   and try again */

#define SEC_err_map_cn_not_found         SEC_pro_err (322)
/*$M The common name to IS principal mapping records could not be found. */

#define SEC_err_map_cn_duplicate        SEC_pro_err (323)
/*$M The common name to IS principal mapping record already exists. */

#define SEC_err_batch_update_prin_by_cn    SEC_pro_err (324)
/*$M Error occurred in SEC_update_prin_by_cn(). See SysLog */

#define SEC_err_cn_size_too_short       SEC_pro_err (325)
```

+++ SEC - Security Services +++

```
/*$M Common Name buffer size too short (Extensible Authentication). */

#define SEC_err_ext_authent_missing_map_entry      SEC_pro_err (326)
/*$M The custom extensible authentication library has validated the user
   credentials, but a mapped security username was not found */

#define SEC_err_sec_map_ext_auth_access          SEC_pro_err (327)
/*$M Must be a member of group fnadmin to use the SEC_map_ext_auth tool */

#define SEC_err_ext_authent_lib_interface_missing  SEC_pro_err (328)
/*$M The extensible authentication library is missing an entry point */

/* ----- transparent login error messages ----- */

#define SEC_logged_in_by_tlcred                  SEC_pro_err (400)
/*$M Not an Error (just informational) */

#define SEC_tlcred_access_error                  SEC_pro_err (401)
/*$M Unable to access (r|w|m) Transparent Login DataBase */

#define SEC_tlcred_unavailable                    SEC_pro_err (402)
/*$M User Transparent Login Credentials already deleted or do not exist*/

#define SEC_user_tlcred_unknown                  SEC_pro_err (403)
/*$M User Transparent Login Credentials are unknown */

#define SEC_tlcred_name_invalid                  SEC_pro_err (404)
/*$M User Transparent Login name does not match the one in the TL DB */

#define SEC_SLU_duplicate                        SEC_pro_err (450)
/*$M Duplicate SLU encountered */

#define SEC_SLU_hard_limit_reached              SEC_pro_err (451)
/*$M Hard limit for some SLU type reached */

#define SEC_time_not_available                  SEC_pro_err (452)
/*$M Unable to find time from the system */

#define SEC_SLU_not_found                        SEC_pro_err (453)
/*$M Unable to find the SLU record */

#define SEC_err_no_SLU_init                      SEC_pro_err (454)
/*$M SLU initialization in shared memory failed */

#define SEC_only_supported_on_NT                SEC_pro_err (455)
/*$M This entry point is only supported on NT */
```

+++ SEC - Security Services +++

```
#define SEC_UI_only_supported_on_WAL          SEC_pro_err (456)
/*$M This user interface is only supported on WAL */

#define SEC_debug_message                    SEC_pro_err (457)
/*$M Debug Message, Don't worry about it */

#define SEC_information_message              SEC_pro_err (458)
/*$M This is an information message, no errors were detected */

#define SEC_invalid_SLU_typ                  SEC_pro_err (459)
/*$M This is an invalid SLU typ*/

/* ----- single logon error messages ----- */

#define SEC_err_SLU_not_found                SEC_pro_err (500)
/*$M Unable to find the SLU record */

#define SEC_err_open_key_failed              SEC_pro_err (501)
/*$M Failed to open registration database */

#define SEC_err_query_key_failed             SEC_pro_err (502)
/*$M Failed to query registration database */

#define SEC_err_create_tls_index             SEC_pro_err (503)
/*$M Failed to create thread local storage index */

#define SEC_err_set_tls_value                SEC_pro_err (504)
/*$M Failed to set value for thread local storage */

#define SEC_err_get_tls_value                SEC_pro_err (505)
/*$M Failed to get value for thread local storage */

#define SEC_err_invalid_username             SEC_pro_err (506)
/*$M The pointer passed in for user name is NULL */

#define SEC_err_logonui                      SEC_pro_err (507)
/*$M Error returned from LogonUI */

#define SEC_err_dipl_copyright_failed        SEC_pro_err (508)
/*$M Failed to diaplay copyright information */

#define SEC_err_already_logon                SEC_pro_err (509)
/*$M User already logged on domain using different name or password doesn't match DB */

#define SEC_err_logonui_cancel               SEC_pro_err (510)
/*$M LogonUI was cancelled */

#define SEC_err_get_NT_credential            SEC_pro_err (511)
```

+++ SEC - Security Services +++

```
/*$M Failed to get the NT credential */
/* ----- single logon error messages end ----- */

#endif /* ifndef SEC_defs */
```

29. SEC SUBROUTINES

29.1. SEC_add_member

```
error_typ
SEC_add_member ( session_nbr, group_id, member_id )
ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_id_typ group_id; /* IN: (key) group id */ (pg373.32)
SEC_id_typ member_id; /* IN: member to add */ (pg373.32)
```

```
/*
This call will add a member id to a specified group id.

group_id is the group to which the member is being added. The group must
exist in order for this call to be successful.

member_id is the object which will be made a member of the specified group.
Members may be either user, groups or devices. The member specified must
exist in the object table in order for this call to be successful.
```

ERRORS:

```
SEC_err_member_add_denied
SEC_err_object_not_found
SEC_err_group_not_found
SEC_err_duplicate_group
*/
```

29.2. SEC_add_object

```
error_typ
SEC_add_object ( session_nbr, nbr_options, option_p, object_id_p )
ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
unsigned short nbr_options; /* IN: nbr of option elements */
SEC_object_opt_typ *option_p; /* IN: array of options */ (pg378.45)
SEC_id_typ *object_id_p; /* OUT: id assigned to obj */ (pg373.32)
```

```
/*
```

+++ SEC - Security Services +++

This call allows the client to create a new object within the object table. A client may specify an entire or partial listing for the addition of an object. The bare minimum information required would be an object name. That information which does not reside in the option list will be filled-in by the security service based on the class of the object being created. This default, or "fill-in" information may be managed by the system administrator. If an object class is not specified, then it will be assumed that the object being created is of the user class.

The `nbr_options` value represents the number of client specified options which will be added to the object in lieu of the default settings.

The `option_p` parameter is an array of option types and related values which will define the object being added to the security data base.

The `object_id_p` parameter is the assigned object id for the newly created object. The security service manages the assignment of object ids and this value may not be set by the client. Object ids are assigned sequentially using the last object id assigned as the starting value. An exception to this exists, however. When an object is deleted, its name, id and object class are added to the deleted table. When an object is added that has the same name as a deleted object, the deleted object information is removed from the deleted table and that old id will be assigned to the newly created object.

ERRORS:

`SEC_err_obj_add_denied`
`SEC_err_group_not_found`
`SEC_err_duplicate_object`
`SEC_err_invalid_object_filter`
`SEC_err_invalid_object_name`
`SEC_err_invalid_object_class`
`SEC_err_invalid_admin_class`
`SEC_err_invalid_admin_class_opt`
`SEC_err_invalid_device_class`
`SEC_err_invalid_admin_group_opt`
`SEC_err_invalid_dev_security`
`SEC_err_invalid_min_range`
`SEC_err_invalid_hour_range`
`SEC_err_invalid_dweek_range`
`SEC_err_invalid_success_opt`
`SEC_err_invalid_failed_opt`
`SEC_err_invalid_error_opt`
`SEC_err_invalid_log`
`SEC_err_invalid_sess_override`
`SEC_err_invalid_max_sessions`

+++ SEC - Security Services +++

```
SEC_err_invalid_comments
SEC_err_invalid_dev_class_opt
SEC_err_invalid_password_opt
SEC_err_pwd_len_out_of_range
SEC_err_no_special_char
SEC_err_invalid_admin_opt
SEC_err_invalid_max_sess_opt
*/
```

29.3. SEC_change_password

error_typ

```
SEC_change_password(session_nbr, cleartext_old_p, cleartext_new_p, object_id)
ASE_session_number_typ      session_nbr;          /* IN:  session handle
*/(pg483.24)
SEC_password_typ            cleartext_old_p;      /* IN:  old pwd, unencrypted
*/(pg371.24)
SEC_password_typ            cleartext_new_p;      /* IN:  new pwd, unencrypted
*/(pg371.24)
SEC_id_typ                   object_id;          /* IN:  uid to change pwd for
*/(pg373.32)
/*
```

This entry point allows a client to change a password for its own account or for the account of another user depending upon the administrative class of the user represented by session_nbr.

cleartext_old_p is the old cleartext password which is to be changed. If the user is of the appropriate administrative class, then this value may be ignored.

clertext_new_p is the new cleartext password which is to be implemented.

object_id is the id of the account against which the password is being changed. If object_id is equal to zero, then the current user's password will be changed.

ERRORS:

```
SEC_err_pwd_update_denied
SEC_err_object_not_found
SEC_err_invalid_password
SEC_err_no_special_char
SEC_err_invalid_pwd_length
*/
```

```
* Function: error_typ SEC_change_password *
* Purpose:  Will change a password for a user who is currently logged on. *
* * *
*****/
```

29.4. SEC_change_password_without_logon

+++ SEC - Security Services +++

```
error_typ
  SEC_change_password_without_logon(user_name_p, cleartext_old_p, cleartext_new_p,
service_name_p)
  SEC_name_typ          user_name_p;      /* IN:  name of the user
*/(pg371.25)
  SEC_password_typ      cleartext_old_p; /* IN:  old pwd, clear
*/(pg371.24)
  SEC_password_typ      cleartext_new_p; /* IN:  new pwd, clear
*/(pg371.24)
  ASE_service_name_typ   *service_name_p; /* IN:  name of security service
*/(pg486.28)
/*
  This entry point allows a client program to change the password of the
  given user without logging on to the security service. In case the user is
  logged on while calling this entry point s/he will have to log off after this
  call and log back on to have the correct credential in the cache.
  Non compliance of this condition will result in unexpected behavior.
*/
* Function: error_typ SEC_change_password_without_logon *
* Purpose:  Allows client program to check password of user who is not logged *
*           onto the system. *
* * * * *
*****/
```

29.5. SEC_check_access

```
error_typ
  SEC_check_access ( session_nbr, ar_p, wanted )
  ASE_session_number_typ session_nbr; /* IN:  session handle */ (pg483.24)
  SEC_access_restrictions_typ *ar_p; /* IN:  access restricts*/ (pg385.2)
  SEC_access_wanted_typ wanted; /* IN:  access wanted */ (pg376.25)
```

/*
This entry point will determine if the logged-on user may perform a specified action (read, write, execute/append) for the given access restriction (ar_p).

Successful access is determined by comparing the access id wanted to the expanded membership list of the user object. If an intersection exists, then this call will return success.

In the event that device security is being enforced, the access id wanted will be compared to an expanded membership list which is the result of the intersection of the device's expanded membership list and the user's expanded membership list.

ERRORS:

SEC_err_object_not_found
SEC_err_read_denied

+++ SEC - Security Services +++

```
SEC_err_write_denied
SEC_err_ax_denied
*/
```

29.6. SEC_check_membership_wid

```
error_typ
SEC_check_membership_wid ( session_nbr, group_id )
ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_id_typ group_id; /* IN: id to check */ (pg373.32)
```

```
/*
This call will identify if the logged-on user (represented through the
session number) has direct or indirect access to the specified group id.
The comparison is based on the user's transitive closure membership list
and the specified group's transitive closure membership list. If an
intersection exists, then success will be returned. The set of errors
which may occur are:
```

errors:

```
SEC_err_object_not_found
SEC_err_not_member
*/
```

29.7. SEC_cleanup_invalid_logons

```
error_typ
SEC_cleanup_invalid_logons()
```

29.8. SEC_delete_member

```
error_typ
SEC_delete_member ( session_nbr, group_id, member_id )
ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_id_typ group_id; /* IN: group id from which to del*/ (pg373.32)
SEC_id_typ member_id; /* IN: member id to del from grp */ (pg373.32)
```

```
/*
This call allows a client to remove a member from a specified group.

group_id is the group from which to remove a member.

member_id is the member to remove from the group.
```

+++ SEC - Security Services +++

ERRORS:

SEC_err_member_delete_denied
SEC_err_invalid_member_deletion
SEC_err_group_not_found

*/

29.9. SEC_delete_object

error_typ

SEC_delete_object (session_nbr, name_p)
ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_name_typ name_p; /* IN: name of object to be del */ (pg371.25)

/*

This call provides a client with the capability to delete an object from the object table. When an object is deleted, its name, id and object class are copied to the deleted table. It is possible to undelete an object from the deleted table by adding an object of the same name to the object table. As described in SEC_add_object, the new object will assume the deleted object's object id.

The name_p parameter is the name of the object to be deleted.

ERRORS:

SEC_err_obj_delete_denied
SEC_err_invalid_object_name
SEC_err_object_not_found
SEC_err_group_not_found

*/

29.10. SEC_find_groups_owning_member

error_typ

SEC_find_groups_owning_member (session_nbr, member_id, start_group_id,
max_nbr, cond_op, nbr_found_p, grplist_pp,
done_p)
ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_id_typ member_id; /* IN: (key) mbr id grps for */ (pg373.32)
SEC_id_typ start_group_id; /* IN: strt grp id key */ (pg373.32)
unsigned short max_nbr; /* IN: max nbr ids wanted */
SEC_conditional_typ cond_op; /* IN: conditional operator */ (pg376.15)
unsigned short *nbr_found_p; /* OUT: nbr grp ids found */
SEC_id_typ **grplist_pp; /* OUT: array of group ids */ (pg373.32)
bool *done_p; /* OUT: search complete? */ (pg491.22)

+++ SEC - Security Services +++

```
/*
This entry point will find up to the specified number of groups to which
a member directly belongs.

member_id is the member on which a search is to be based. A member may
be either a user, a device or a group.

start_group_id is the group at which to begin the sequential search.

max_nbr is the maximum number of groups to retrieve.

cond_op may be either SEC_EQL, SEC_GEQ or SEC_GTR.

nbr_found_p is the number of elements in the grplist_pp array.

done_p indicates that the search is completed.

ERRORS:

SEC_err_group_not_found
SEC_err_grp_read_denied
*/
```

29.11. SEC_find_members_of_group

```
error_typ
SEC_find_members_of_group ( session_nbr, group_id, start_member_id, max_nbr,
cond_op, nbr_found_p, mbrlist_pp, done_p )
ASE session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_id_typ group_id; /* IN: (key)parent group id */ (pg373.32)
SEC_id_typ start_member_id; /* IN: member to start srch from*/ (pg373.32)
unsigned short max_nbr; /* IN: max nbr items wanted */
SEC_conditional_typ cond_op; /* IN: conditinal operator */ (pg376.15)
unsigned short *nbr_found_p; /* OUT: nbr members found */
SEC_member_list_typ **mbrlist_pp; /* OUT: linked list of members */ (pg383.34)
bool *done_p; /* OUT: search complete? */ (pg491.22)

/*
```

```
This entry point allows a client to find all members directly belonging to
a specified group in the groups table.

group_id is the group for which (up to max_nbr) members are to be retrieved.

start_member_id is the member id from which to start the search.

max_nbr is the maximum number of member items to retrieve.
```

+++ SEC - Security Services +++

cond_op is the conditional operator identifying SEC_EQL, SEC_GTR or SEC_GEQ.

nbr_found_p is the number of group members found in the data base.

mbrlist_pp is the linked list of members for the specified group_id. The linked list is terminated when the next pointer is equal to nil. An SEC_free call should be performed on mbrlist_pp for each call to SEC_find_members_of_group.

done_p indicates that all search conditions have been satisfied.

ERRORS:

SEC_err_group_not_found
SEC_err_mbr_read_denied

*/

29.12. SEC_find_object

error_typ

```
SEC_find_object ( session_nbr, object_name_p, object_id, nbr_filters,
  filter_p, max_nbr, cond_op, nbr_found_p, object_pp, done_p )
  ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_name_typ object_name_p; /* IN: object name as srch key */ (pg371.25)
SEC_id_typ object_id; /* IN: object id as srch key */ (pg373.32)
unsigned short nbr_filters; /* IN: nbr of search items */
SEC_object_opt_typ *filter_p; /* IN: array of search items */ (pg378.45)
unsigned short max_nbr; /* IN: max nbr to find */
SEC_conditional_typ cond_op; /* IN: conditional operator */ (pg376.15)
unsigned short *nbr_found_p; /* OUT: nbr objects found */
SEC_object_list_typ **object_pp; /* OUT: linked list of objs */ (pg379.48)
bool *done_p; /* OUT: search complete */ (pg491.22)
```

/*

This entry point allows a client to find an object based on a variety of filters or keys. If multiple objects are requested (through the max_nbr parameter) and multiple objects are found, these will be contained in a linked list represented by object_pp. The last element in the linked list will be pointing to "nil".

Each call to the SEC_find_object entry point must eventually have an accompanying SEC_free call for the sake of freeing memory that was allocated for the requested linked list.

The three keys by which a search may be performed are the object name

+++ SEC - Security Services +++

(object_name_p), the object id (object_id), or the object class :

- (1) If an object name is provided, this will be used as the search key irregardless of the object_id value;
- (2) In order to override a name key search, the first byte of the area pointed to by object_name_p must be zero, or object_name_p must be a pointer to "nil". In this event, whatever value is contained in object_id will be used as the search key.
- (3) The search is performed by object class when the object name is NULL and the object id is zero. There are four object classes which are system, group, user and device.

To search for objects by object class, filter_p->option_type MUST be set to SEC_OPTOBJ_CLASS_KEY the first time SEC_find_object is called. This will set the security data base cursor to the beginning. When it returns from the first iteration and done_p is FALSE, filter_p->option_type must be reset to SEC_OPTOBJ_OBJECT_CLASS so the security database will not be reset to the beginning in the subsequent iterations, and only pick up from where the cursor left off from the previous call. This is due to the fact that object class is not a unique key. The search by object class increases performance and efficiency when a specific class of objects is requested.

To obtain a complete list of objects from the security data base, object_name must be NULL, object_id must be 1 and cond_op must be SEC_GTR for "greater than". The reason for object_id and cond_op to be 1 and SEC_GTR respectively is because the first valid object id in security data base starts with 2.

The nbr_filters parameter is the value which identifies how many search filters are contained in the filter_p array. Each filter may only be used once in a search. Multiple filters of the same option_type will result in unexpected behavior.

The filter_p array consists of an array of filter types and respective filter arguments through which returned objects may be filtered. The search through the data base will be based upon the provided key. Subsequently, various values may be returned if the filter is equal to the item encountered in the object table. For example, a filter value may be set to search for all objects which are of the administrative class "system". To perform a search on all elements of the object table, the object_name_p must be set to point to "nil" , the object_id must be set to a 1, and the conditional operator (cond_op) must be set to SEC_GTR. As objects are sequentially pulled from the data base, each filter type value is checked for equality to the object found in the object table. If the object meets all the filter criteria (in this case the object is of the administrative class "system"), then it will be added to the linked list.

The max_nbr value is the maximum number of objects which the requesting client wants returned. This entry point will not return more than 200 objects per call.

+++ SEC - Security Services +++

The `cond_op` (conditional operator) value is a predefined SEC value which specifies if the search should be a comparison of "equal", "greater than" or "greater than or equal".

The `nbr_found_p` value is the number of objects found (up to `max_nbr`) meeting the search criteria specified. The value limit of this parameter is bounded by its type (ie. unsigned short).

The `object_pp` is a linked list of objects which may be traversed until the next pointer in the `SEC_object_list_typ` structure is nil. The memory allocated to `object_pp` may be returned by use of the `SEC_free()` entry-point.

The `done_p` is a boolean describing if no more elements of the specified search criteria exist. The `done_p` flag will be set to TRUE if an error occurs or if the search condition has been satisfied.

ERRORS:

`SEC_err_obj_read_denied`
`SEC_err_object_not_found`
`SEC_err_invalid_object_filter`
`SEC_err_invalid_object_name`
`SEC_err_invalid_object_class`
`SEC_err_invalid_password_opt`
`SEC_err_invalid_admin_class`
`SEC_err_invalid_admin_opt`
`SEC_err_invalid_device_class`
`SEC_err_invalid_dev_security`
`SEC_err_invalid_min_range`
`SEC_err_invalid_hour_range`
`SEC_err_invalid_dweek_range`
`SEC_err_invalid_exp_time_opt`
`SEC_err_invalid_log`
`SEC_err_invalid_sess_override`
`SEC_err_invalid_max_sessions`
`SEC_err_invalid_comments`
`SEC_err_invalid_comments_opt`

*/

29.13. `SEC_find_object_without_membership`

```
error_typ
SEC_find_object_without_membership ( session_nbr, object_name_p, object_id,
nbr_filters,
filter_p, max_nbr, cond_op, nbr_found_p, object_pp, done_p )
```

+++ SEC - Security Services +++

```
ASE_session_number_typ      session_nbr;    /* IN:  session handle
*/(pg483.24)
SEC_name_typ                object_name_p; /* IN:  object name as srch key */(pg371.25)
SEC_id_typ                  object_id;          /* IN:  object id as srch key*/ (pg373.32)
unsigned short              nbr_filters;        /* IN:  nbr of search items    */
SEC_object_opt_typ         *filter_p;          /* IN:  array of search items  */(pg378.45)
unsigned short              max_nbr;           /* IN:  max nbr to find       */
SEC_conditional_typ        cond_op;           /* IN:  conditional operator   */(pg376.15)
unsigned short              *nbr_found_p;      /* OUT: nbr objects found     */
SEC_object_list_typ        **object_pp;       /* OUT: linked list of objjs  */(pg379.48)
bool                        *done_p;           /* OUT: search complete       */(pg491.22)
```

```
/*
This entry point is identical to SEC_find_object() with one exception. The linked
list of objects will not have the membership list. This is done to increase performance
in the event the client applications do not have a need for the membership list.
Otherwise, the comments for SEC_find_object() also apply to this function.
*/
```

```
*
* SEC_find_object_without_membership()
*
*****/
```

29.14. SEC_find_system_defaults

```
error_typ
SEC_find_system_defaults ( session_nbr, system_pp )
ASE_session_number_typ session_nbr;    /* IN:  session handle    */ (pg483.24)
SEC_system_desc_typ  **system_pp;     /* OUT: system defaults struct*/ (pg382.30)
```

```
/*
This entry point will return from the security data base the system
default settings for the security service from which the session_nbr
was obtained. If the entry in the data base is found, then success
will be returned. The absence of a system defaults record in the
data base may be considered a critical problem.
```

ERRORS:

```
SEC_err_system_not_found
*/
```

29.15. SEC_free

```
void
SEC_free ( list_pp )
```

+++ SEC - Security Services +++

```
any_ptr      *list_pp; /* IN: session handle */ (pg491.24)
/*
For each security call which begins with the prefix "SEC_find", a
subsequent call should be made to free the area allocated by that call.
An invalid pointer will cause SEC to force a core dump.

list_pp is the address of the pointer returned from the SEC_find_???
call.

ERRORS:
*/
```

29.16. SEC_gateway_logon

```
error_typ
SEC_gateway_logon(user_name_p, password_p, service_name_p,
slu_typ, sess_nbr_p)
SEC_name_typ    user_name_p; /* [in] */ (pg371.25)
SEC_password_typ password_p; /* [in] */ (pg371.24)
ASE_service_name_typ *service_name_p; /* [in] */ (pg486.28)
ASE_slu_typ     slu_typ; /* [in] */ (pg490.11)
ASE_session_number_typ *sess_nbr_p; /* [out] */ (pg483.24)
```

```
/* This function is an internal function used by client applications, such
* as VisualWorkflo, Panama, and Mendecino. This function is fully supported
* on the NT platform. Thread Local Storage feature needs to be implemented
* for the Unix platforms in order for other services logon function to
* work.
*
* This function logs the user on to the security service, increments the
* SLU count by one for the indicated type of SLU, and stores the results
* of this logon in *sess_nbr_p.
*
* The application is responsible for calling other service logon functions
* individually as desired using this credential.
*
* All parameters must be filled in. The returned values, provided that
* all parameters are valid, are a non-zero sess_nbr_p.
*
* This function should be called for each user the first time any thread
* or process in that application tries to access the IMS. Subsequent
* threads or processes can use the sess_nbr_p parameter to identify the
* user whose existing session is to be inherited.
*
* The object_name in service_name_p is not used and may be any value. This
* function only uses the domain name and organization to identify the
```


+++ SEC - Security Services +++

```
* security service.
*
* Any logon must be accompanied by SEC_logoff when the session is no longer
* needed.
*
* ERRORS:
* SEC_err_invalid_service_name
* SEC_err_invalid_session_nbr
* SEC_err_invalid_terminal_name
* SEC_err_invalid_username
* SEC_err_no_logon
* SEC_err_invalid_device_name
* SEC_err_object_not_found
* SEC_err_user_expired
* SEC_err_invalid_password
* SEC_err_password_expired
* SEC_err_invalid_time
* SEC_err_invalid_logon_time
* SEC_err_max_sessions
* SEC_err_max_logons_reached
* SEC_err_no_such_service
* SEC_err_bad_magic
* SEC_err_invalid_handle
*/
```

```
* Function: error_typ SEC_gateway_logon
*****/
```

29.17. SEC_get_concurrent_limits

```
error_typ
SEC_get_concurrent_limits ( logged_on_p, max_allowed_p, rejects_p, highwater_p )
    unsigned long      *logged_on_p; /* OUT: */
unsigned long      *max_allowed_p; /* OUT: */
unsigned long      *rejects_p; /* OUT: */
unsigned long      *highwater_p; /* OUT: */
/*
```

logged_on_p is the number of users currently logged-onto the system.

max_allowed_p is the maximum number of users which will be allowed to logon to the system based upon the concurrent license agreement.

rejects_p is the number of rejections which have occurred based upon reaching the concurrent license limit. This number is a running total since the last time that the FileNet software was restarted.

highwater_p is the most number of concurrent logons achieved on this

+++ SEC - Security Services +++

system.
*/

29.18. SEC_get_id

```
error_typ
  SEC_get_id ( session_nbr, id_p )
  ASE_session_number_typ session_nbr; /* IN: session handle */      (pg483.24)
SEC_id_typ      *id_p;          /* OUT: user id logged on */      (pg373.32)

/*
  This call will return the security services assigned user id for the
  logged-on user represented by session_nbr.

  ERRORS:
*/
```

29.19. SEC_get_last_logon_info

```
error_typ
  SEC_get_last_logon_info(user_name, service_name_p, action)
  SEC_name_typ      user_name;          /* [OUT] */      (pg371.25)
ASE_service_name_typ *service_name_p; /* [OUT] */      (pg486.28)
int                 action;            /* [IN] SEC_SET or SEC_GET */
```

29.20. SEC_get_message_without_logon

```
error_typ
  SEC_get_message_without_logon ( service_name_p, error, maxlen, firstline, msg_p )
  ASE_service_name_typ *service_name_p; /* IN: name of the security service */      (pg486.28)
error_typ      error;          /* IN: error number */      (pg493.26)
long           maxlen;        /* IN: length of msg_p */
bool           firstline;     /* IN: TRUE(non zero) for one line */      (pg491.22)
char           *msg_p;        /* OUT: character array for message */

/*
```

This entry point gets the message for the given error number from IMS server. The first line would be a short description of the error and the subsequent lines will have more details. if parameter firstline is set to TRUE(non zero) only the first line is returned.

If the message is longer than the length provided an error will be returned. A length of 1000 bytes for full message and 124 bytes for first line of the message is recommended.

*/

+++ SEC - Security Services +++

29.21. SEC_get_name

```
error_typ
  SEC_get_name ( session_nbr, name_p )
  ASE_session_number_typ session_nbr; /* IN: session handle */      (pg483.24)
SEC_name_typ      name_p;      /* OUT: user logged on */          (pg371.25)

/*
  This entry point returns the name of the logged-on user represented through
  the session_nbr.

  ERRORS:
*/
```

29.22. SEC_get_security_info

```
error_typ
  SEC_get_security_info ( session_nbr, name_p, id_p, info_p )
  ASE_session_number_typ session_nbr; /* IN: session handle */      (pg483.24)
SEC_name_typ      name_p;      /* IN: overrides impl handle */    (pg371.25)
SEC_id_typ      *id_p;      /* IN: overrides impl handle */      (pg373.32)
SEC_object_info_typ *info_p; /* OUT: */                          (pg388.18)

/*
  This entry point is used to obtain value for the element specified in
  SEC_object_info_typ structure.

  Note: either the name_p or the id_p can be specified. It is not necessary
  to passed in value for both of these variable.
*/
```

29.23. SEC_get_security_info_using_cache

```
error_typ
  SEC_get_security_info_using_cache(session_nbr, name_p, id_p, info_p)
  ASE_session_number_typ session_nbr; /* IN: session handle
*/(pg483.24)
SEC_name_typ      name_p; /* IN: overrides impl handle
*/(pg371.25)
SEC_id_typ      *id_p; /* IN: overrides impl handle
*/(pg373.32)
SEC_object_info_typ *info_p; /* OUT:
*/(pg388.18)

/*
  This entry point is used to obtain the SEC_object_info_typ data for an SEC
  object specified by either name or id. The information is accessed from
  the SEC cache and only results in an SEC RPC upon a cache miss.
*/
```

Note: either the name_p or the id_p can be specified. It is not necessary

+++ SEC - Security Services +++

to passed in value for both of these variable.

```
*/
* Function: error_typ SEC_get_security_info_using_cache *
* Purpose: This entry point is used to obtain the SEC_object_info_typ data *
* for an SEC object specified by either name or id. The information *
* is accessed from the SEC cache and only results in an SEC RPC *
* upon a cache miss. *
* *
* STR 52183 *
*****/
```

29.24. SEC_get_service

```
error_typ
  SEC_get_service ( session_nbr, service_name_p )
  ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
ASE_service_name_typ *service_name_p; /* OUT: security service name */ (pg486.28)
/*
  This will inform the calling client of the service name for a given
  session number.

  service_name_p is the service name of the security service for the
  given session number.

  ERRORS:

  SEC_err_system_not_found
*/
```

29.25. SEC_get_stats

```
error_typ
  SEC_get_stats ( object_name_p, service_name_p, stats_p )
  SEC_name_typ object_name_p; /* IN: name of user */ (pg371.25)
ASE_service_name_typ *service_name_p; /* IN: security service */ (pg486.28)
SEC_stats_desc_typ *stats_p; /* OUT: returned user stats */ (pg384.40)
/*
  This entry point returns the logon statistics for a given user name.
  The information contained in the returned structure may be used by a
  calling application for the purposes of warning a user that they will
  need to change their password.

  object_name_p is the name of the user for which the logon statistics are
  being requested.
```

+++ SEC - Security Services +++

stats_p is a structure containing the logon statistics for a given user.

ERRORS:

```
SEC_err_object_not_found
SEC_err_invalid_service_name
*/
```

29.26. SEC_id_to_name

```
error_typ
  SEC_id_to_name ( session_nbr, id, name_p )
  ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_id_typ      id; /* IN: object id */ (pg373.32)
SEC_name_typ    name_p; /* OUT: id related name */ (pg371.25)
/*
```

This entry point will provide a name for the provided object id. If the id is not found in the object table, then an error will be returned. The search is performed against the security service data base to which session_nbr represents a logon instance.

SEC_id_to_name() as well as SEC_name_to_id() will also cache error objects; that is, objects for which a corresponding name or id could not be found in the data base. Calls to these entry points will first search the error object cache for the requested information. If the object is found, then the error value found in the error object cache will be returned. If the object is not found, then the valid object cache will be searched. If this search fails, then SEC will perform an RPC and look for the object in the security data base. If the security data base search fails, then any available information obtained from the parameters along with the error returned from the RPC will be added to the error object cache.

The advantage of this caching method prevents applications/processes from consuming bandwidth and CPU time on repetitive search failures.

ERORRS:

```
SEC_err_object_not_found
*/
```

29.27. SEC_logoff

```
error_typ
```

+++ SEC - Security Services +++

```
SEC_logoff ( session_nbr )
ASE_session_number_typ session_nbr; /* IN: session logging off from */ (pg483.24)
```

/*

This entry point will terminate a user's logon session defined by the session number obtained from the SEC_logon() call. In the event that the process is a parent process, then this logon instance will be removed from the logon table managed by the security service. All subsequent use of this session number will cause errors to occur.

In the event that the calling process is a child process which has inherited this logon instance, the local handle for this session will be cleared, but the use of this session will still be valid for other children or parent processes.

errors:

SEC_err_no_logon

*/

29.28. SEC_logon

error_typ

```
SEC_logon ( user_name_p, password_p, terminal_p, service_name_p, session_nbr_p )
SEC_name_typ      user_name_p; /* IN: 84 char null_term name */ (pg371.25)
SEC_password_typ  password_p; /* IN: clear text password */ (pg371.24)
SEC_terminal_typ  terminal_p; /* IN: terminal name */ (pg371.26)
ASE_service_name_typ *service_name_p; /* IN: service requested */ (pg486.28)
ASE_session_number_typ *session_nbr_p; /* OUT: session nbr */ (pg483.24)
```

/* This routine logs the user on to the security service, and enables the user to access the other services in the same domain ("same domain" implies that the "organization" and "domain" of the ASE_service_name_typ records are the same). All calls to the DOC, CSM, INX, BES, WQS, or PRI modules must be done after an SEC_logon to the same domain and prior to an SEC_logoff from this domain. Attempts to access any of these modules without an applicable SEC_logon will result in an error. Note that the IMS_logon routine calls SEC_logon, so the other services can also be used between IMS_logon and IMS_logoff calls (see the IMS module).

The user name may consist of a one part (eg. "bob" or "bob:."), two part (eg. "bob:pandora" or "bob:pandora:."), or three-part (eg. "bob:pandora:FileNet") null-terminated name. The name lengths for object, domain and organization comply with FileNet standard lengths for these fields (ie. 40 characters, 20 characters and 20 characters respectively). SEC will assume the domain or organization of the service name if these items are not in the user name itself.

+++ SEC - Security Services +++

The password is a pointer to a null-terminated string with a maximum string length of eight characters.

The terminal name is a null-terminated string which may have a maximum length of 16 characters. The combination of the terminal name and the network address of the workstation executing the SEC_logon call must be a unique name.

The terminal name should not be generated from a random number/name generator. Each terminal name used will be put into the security database on the server to allow setting up terminal security (see the security administration utility), and using random terminal names would eventually cause the security database to overflow.

SEC_logon provides the option to receive a null pointer for the terminal name. In this event, SEC will create a name for the terminal.

Between the time that a client executes a successful SEC_logon and SEC_logoff, that client is considered a user of the FileNet system. The number of concurrent users is limited by the license agreement which has been purchased, and exceeding the limit results in SEC_logon returning an error and refusing the attempt.

If an application calls SEC_logon and then the workstation is rebooted prior to an SEC_logoff, the application may call SEC_logon after the reboot with the same terminal name to invalidate the first logon and not increase the count of the concurrent users on the system. This feature prevents workstation reboots from using up all of the logon slots of the concurrent user license.

Two independent applications on the same workstation should not log on using the same terminal name. If this situation occurs, the second instance of the logon will invalidate the first, and the application which did the first SEC_logon will no longer be able to successfully execute any call to the FileNet system.

A process may inherit logon information of a prior call to SEC_logon if either the parent process or the current process has called SEC_logon previously. However on NT the credential can be inherited from a peer process or parent process which has either done a SEC_logon or SEC_shared_logon.

On UNIX platform following are the conditions when inheritance will occur.

1. If user_name p, password p, terminal p, and service name p all equal to NULL, the user will be logged on to the first security service that the parent process or current process previously logged on to. The values used for user name, password, terminal, and service name

+++ SEC - Security Services +++

are the same as those used in the previous call. If no previous logon occurred, this type of call will return an error. This logon generates a "duplicate" of the session (see below).

2. If `user_name_p`, `password_p`, and `terminal_p` as NULL, but `*service_name_p` is a valid service name, and the parent process or current process has previously logged on to `*service_name_p` (and hasn't logged off), the user will be logged on to the specified service with the user name, password, and terminal being the same as on the prior call. This logon generates a "duplicate" of the session (see below).
3. If the `user_name_p`, `password_p`, and `terminal_p` are NULL, but `*service_name_p` is a valid service name, and the parent process or current process did NOT previously log on to `*service_name_p`, but did log on to some other security service, then service `*service_name_p` is logged on to, and the values used for user name, password, and terminal are the same as those used for the first logon of either the current process or parent. This logon is not a "duplicate" of the session and counts as another instance of a user for licensing considerations.
4. If `user_name_p`, `password_p`, `terminal_p`, and `service_name_p` are all non-NULL, and the current process or parent process has previously logged on using the exact same parameters, the user is logged on to the specified service and the logon creates a "duplicate" of the session.

A logon which generates a "duplicate" of a session does not count as a new concurrent user for licensing considerations. Additionally, when `SEC_logoff` is called using the duplicate handle, the parent or original logon stays in effect, and the only activity is the memory used to allocate the duplicate handle is freed up. However, if the original handle is logged off, any duplicate handles in existence become invalid and will not work for any SEC call. Therefore the handle obtained first from `SEC_logon` for a given service should not be used with `logoff` until all duplicate handles have already been logged off.

One restriction required in order to prevent inheritance from becoming ambiguous is that a process or a child of a process cannot call `SEC_logon` to the same security service twice with no intervening `logoff`, and supply different values for the user, password, or terminal. Such an attempt will be rejected with an error.

If a child process needs to logon to a security service that the parent previously logged on to, but with a different terminal name, then the child process should call `SEC_remove_inheritance` before doing `SEC_logon` call. See the description of `SEC_remove_inheritance` also in

+++ SEC - Security Services +++

this module. However, this feature is not available on WAL.

On NT platform a process will inherit the credential if it calls the SEC_logon with NULL for user_id and password, and a valid security service name; provided some peer/parent process is already logged on to the same security services via SEC_logon or SEC_shared_logon. Since NT SEC_logon does not follow parent child paradigm therefor it does not matter in which order the SEC_logoff's are called. However it is necessary that for each SEC_logon a matching SEC_logoff is called. The user remains logged on to the security services untill the last matching logoff is called.

Other restrictions also apply to prevent a user from logging-on. These consist of logon time restrictions, password violations, or device security. The set of restrictions used against a logon are bounded by the system defaults restrictions, the user defined restrictions, the device (terminal) restrictions, or the user's primary group restrictions. The determination of the intersection of this information is triggered by the system override field within each object's structure. If the system override is not set in either the user, primary group or device object, then the system default settings will be used.

The purpose of device security is to restrict access to a terminal by a user or a document. Whether or not device security is enforced is dependent upon the system override field. If the system override field is set in either the user, primary group or device object AND if the device security field is activated on either of these objects, then device security will be implemented. The opposite may also be true. If the system override field is set in either the user, primary group or device object AND the device security field is turned off in all three objects, then no device security will be applied. If no system override exists in either of these three objects, then the system default setting for device security will be used.

In the event that device security is applied upon logon, the user's logon session will be bounded by the intersection of the device's expanded membership list and the user's expanded membership list. An expanded membership list is the transitive closure of all group memberships.

A logon may also be restricted based upon a set expiration time. If either the user, device or terminal have exceeded the set expiration time within the respective object structures, then access will be denied. Through this method, it is possible to invalidate users, devices or users who belong to a specific primary group. If an expiration time is set to 0 (zero), then no expiration will apply to that object.

Another method of restricting access may be based upon logon time ranges. Logon time ranges may be set for either users, devices or primary groups, as well as in the system defaults record. A logon time range indicates

+++ SEC - Security Services +++

that a user, device or primary group (for the sake of logon) may only be used between the times specified. Whether the object specified settings or the system defaults settings are used is, again, dependent upon the system override field within each object. If either of the three objects has this value activated (set to TRUE), then a session expiration time is determined through the most restrictive set of the three objects. This expiration time is stored in memory for this user session. As subsequent RPCs are processed by the security service, this expiration time will be compared to the current time. If a point is reached in which this time has been exceeded, then RPCs for this logon session will be rejected, causing the user to be effectively logged-off.

ERRORS:

SEC_err_no_logon
SEC_err_duplicate_logon
SEC_err_too_many_term_ids
SEC_err_invalid_device_name
SEC_err_object_not_found
SEC_err_not_user
SEC_err_user_expired
SEC_err_device_expired
SEC_err_group_expired
SEC_err_invalid_password
SEC_err_password_expired
SEC_err_device_access_denied
SEC_err_invalid_time
SEC_err_invalid_logon_time
SEC_err_max_sessions
SEC_err_max_logons_reached

*/

29.29. SEC_name_to_id

error_typ

SEC_name_to_id (session_nbr, name_p, id_p)
ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_name_typ name_p; /* IN: object name */ (pg371.25)
SEC_id_typ *id_p; /* OUT: name related id */ (pg373.32)

/*

This entry point will provide an id for a given object name. If the name is not found in the object table, then an error will be returned. The searched is performed against the security service data base to which session_nbr represents a logon instance.

For a discussion on error object caching see the explanation under the

+++ SEC - Security Services +++

SEC_id_to_name() entry point.

ERRORS:

SEC_err_object_not_found
*/

29.30. SEC_pooled_logon

error_typ

SEC_pooled_logon(user_name_p,
password_p,
service_name_p,
slu_typ,
slu_policy,
sess_nbr_p,
sess_guid_p,
sess_ref_count_p)

SEC_name_typ	user_name_p;	/* [in]	*/	(pg371.25)
SEC_password_typ	password_p;	/* [in]	*/	(pg371.24)
ASE_service_name_typ	*service_name_p;	/* [in]	*/	(pg486.28)
ASE_slu_typ	slu_typ;	/* [in]	*/	(pg490.11)
SEC_slu_accounting_typ	slu_policy;	/* [in]	*/	(pg386.28)
ASE_session_number_typ	*sess_nbr_p;	/* [out]	*/	(pg483.24)
SEC_guid_typ	*sess_guid_p;	/* [out]	*/	(pg386.26)
unsigned long	*sess_ref_count_p;	/* [out]	*/	

/*
* This function is only supported on NT platform.
*
* This function logs the user on to the security service. Based on slu_typ,
* it may or may not increment the SLU count.
*
* The 3 possible values for slu_type are:
* SEC_SLU_NO_POOLING - increments SLU count
* SEC_SLU_ACCOUNTING_STRICT - increments SLU count
* SEC_SLU_ACCOUNTING_APPROX - SLU count remains same
*
* The returning handle of this logon is stored in *sess_nbr_p
*
* The application is responsible for calling other service logon functions
* individually as desired using this credential.
*
* All parameters must be filled in. The returned values, provided that
* all parameters are valid, are a non-zero sess_nbr_p, a session GUID and
* a reference count.

+++ SEC - Security Services +++

```
*
* This function should be called for each user the first time any thread
* or process in that application tries to access the IMS. Subsequent
* threads or processes can use the sess_nbr_p parameter to identify the
* user whose existing session is to be inherited.
*
* The object_name in service_name_p is not used and may be any value. This
* function only uses the domain name and organization to identify the
* security service.
*
* When slu_policy is SEC_SLU_ACCOUNTING_STRICT, a new handle is created and
* added to the pooled hash table for reuse. When slu_policy is
* SEC_SLU_ACCOUNTING_APPROX, this function first tries to locate a reusable
* handle from the pooled hash table. If a reusable handle is found, that handle
* is returned to the caller. If a reusable handle is not found, the function
* will create a new handle and add the newly created handle to the pool.
* When slu_policy is SEC_SLU_NO_POOLING, a new handle is created but it is
* not available for pooling. Calling SEC_pooled_logon() with SEC_SLU_NO_POOLING
* is the same as calling SEC_gateway_logon().
*
* If username and password are both NULL, unified logon will be tried as the
* last alternative if everything else failed.
*
* A SEC_logoff() should be called for each unique security handle.
*
* ERRORS:
* SEC_err_invalid_service_name
* SEC_err_invalid_session_nbr
* SEC_err_invalid_terminal_name
* SEC_err_invalid_username
* SEC_err_no_logon
* SEC_err_invalid_device_name
* SEC_err_object_not_found
* SEC_err_user_expired
* SEC_err_invalid_password
* SEC_err_password_expired
* SEC_err_invalid_time
* SEC_err_invalid_logon_time
* SEC_err_max_sessions
* SEC_err_max_logons_reached
* SEC_err_no_such_service
* SEC_err_bad_magic
* SEC_err_invalid_handle
*/
* Function: SEC_pooled_logon()
* Purpose: This function logs the user on to the security service and, based
* on slu_typ, it may or may not increment the SLU count.
```

+++ SEC - Security Services +++

*
*****/

29.31. SEC_shared_logon

error_typ

```
SEC_shared_logon(user_name, password, service_name, slu_typ, flags, sess_nbr)
SEC_name_typ      user_name;      /* [IN] */      (pg371.25)
SEC_password_typ  password;      /* [IN] */      (pg371.24)
ASE_service_name_typ *service_name; /* [IN] */      (pg486.28)
ASE_slu_typ       slu_typ;        /* [IN] */      (pg490.11)
unsigned long     flags;          /* [IN] */
ASE_session_number_typ *sess_nbr; /* [OUT] */      (pg483.24)
```

```
/* This routine is currently supported on the NT platform only.
*
* This routine logs the user on to the security service. It establishes
* a session with the IMS for each unique combination of user name, slu type,
* and target domain. The application is responsible for calling other
* service logon functions, such as DOC_logon, individually as desired using
* this credential.
*
* The user may not log on to the same domain with two different IMS user
* names. The user may, however, log on to two different domains with two
* different user names, or two different domains with the same user name.
*
* The user name, password, and domain parameters may be NULL. This
* routine will try to determine the default values for these parameters
* from the environment. If defaults cannot be determined from the environemnt
* and the FN_DONT_SHOW_UI is not set, then a UI dialog will be displayed
* asking the user for the logon parameters. If the FN_DONT_SHOW_UI is set,
* then a default which cannot be determined from the environment will cause
* an error to be returned from this routine.
*
* The algorithm used to determine the defaults for NULL values, and the point
* at which the UI dialog is displayed, if allowed, is as follows:
*
* If service_name is NULL:
*
* 1. The registry is searched for a default domain. If found, the default
* domain is used. But if not found, then the logon UI is displayed
* (if flags is set to FN_SHOW_UI).
*
* The search order for the registry key is as follows:
*
* HKEY_CURRENT_USER/Software/FileNet/WAL/CurrentVersion
```

+++ SEC - Security Services +++

* HKEY_USER/Software/FileNet/WAL/CurrentVersion
*
* The value name for the key is domain and contains domain:organization
* string.
*
* 2. When the UI dialog is displayed, the domain is allowed to be entered if
* the domain field is blank. The domain field may not be changed if a
* non-blank name is given and the UI is being displayed to get some other
* piece of logon parameters.
*
* If user_name is NULL:
*
* 1. The credential cache is searched for a session with the given
* domain and sltu type. If any other processes with the same native
* user have done a FileNet logon to the same domain and with the same sltu
* type, and one and only one FileNet user name has been used with this
* domain, then this user's name will be used for the current logon.
*
* 2. If no user has been previously logged on and if unified logon is setup
* in the registry, the native user name combined with the domain name
* (above) is used as the FileNet user name.
*
* [NOTE: the native user name means the user associated with the
* current process according to host operating system. The native name is
* used as the FileNet user name.]
*
* 3. If no user has been previously logged on with the current domain and if
* unified logon is not setup, then the logon UI dialog is displayed (if
* allowed) with a blank user name.
*
* If username and password are both NULL, unified logon will be tried
* as the last alternative if everything else failed.
*
* If the logon UI is displayed with either a blank or non-blank user name,
* and/or password, the user name and/or password may be changed via the UI.
*
* If the session establishment fails due to invalid user name, password,
* or domain, then the logon UI will reappear (if allowed), display the error,
* and let the user retry. The logon UI will not appear on an error if the
* FN_WITHOUT_UI flag is set. Instead, the error will be returned to the
* application.
*
* The value for the 'flags' parameter is either FN_SHOW_UI or
* FN_DONT_SHOW_UI. A background process might set the flags to
* FN_DONT_SHOW_UI to disallow the display of the logon UI.
*
* It is expected that most interactive applications will just pass in a SLU
* type and domain, and that the user name and password will either be

+++ SEC - Security Services +++

* unnecessary because the user already has an established session, or will
* be obtained via the logon UI function.
*
* The "object" field in the domain name is not used. Only the "organization"
* and the "domain" fields need to be set.
*
* If a call to SEC_shared_logon is made and the user, password, domain,
* and SLU_id have not been seen before, the SLU count for the indicated
* SLU type is incremented. Conversely, if the user, password, domain,
* and SLU_id have already been logged on, the SLU count will not be
* incremented. If a call to SEC_logoff is made and the last use of a
* user, password, domain, SLU_id is logged off, the SLU usage count will
* be decremented for the SLU type.
*
* Any logon must be accompanied by a logoff when the session is no longer
* needed. No logoff done by one process or thread will affect the logon
* done by another process or thread.
*
* ERRORS:
* SEC_err_invalid_service_name
* SEC_err_bad_service
* SEC_err_open_key_failed
* SEC_err_query_key_failed
* SEC_err_no_logon
* SEC_err_duplicate_logon
* SEC_err_too_many_term_ids
* SEC_err_invalid_device_name
* SEC_err_object_not_found
* SEC_err_not_user
* SEC_err_user_expired
* SEC_err_device_expired
* SEC_err_group_expired
* SEC_err_invalid_password
* SEC_err_password_expired
* SEC_err_device_access_denied
* SEC_err_invalid_time
* SEC_err_invalid_logon_time
* SEC_err_max_sessions
* SEC_err_maximum_logons_reached
* SEC_err_already_logon
* SEC_err_logonui
* SEC_err_invalid_domain
* SEC_err_invalid_name
* SEC_err_SLU_not_found
*
*/

* Function: error_typ SEC_shared_logon

+++ SEC - Security Services +++

* NOTE: This function is only supported on NT.
*****/

29.32. SEC_update_object

```
error_typ
SEC_update_object ( session_nbr, object_id, nbr_options, option_p )
ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
SEC_id_typ object_id; /* IN: (key)obj id to update */ (pg373.32)
unsigned short nbr_options; /* IN: nbr of options */
SEC_object_opt_typ *option_p; /* IN: option list for change*/ (pg378.45)
```

/*
This call allows a client to modify the various values of an object.

The object_id is the key value of the object to be changed.

The nbr_options is the number of elements in the array described by option_p.

The option_p parameter is an array of option types and related values which will be modified for the specified object in the security data base.

ERRORS:

- SEC_err_update_obj_denied
- SEC_err_obj_add_denied
- SEC_err_group_not_found
- SEC_err_duplicate_object
- SEC_err_invalid_object_filter
- SEC_err_invalid_object_name
- SEC_err_invalid_object_class
- SEC_err_invalid_admin_class
- SEC_err_invalid_admin_class_opt
- SEC_err_invalid_device_class
- SEC_err_invalid_admin_group_opt
- SEC_err_invalid_dev_security
- SEC_err_invalid_min_range
- SEC_err_invalid_hour_range
- SEC_err_invalid_dweek_range
- SEC_err_invalid_success_opt
- SEC_err_invalid_failed_opt
- SEC_err_invalid_error_opt
- SEC_err_invalid_log
- SEC_err_invalid_sess_override
- SEC_err_invalid_max_sessions
- SEC_err_invalid_comments

+++ SEC - Security Services +++

```
SEC_err_invalid_dev_class_opt
SEC_err_invalid_password_opt
SEC_err_pwd_len_out_of_range
SEC_err_no_special_char
SEC_err_invalid_admin_opt
SEC_err_invalid_max_sess_opt
*/
```

29.33. SEC_update_system_defaults

error_typ

```
SEC_update_system_defaults ( session_nbr, nbr_options, option_p )
    ASE_session_number_typ session_nbr; /* IN: session handle */ (pg483.24)
unsigned short          nbr_options; /* IN: nbr of options */
SEC_system_opt_typ     *option_p; /* IN: options to change*/ (pg382.4)
```

/*

This call allows a client to change the values of the system defaults table entry. The option_p parameter is an array of option types and the respective values for update. The nbr_options parameter represents the number of elements in the option_p array.

The only administrator which may alter the system-wide settings is the SysAdmin user.

ERRORS:

```
SEC_err_access_denied
SEC_err_system_not_found
SEC_err_invalid_system_options
SEC_err_invalid_dev_security
SEC_err_invalid_func_def
SEC_err_invalid_max_sessions
SEC_err_invalid_pwd_spec_char
SEC_err_invalid_pwd_min_length
SEC_err_invalid_pwd_renewal_days
SEC_err_invalid_grace_period
SEC_err_invalid_pwd_attempts
SEC_err_invalid_failure_mins
SEC_err_invalid_log
SEC_err_invalid_min_range
SEC_err_invalid_hour_range
SEC_err_invalid_dweek_range
SEC_err_invalid_language
SEC_err_invalid_system_filter
*/
```

+++ SEC - Security Services +++

29.34. SEC_validate_session

error_typ

SEC_validate_session(session_nbr)

ASE_session_number_typ session_nbr; /* IN: session nbr */

(pg483.24)

/* This function determines if a session is valid. A valid session implies a valid logon.

ERRORS:

SEC_err_invalid_handle

SEC_err_invalid_session_nbr

SEC_err_stale_session

SEC_err_no_logon

SEC_err_invalid_logon_time

*/

30. Sqi DECLARATIONS

Sqi

SQL Services provides its clients with access to various relational database. It does so by using SQL (Structured Query Language) as the interface language. Users of Sqi are should be familiar with SQL before using Sqi.

Sqi entry points have a number of common features:

session handle: a session handle is created by logging on to Sqi. each entry point for the session receives the handle as a parameter. The logon and logoff entry points receive a pointer to the handle to return or zero it respectively.

cursor handle: Sqi_create_cursor and Sqi_prepare_blob_update create cursor handles. All related calls, such as Sqi_bind_value, Sqi_bind_result, and Sqi_exec_cursor receive the cursor handle as a parameter (in addition to the session handle) to identify the SQL operator to which they apply.

sqlca: the SQL Communication Area is common to many calls. It returns error information about the call as follows:

Each client application must allocate its own Sqi_sqlca.

```
typedef struct
{ char          sqlcaid [8];
  long          sqlcabc;
  error_typ     sqlcode;
  short         sqlerrml;
  char          sqlerrmc [70];
  char          sqlerrp [8];
  long          sqlerrd [6];
  struct
  { char        sqlwarn0;
    char        sqlwarn1;
    char        sqlwarn2;
    char        sqlwarn3;
```

+++ SQI - SQL Services +++

```
        char    sqlwarn4;  
        char    sqlwarn5;  
        char    sqlwarn6;  
        char    sqlwarn7;  
    } sqlwarn;  
    char        sqltext [8];  
} SQI_sqlca_typ
```

sqlcaid is a null-terminated string containing "SQLCA ".

sqlcabc contains the length of the SQLCA.

sqlcode contains the SUCCESS error tuple for a successful operation. Otherwise, it contains either a DBMS message or an SQI db message. The error tuple in sqlcode is also used as the return code to the calling client application.

sqlerrml is the length of sqlerrmc.

sqlerrmc contains the error descriptions.

sqlerrp contains the DBMS diagnostic information.

sqlerrd is an integer array of 6 elements. sqlerrd[3] shows the number of rows affected after an INSERT, UPDATE, or DELETE.

sqlwarn [0] is blank if all other indicators are blank. It contains 'W' if at least one other indicator contains 'W'.

sqlwarn [1] contains 'W' if a string column was truncated when assigned to a host variable.

sqlwarn [2] contains 'W' if null values were eliminated from the argument of a function.

sqlwarn [3] contains 'W' if the number of columns is larger than the number of host variables.

sqlwarn [4] contains 'W' if a prepared UPDATE or DELETE statement did not include a WHERE clause.

sqlwarn [5] contains 'W' if one or more output fields is null, underflowed, overflowed, or can't be converted into the designated type.

sqlwarn [6] is DBMS dependent.

sqlwarn [7] is DBMS dependent.

+++ Sqi - SQL Services +++

sqltext is not used.

```
/*
 *   File : Sqi.defs
 *   $Revision: 1.59 $ $Date: 08/10/08 14:02:43 $
 *
 *   This file contains the constants, typedefs, #defines, and error
 *   tuples used by the Sqi service.
 *
 *   Client applications should include this file if they use Sqi.
 */

#ifndef SQIDEFS
#define SQIDEFS

#include <ErrEncode.h>
#include <AS_externals.h>

#define Sqi_SQLCA_ID      "SQLCA  "
#define Sqi_SQLDA_ID     "SQLDA  "
#define Sqi_SQL_WARNING  'W'

#define Sqi_GOOD_RESULT_INDICATOR      0
#define Sqi_NULL_INDICATOR            -1
#define Sqi_CONVERSION_ERROR_INDICATOR -2

typedef ASE_session_number_typ Sqi_session_handle_typ; /* session handle */ (pg483.24)
typedef unsigned long          Sqi_cursor_handle_typ; /* cursor handle */

/*
 *   DBMS identification
 */
typedef unsigned long          Sqi_dbms_typ; /* Which DBMS we are using */
#define Sqi_dbms_oracle 1
#define Sqi_dbms_sybase 2
#define Sqi_dbms_allbase 3
#define Sqi_dbms_mssql 4
#define Sqi_dbms_db2 5 /* DB2 Support */

typedef struct
{
    unsigned long ver_major; /* Major release number */
    unsigned long ver_minor; /* Minor release number (if available) */
} Sqi_dbms_version_typ;

typedef struct
```

+++ Sqi - SQL Services +++

```

{
  Sqi_dbms_typ      dbms_type;      /* Which dbms */                (pg439.35)
  Sqi_dbms_version_typ ver_current; /* version in use */            (pg439.46)
  Sqi_dbms_version_typ ver_oldest; /* oldest version we know about */ (pg439.46)
  Sqi_dbms_version_typ ver_newest; /* newest version we know about */ (pg439.46)
} Sqi_dbms_info_typ;

/*
  Data types in Sqi
*/

#define Sqi_boolean      1      /* TRUE or FALSE */
#define Sqi_byte         2      /* signed two's complement 8 bit quantity */
#define Sqi_unsigned_byte 3     /* unsigned 8 bit quantity */
#define Sqi_short        4      /* signed two's complement 16 bit quantity */
#define Sqi_unsigned_short 5    /* unsigned 16 bit quantity */
#define Sqi_long         6      /* signed two's complement 32 bit quantity */
#define Sqi_unsigned_long 7     /* unsigned 32 bit quantity */
#define Sqi_FPnum        8      /* FileNet floating point number */
#define Sqi_ASCII        9      /* ASCII string data ( null terminating ) */
#define Sqi_date         10     /* FileNet encoded date */
#define Sqi_time         11     /* FileNet encoded date and time */
#define Sqi_menu         12     /* integer values encoding string */
#define Sqi_text_object  13     /* lots of characters, with length */
#define Sqi_binary_object 14    /* lots of bytes (BLOB), with length */
#define Sqi_null         20     /* NULL */
typedef short Sqi_data_type_typ;

/*
  Type declarations for
  SQLCA ( SQL communication area )
  SQLDA ( SQL description area )
*/

/*
  Maximum sizes of various fields in SQLCA
*/
#define Sqi_MAX_SQL_MSG_SIZE 70
#define Sqi_MAX_SQL_ID_SIZE 8
#define Sqi_MAX_SQL_ERRP_SIZE 8
#define Sqi_MAX_SQL_ERRD_SIZE 6
#define Sqi_MAX_SQL_EXT_SIZE 8

#define Sqi_MAX_SQL_NAME_SIZE 30 /* Max name size of column/expression */

typedef struct
{
  char      sqlcaid [Sqi_MAX_SQL_ID_SIZE];
  long      sqlcabc;

```

+++ Sqi - SQL Services +++

```

error_typ      sqlcode;                                (pg493.26)
short          sqlerrml;
char           sqlerrmc [SQI_MAX_SQL_MSG_SIZE + 1 ];
char           sqlerrp  [SQI_MAX_SQL_ERRP_SIZE + 1];
long          sqlerrd  [SQI_MAX_SQL_ERRD_SIZE];
struct        { char      sqlwarn0;
               char      sqlwarn1;
               char      sqlwarn2;
               char      sqlwarn3;
               char      sqlwarn4;
               char      sqlwarn5;
               char      sqlwarn6;
               char      sqlwarn7;
               } sqlwarn;
char          sqlnext [SQI_MAX_SQL_EXT_SIZE + 1];
} Sqi_sqlca_typ;

```

```

typedef struct
{ Sqi_data_type_typ sqltype;                            (pg440.27)
  short             sqlllen;
  char              *sqldata;
  short            *sqlcind;
  char             sqlname [SQI_MAX_SQL_NAME_SIZE + 1];
  char             unused; /* make the structure even size */
} Sqi_sqlvar_typ;

```

```

typedef struct
{ char             sqldaaid [SQI_MAX_SQL_ID_SIZE];
  long            sqldabc;
  short          sqln;
  short          sqld;
  Sqi_sqlvar_typ sqlvar [1];                            (pg441.26)
} Sqi_sqlda_typ;

```

```

/* Data structure for passing/receiving blobs.
 * Since other length fields are only two bytes,
 * this four byte length field allows for objects > 64KB
 */

```

```

typedef struct
{
  long          bl_buf_len; /* size of bl_data array */
  long          bl_len; /* length of contents of bl_data */
  unsigned char bl_data[1]; /* Actual blob data */
} Sqi_blob_typ;

```

```

#define Sqi_BLOB_DATA_OFFSET ((int)((Sqi_blob_typ *) (0)) -> bl_data)

```

+++ SQI - SQL Services +++

```
#define SQI_BLOB_ALLOC_LEN(l) (l + SQI_BLOB_DATA_OFFSET )
#define SQI_BLOB_TYP_LEN(bp) (bp->bl_len + SQI_BLOB_DATA_OFFSET)
#define SQI_BLOB_TYPE(t) (t == SQI_text_object || t == SQI_binary_object)
#define SQI_DEFAULT_BLOB_BUF 4096 /* Not too big, by default */
```

```
/* We have to allocate something in the row buffers for blob columns,
   but the real data is found indirectly. This value is used to
   specify the fake size of the blob column in the internal row buffer */
#define SQI_DUMMY_BLOB_LEN 4
```

```
/*
   Errors
*/
```

```
/*
   function byte in error tuples
*/
```

```
#define SQI_COR_MSG_FUNC_CODE 0
#define SQI_DB_MSG_FUNC_CODE 1
#define SQI_INT_MSG_FUNC_CODE 2
```

```
/*
   Definitions for SQI Error Returns.
*/
```

```
/*
   SQI operational messages - defined in Courier protocol.
*/
```

```
#define SQI_OTHER_ERROR 1
#define SQI_CANCELLED 2
#define SQI_NO_RESOURCES 3
#define SQI_DB_SERVER_NOT_AVAIL 4
#define SQI_BAD_USER 5
#define SQI_INVALID_SESSION_HANDLE 6
#define SQI_INVALID_CURSOR_HANDLE 7
#define SQI_INVALID_DATA_TYPE 8
#define SQI_INVALID_BIND_VALUE 9
#define SQI_INVALID_BIND_RESULT 10
#define SQI_BIND_RESULT_MISSING 11
#define SQI_BAD_DB_VERSION 12
#define SQI_FETCH_WITHOUT_EXECUTION 13
#define SQI_VALUE_OUT_OF_RANGE 14
#define SQI_DATA_TYPE_NOT_SUPPORTED 15
#define SQI_LOGON_TWICE 16
#define SQI_NO_CURRENT_QUERY 17
```


+++ SQI - SQL Services +++

```
#define  SQI_NO_LICENSE          18

/*
  SQI operational message - not Courier defined.
*/

#define  SQI_SQLCA_MISSING      100
#define  SQI_SQLDA_MISSING      101
#define  SQI_SQL_SERVICE_NOT_AVAIL  102
#define  SQI_INT_INVALID_FIELD_SIZE  103
#define  SQI_INVALID_MESSAGE      104
#define  SQI_INVALID_PROCEDURE_CALL  105
#define  SQI_CONNECTION_BROKEN     106
#define  SQI_NO_CONFIG_FILE        107
#define  SQI_POS_CONFIG_FILE        108
#define  SQI_INVALID_FIELD_POS      109
#define  SQI_INVALID_SQLDA_SIZE     110
#define  SQI_NOT_A_SELECT_QUERY     111
#define  SQI_BAD_VERSION            112
#define  SQI_NO_SUCH_DBMS           113
#define  SQI_INVALID_LOGON_CALL     114
#define  SQI_USER_NOT_ALLOWED       115

/*
  SQI db messages
*/

#define  SQI_INVALID_COLUMN_NAME    704
#define  SQI_SYNTAX_ERROR           900
#define  SQI_INVALID_TABLE_NAME     942
#define  SQI_TABLE_ALREADY_EXIST    955
#define  SQI_DUPLICATE_COLUMN_NAME  957
#define  SQI_END_OF_FETCH            1403
#define  SQI_OUTPUT_WARNING         1405
#define  SQI_DB_CANCELLED           9999

/*
  Error tuples
*/

#define  SQI_cor_msg(x)      err_encode( err_SQI, SQI_COR_MSG_FUNC_CODE, x )
#define  SQI_int_msg(x)     err_encode( err_SQI, SQI_INT_MSG_FUNC_CODE, x )
#define  SQI_db_msg(x)      err_encode( err_SQI, SQI_DB_MSG_FUNC_CODE, x )

/* Operational messages that are defined in the Courier protocol of SQI. */
#define  SQI_ERR_CANCELLED          SQI_cor_msg(SQI_CANCELLED)
```

+++ SQI - SQL Services +++

/*\$M Command cancelled */

#define SQI_ERR_NO_RESOURCES SQI_cor_msg(SQI_NO_RESOURCES)
/*\$M No resources */

#define SQI_ERR_DB_SERVER_NOT_AVAIL SQI_cor_msg(SQI_DB_SERVER_NOT_AVAIL)
/*\$M Requested server is not available */

#define SQI_ERR_BAD_USER SQI_cor_msg(SQI_BAD_USER)
/*\$M No such user or password in database */

#define SQI_ERR_INVALID_SESSION_HANDLE SQI_cor_msg(SQI_INVALID_SESSION_HANDLE)
/*\$M Invalid session handle */

#define SQI_ERR_INVALID_CURSOR_HANDLE SQI_cor_msg(SQI_INVALID_CURSOR_HANDLE)
/*\$M Invalid cursor handle */

#define SQI_ERR_DATA_TYPE_NOT_SUPPORTED SQI_cor_msg(SQI_DATA_TYPE_NOT_SUPPORTED)
/*\$M Data type specified is not supported */

#define SQI_ERR_INVALID_BIND_VALUE SQI_cor_msg(SQI_INVALID_BIND_VALUE)
/*\$M Invalid bind value specification */

#define SQI_ERR_INVALID_BIND_RESULT SQI_cor_msg(SQI_INVALID_BIND_RESULT)
/*\$M Invalid bind result specification */

#define SQI_ERR_BIND_RESULT_MISSING SQI_cor_msg(SQI_BIND_RESULT_MISSING)
/*\$M bind result(s) is expected */

#define SQI_ERR_BAD_DB_VERSION SQI_cor_msg(SQI_BAD_DB_VERSION)
/*\$M Desired version of the database is not available */

#define SQI_ERR_FETCH_WITHOUT_EXECUTION SQI_cor_msg(SQI_FETCH_WITHOUT_EXECUTION)
/*\$M The query must be executed before rows can be fetched */

#define SQI_ERR_VALUE_OUT_OF_RANGE SQI_cor_msg(SQI_VALUE_OUT_OF_RANGE)
/*\$M Value specified is out of the range of the underlying data type */

#define SQI_ERR_INVALID_DATA_TYPE SQI_cor_msg(SQI_INVALID_DATA_TYPE)
/*\$M Invalid data type */

#define SQI_ERR_LOGON_TWICE SQI_cor_msg(SQI_LOGON_TWICE)
/*\$M User has already logged on */

#define SQI_ERR_NO_CURRENT_QUERY SQI_cor_msg(SQI_NO_CURRENT_QUERY)
/*\$M No current query to cancel */

#define SQI_ERR_NO_LICENSE SQI_cor_msg(SQI_NO_LICENSE)

+++ SQI - SQL Services +++

```
/*$M Server is not licensed for requested operation */

/* Operational messages that are not defined in Courier protocol of SQI. */

#define SQI_ERR_SQLCA_MISSING SQI_int_msg(SQI_SQLCA_MISSING)
/*$M SQLCA area is missing */

#define SQI_ERR_SQLDA_MISSING SQI_int_msg(SQI_SQLDA_MISSING)
/*$M SQLDA area is missing */

#define SQI_ERR_SQL_SERVICE_NOT_AVAIL SQI_int_msg(SQI_SQL_SERVICE_NOT_AVAIL)
/*$M The requested SQL Service is not available */

#define SQI_ERR_INT_INVALID_FIELD_SIZE SQI_int_msg(SQI_INT_INVALID_FIELD_SIZE)
/*$M Invalid field size */

#define SQI_ERR_INVALID_MESSAGE SQI_int_msg(SQI_INVALID_MESSAGE)
/*$M Invalid network message */

#define SQI_ERR_INVALID_PROCEDURE_CALL SQI_int_msg(SQI_INVALID_PROCEDURE_CALL)
/*$M Invalid procedure call */

#define SQI_ERR_CONNECTION_BROKEN SQI_int_msg(SQI_CONNECTION_BROKEN)
/*$M Connection broken */

#define SQI_ERR_NO_CONFIG_FILE SQI_int_msg(SQI_NO_CONFIG_FILE)
/*$M Configuration file not found */

#define SQI_ERR_POS_CONFIG_FILE SQI_int_msg(SQI_POS_CONFIG_FILE)
/*$M Bad configuration file */

#define SQI_ERR_INVALID_FIELD_POS SQI_int_msg(SQI_INVALID_FIELD_POS)
/*$M Invalid output field position */

#define SQI_ERR_INVALID_SQLDA_SIZE SQI_int_msg(SQI_INVALID_SQLDA_SIZE)
/*$M Invalid SQLDA area size */

#define SQI_ERR_NOT_A_SELECT_QUERY SQI_int_msg(SQI_NOT_A_SELECT_QUERY)
/*$M The query is not for retrieval */

#define SQI_ERR_BAD_VERSION SQI_int_msg(SQI_BAD_VERSION)
/*$M Attempt to link to SQI with an invalid version number */

#define SQI_ERR_NO_SUCH_DBMS SQI_int_msg(SQI_NO_SUCH_DBMS)
/*$M DBMS requested in logon call is not supported */

#define SQI_ERR_INVALID_LOGON_CALL SQI_int_msg(SQI_INVALID_LOGON_CALL)
```

+++ SQI - SQL Services +++

```
/*$M Invalid rpc number in serialize logon */

#define SQI_ERR_USER_NOT_ALLOWED      SQI_int_msg(SQI_USER_NOT_ALLOWED)
/*$M Not allowed to map to reserved db user name */

/* SQI db messages . */

#define SQI_ERR_INVALID_COLUMN_NAME   SQI_db_msg(SQI_INVALID_COLUMN_NAME)
/*$M Invalid column name */

#define SQI_ERR_SYNTAX_ERROR          SQI_db_msg(SQI_SYNTAX_ERROR)
/*$M Syntax error */

#define SQI_ERR_INVALID_TABLE_NAME    SQI_db_msg(SQI_INVALID_TABLE_NAME)
/*$M Invalid table name */

#define SQI_ERR_TABLE_ALREADY_EXIST   SQI_db_msg(SQI_TABLE_ALREADY_EXIST)
/*$M Table already exists */

#define SQI_ERR_DUPLICATE_COLUMN_NAME SQI_db_msg(SQI_DUPLICATE_COLUMN_NAME)
/*$M Duplicate column name */

#define SQI_ERR_END_OF_FETCH          SQI_db_msg(SQI_END_OF_FETCH)
/*$M End of fetch */

#define SQI_ERR_OUTPUT_WARNING        SQI_db_msg(SQI_OUTPUT_WARNING)
/*$M Some of the indicators of the output fields are set */

#define SQI_ERR_DB_CANCELLED          SQI_db_msg(SQI_DB_CANCELLED)
/*$M Command cancelled */

#endif /* SQIDEFS */
```

31. SQI SUBROUTINES

31.1. SQI_begin_transaction

```
error_typ
SQI_begin_transaction(handle, sqlca_p)
    ASE_session_number_typ  handle; /* IN: session handle */           (pg483.24)
    SQI_sqlca_typ          *sqlca_p; /* OUT: standard sqlca */       (pg441.16)
/*
```

SQI_begin_transaction is used to tell a DBMS that the client wants to begin a new transaction. this will be interpreted as a NO-OP (no operation) for a DBMS that does not support begin transaction.

+++ SQI - SQL Services +++

ERRORS:
database errors

*/

31.2. SQI_bind_result

error_typ

```
SQI_bind_result(handle, ch, nth, buf_p, buf_type, buf_size, ind_p, sqlca_p)
  ASE_session_number_typ  handle; /* IN: session handle */ (pg483.24)
  SQI_cursor_handle_typ   ch;     /* IN: handle of cursor holding SQL */ (pg439.30)
  unsigned short         nth;     /* IN: position of bound result */
  char                   *buf_p; /* IN: address receiving data */
  SQI_data_type_typ      buf_type; /* IN: type of variable being bound */ (pg440.27)
  short                 buf_size; /* IN: size in bytes of buffer */
  short                 *ind_p; /* IN: address of indicator variable */
  SQI_sqlca_typ          *sqlca_p; /* OUT: standard sqlca */ (pg441.16)
```

/*

SQI_bind_result is used to define an output buffer for an expression in a SELECT list within an SQL statement. It is also termed "output binding".

The SELECT SQL statement must have been parsed with the use of SQI_create_cursor. The output buffer defined is filled with the expression value by subsequent SQI_fetch calls.

Output binds can be redone during a fetch sequence, resulting in data for different rows being received in different variables.

For special considerations on binding blob data, see SQI_prepare_blob_update.

The numbering for "nth" begins at one.

Indicator variables, which are optional, are returned as follows:

zero: successful fetch
negative: expression is NULL
positive: expression truncated, indicator is actual length

ERRORS:

SQI_ERR_INVALID_DATA_TYPE
SQI_ERR_INVALID_BIND_RESULT
SQI_ERR_INVALID_FIELD_POS
SQI_ERR_NOT_A_SELECT_QUERY

*/

31.3. SQI_bind_value

error_typ

SQI_bind_value(handle, ch, host_name_p, host_var_type, host_var_size,
host_var_p, sqlca_p)

ASE_session_number_typ	handle;	/* IN: session handle */	(pg483.24)
SQI_cursor_handle_typ	ch;	/* IN: cursor holding SQL */	(pg439.30)
char	*host_name_p;	/* IN: bind variable name in SQL */	
SQI_data_type_typ	host_var_type;	/* IN: type of host variable */	(pg440.27)
short	host_var_size;	/* IN: size of host data */	
char	*host_var_p;	/* IN: address of host data */	
SQI_sqlca_typ	*sqlca_p;	/* OUT: standard sqlca */	(pg441.16)

/*

SQI_bind_value binds the run-time values of host variables in a client application into a parsed SQL statement. These host variables, usually used in WHERE and VALUES clauses, are called input host variables because they furnish the data needed before processing the statement. Each SQI_bind_value call binds an input host variable into an SQL statement. The values are taken from their specified locations and sent to the server when SQI_exec_cursor is called. The addresses specified via host_var_p must be valid at that time. Multiple occurrences of the same bind variable name in the statement will receive the same value as a result of one SQI_bind_value call.

For special considerations on binding blob data, see SQI_prepare_blob_update.

NULL values are bound using host_var_type = SQI_null or host_var_size = 0. Null terminated strings of type SQI_ASCII can have size specified as -1; the length will be derived from the data.

ERRORS:

SQI_ERR_INVALID_BIND_VALUE (type not recognized)

*/

31.4. SQI_create_cursor

error_typ

SQI_create_cursor(handle, sql_stmt, ch_p, no_cols_p, sqlca_p)

ASE_session_number_typ	handle;	/* IN: session handle */	(pg483.24)
char	sql_stmt[];	/* IN: text of SQL to hold in cursor */	
SQI_cursor_handle_typ	*ch_p;	/* OUT: returned cursor handle */	(pg439.30)
short	*no_cols_p;	/* OUT: number of columns (if SELECT) */	
SQI_sqlca_typ	*sqlca_p;	/* OUT: standard sqlca */	(pg441.16)

/*

SQI_create_cursor opens a cursor, parses an SQL statement, and attaches

+++ SQI - SQL Services +++

the parsed statement to the cursor. The cursor handle returned should be used on subsequent operations that are related to the parsed SQL statement. For all SQL statements, after any substitution variables are bound using `SQI_bind_value`, `SQI_exec_cursor` must be called. In addition, for `SELECT` statements, result columns must be bound using `SQI_bind_result`, and `SQI_fetch` called to retrieve the data.

The SQL statement in `sql_stmt` is a null terminated string.

ERRORS:

`SQI_ERR_SYNTAX_ERROR`
`SQI_ERR_DATA_TYPE_NOT_SUPPORTED` (for a select expression)
`SQI_ERR_TABLE_ALREADY_EXIST`
`SQI_ERR_DUPLICATE_COLUMN_NAME`
`SQI_ERR_INVALID_COLUMN_NAME`
`SQI_ERR_INVALID_TABLE_NAME`
database errors

*/

31.5. `SQI_dbms_version`

error_typ

`SQI_dbms_version(handle, dbms_p, sqlca_p)`

`ASE_session_number_typ` handle; /* IN: session handle */ (pg483.24)
`SQI_dbms_info_typ` *dbms_p; /* OUT: dbms version info returned */ (pg440.6)
`SQI_sqlca_typ` *sqlca_p; /* OUT: standard sqlca */ (pg441.16)

/*

`SQI_dbms_version` is used to determine which DBMS and version of DBMS has been logged on to. The information is returned to *dbms_p.

ERRORS:

none

*/

31.6. `SQI_describe`

error_typ

`SQI_describe(handle, ch, sqllda_p, sqlca_p)`

`ASE_session_number_typ` handle; /* IN: session handle */ (pg483.24)
`SQI_cursor_handle_typ` ch; /* IN: handle of cursor holding SQL */ (pg439.30)
`SQI_sqllda_typ` *sqllda_p; /* OUT: column descriptions returned */ (pg441.35)
`SQI_sqlca_typ` *sqlca_p; /* OUT: standard sqlca */ (pg441.16)

/*

`SQI_describe` returns the internal data type and size information for the expressions listed in the `SELECT` clause of an SQL statement. The `SELECT` statement must have been parsed by `SQI_create_cursor`.

+++ SQI - SQL Services +++

The sqlda area must be large enough to hold the column descriptions. It can be allocated based on the returned number of columns from SQI_create_cursor.

ERRORS:

 SQI_ERR_SQLDA_MISSING
 SQI_ERR_INVALID_SQLDA_SIZE

*/

31.7. SQI_drop_cursor

error_typ

SQI_drop_cursor(handle, ch_p, sqlca_p) (pg483.24)
 ASE_session_number_typ handle; /* IN: session handle */ (pg439.30)
 SQI_cursor_handle_typ *ch_p; /* IN/OUT: handle of cursor to drop */ (pg441.16)
 SQI_sqlca_typ *sqlca_p; /* OUT: standard sqlca */ (pg441.16)

/*

 SQI_drop_cursor closes the cursor specified by *ch_p.
 The cursor handle pointed to by ch_p is zeroed.

ERRORS:

 database errors

*/

31.8. SQI_exec_cursor

error_typ

SQI_exec_cursor(handle, ch_p, commit, drop_cursor, max_rows, sqlca_p) (pg483.24)
 ASE_session_number_typ handle; /* IN: session handle */ (pg439.30)
 SQI_cursor_handle_typ *ch_p; /* IN/OUT: pointer to cursor handle */ (pg491.22)
 bool commit; /* IN: TRUE=>commit transaction */ (pg491.22)
 bool drop_cursor; /* IN: TRUE=>drop cursor after exec */ (pg491.22)
 unsigned short max_rows; /* IN: number of rows to fetch on
 one SQI_fetch RPC */ (pg441.16)
 SQI_sqlca_typ *sqlca_p; /* OUT: standard sqlca */ (pg441.16)

/*

 SQI_exec_cursor causes a parsed SQL statement to be processed. If the SQL statement is a data manipulation, data definition, or data control statement, the entire SQL statement is performed at this time. The statement must have been prepared with SQI_create_cursor or SQI_prepare_blob_update. Input binds must have been completed for any substitution variables in the statement. If it is a SELECT statement, the client application must explicitly request each row using the SQI_fetch. Output binds can occur any time before an SQI_fetch call.

+++ SQI - SQL Services +++

If `drop_cursor` is specified TRUE the cursor is closed unconditionally, and the cursor handle pointed to by `ch_p` is zeroed.

ERRORS:

`SQL_ERR_VALUE_OUT_OF_RANGE` (for a bound variable)
`SQL_ERR_DATA_TYPE_NOT_SUPPORTED` (for a bound variable)
database errors

*/

31.9. `SQL_exec_sql`

error_typ

```
SQL_exec_sql(handle, sql_stmt, commit, sqlca_p)
  ASE_session_number_typ  handle; /* IN: session handle */           (pg483.24)
  char                    sql_stmt[]; /* IN: text of SQL statement */
  bool                    commit; /* IN: TRUE=>commit the transaction */ (pg491.22)
  SQL_sqlca_typ          *sqlca_p; /* OUT: standard sqlca */       (pg441.16)
```

/*

`SQL_exec_sql` immediately executes a simple SQL statement, specified by the null terminated string in `sql_stmt`. A simple SQL statement does not use any input or output host variables or cursors.

ERRORS:

`SQL_ERR_SYNTAX_ERROR`
`SQL_ERR_TABLE_ALREADY_EXIST`
`SQL_ERR_DUPLICATE_COLUMN_NAME`
`SQL_ERR_SYNTAX_ERROR`
`SQL_ERR_INVALID_COLUMN_NAME`
`SQL_ERR_INVALID_TABLE_NAME`
database errors

*/

31.10. `SQL_fetch`

error_typ

```
SQL_fetch(handle, ch, sqlca_p)
  ASE_session_number_typ  handle; /* IN: session handle */           (pg483.24)
  SQL_cursor_handle_typ   ch; /* IN: handle of cursor holding SQL */ (pg439.30)
  SQL_sqlca_typ          *sqlca_p; /* OUT: standard sqlca */       (pg441.16)
```

/*

`SQL_fetch` returns a row of an executed SELECT query to the client application. The value of each expression in the SELECT list in the query is placed into the corresponding output buffer bound by `SQL_bind_result`. The corresponding indicator variable, if any, is

+++ SQR - SQL Services +++

also set to the expression's return code.

ERRORS:

SQR_ERR_END_OF_FETCH
SQR_ERR_CANCELLED
SQR_ERR_BIND_RESULT_MISSING
SQR_ERR_FETCH_WITHOUT_EXECUTION
SQR_ERR_NOT_A_SELECT_QUERY
SQR_ERR_INVALID_BIND_RESULT
SQR_ERR_VALUE_OUT_OF_RANGE

*/

31.11. SQR_full_use_logon

error_typ

SQR_full_use_logon(db_server_p, handle_p, sqlca_p)

ASE_service_name_typ *db_server_p; /* IN: name of SQL service */ (pg486.28)
ASE_session_number_typ *handle_p; /* OUT: session handle returned */ (pg483.24)
SQR_sqlca_typ *sqlca_p; /* OUT: standard sqlca */ (pg441.16)

/*

SQR_full_use_logon establishes an SQL Services session and returns a session handle to the client which identifies the session. This handle is used in all subsequent operations for the session.

The user is logged on to the database as "F_OPEN".

ERRORS:

SQR_ERR_SQL_SERVICE_NOT_AVAIL
SQR_ERR_LOGON_TWICE

*/

31.12. SQR_logoff

error_typ

SQR_logoff(handle_p, sqlca_p)

ASE_session_number_typ *handle_p; /* IN/OUT: session handle address */ (pg483.24)
SQR_sqlca_typ *sqlca_p; /* OUT: standard sqlca */ (pg441.16)

/*

SQR_logoff terminates a session.
Any incomplete transaction is committed.
Any open cursor is closed.
The handle pointed to by handle_p is zeroed.

ERRORS: none

*/

+++ SQI - SQL Services +++

31.13. SQI_max_blob_size

```
error_typ
SQI_max_blob_size(handle, bufsize, sqlca_p)
  ASE_session_number_typ  handle; /* IN: session handle */           (pg483.24)
  long                    bufsize; /* IN: desired size of server's blob
                               fetch buffer */
  SQI_sqlca_typ          *sqlca_p; /* OUT: standard sqlca */       (pg441.16)
/*
  SQI_max_blob_size sets the buffer size on the server for blob
  column fetches, limiting the number of bytes of data that can
  be returned. Larger columns which are fetched will be truncated.
  Buffers for blob column input are allocated dynamically to the
  correct size and are unaffected by this call.

  This call takes effect upon subsequent execution of an
  SQI_exec_cursor call for the same session.
  Outstanding SQI_fetch sequences are not affected.

  ERRORS: none
*/
```

31.14. SQI_n_describe

```
error_typ
SQI_n_describe(handle, ch, nth, sda_p, sqlca_p)
  ASE_session_number_typ  handle; /* IN: session handle */           (pg483.24)
  SQI_cursor_handle_typ   ch;     /* IN: handle of cursor holding SQL */ (pg439.30)
  short                   nth;    /* IN: position of expression wanted */
  SQI_sqlvar_typ          *sda_p; /* OUT: holds returned description */ (pg441.26)
  SQI_sqlca_typ          *sqlca_p; /* OUT: standard sqlca */       (pg441.16)
/*
  SQI_n_describe returns the internal data type and size information
  of the nth expression listed in the SELECT clause of a parsed
  SQL SELECT statement. The numbering for "nth" starts at one.

  ERRORS:
  SQI_ERR_NOT_A_SELECT_QUERY
  SQI_ERR_INVALID_FIELD_POS
*/
```

31.15. SQI_prepare_blob_update

```
error_typ
SQI_prepare_blob_update(handle, table_name_p, where_cond_p,
```

+++ Sqi - SQL Services +++

```
        blob_col_name_p, bind_name_p, ch_p, sqlca_p)
ASE_session_number_typ  handle;      /* IN: session handle */           (pg483.24)
char                    *table_name_p; /* IN: name of table to update */
char                    *where_cond_p; /* IN: row selection condition */
char                    *blob_col_name_p;
                        /* IN: name of blob column updated */
char                    *bind_name_p; /* IN: bind variable name to use */
Sqi_cursor_handle_typ  *ch_p;        /* OUT: returned cursor handle */       (pg439.30)
Sqi_sqlca_typ          *sqlca_p;     /* OUT: standard sqlca */           (pg441.16)
/*
```

Sqi_prepare_blob_update is provided to support those DBMSs which cannot handle blobs using normal SQL and binds.

It is equivalent to Sqi_create_cursor with an SQL statement:
update <table_name> set <blob_col_name>=:<bind_name>
where <where_cond>

Sqi_prepare_blob_update opens a cursor, creates an SQL statement, and parses the statement in the cursor. The returned cursor handle should be used on subsequent operations that are related to the parsed SQL statement.

If no where condition is specified, all rows of the table are updated.

Using Sqi_bind_value, the blob column data should be bound to the substitution variable name given in bind_name_p. When binding blobs, for either input or output, the length parameter of the Sqi_bind_value or Sqi_bind_result call should be -1, with the length of the data or the length of the buffer in the blob variable itself (Sqi_blob_typ).

Any substitution variables in the where condition must also be bound. Then Sqi_exec_cursor is called to effect the update.

ERRORS:

```
Sqi_err_invalid_table_name
Sqi_err_invalid_column_name
database errors
```

*/

+++ WQS - WorkFlo Queue Services +++

32. WQS DECLARATIONS

```
/* *****  
**                                                                 **  
**   WorkFlo Queue Services (WQS)                               **  
**                                                                 **  
**   *****/
```

```
/* This is the implementation for Queue Services (WQS).
```

Queue Services is intended primarily to support the overall WorkFlo environment by providing functions for the manipulation of queues and queue entries.

```
**/
```

```
/* *****  
**                                                                 **  
**   WQS.defs                                                   **  
**                                                                 **  
**   *****/
```

```
#ifndef WQS_defs  
#define WQS_defs
```

```
#ifndef FileNet_h  
#include <FileNet.h>  
#endif
```

```
#ifndef AS_externals_h  
#include <AS_externals.h>  
#endif
```

```
#ifndef SEC_defs  
#include <SEC.defs>  
#endif
```

```
#ifndef FP_defs  
#include <FP.defs>  
#endif
```

```
/* DB2 defines */  
#define WQS_db2_user_tbl_space "user_tbl_space"  
#define WQS_db2_index_tbl_space "index_tbl_space"
```

+++ WQS - WorkFlo Queue Services +++

```
/* constants */
#define WQS_no_delay ASE_MIN_TIME /* -1999999999 */
#define WQS_no_timeout ASE_UNDEF_TIME /* -2000000000 */
#define WQS_current_time ASE_SPECIAL_TIME1 /* -2000000001 */
/* when specified for entry time,
converts to "now" */
#define WQS_min_priority 0
#define WQS_max_priority 9
#define WQS_default_priority 5
#define WQS_FN_FIELDS 6
#define WQS_max_sys_fields 7
#define WQS_max_name_length 14
#define WQS_max_field_name_length 18
#define WQS_max_text_desc_length 800

#define WQS_max_queue_table 999999
#define WQS_db2_max_col 1012

/* Length of Oracle rowid. */
#define WQS_ROWID_SIZE 18

/* FIELD TYPES */

#define WQS_field_type_number 1
#define WQS_field_type_string 2
#define WQS_field_type_time 3
#define WQS_field_type_selection 4
#define WQS_field_type_document 5
#define WQS_field_type_folder 6
#define WQS_field_type_int 7
#define WQS_field_type_date 8
#define WQS_field_type_access 9 /* not supported */
#define WQS_field_type_boolean 10
#define WQS_field_type_null 11 /* for NULL value */
#define WQS_field_type_decimal 12 /* for decimal value */
typedef unsigned short WQS_queue_field_typ;

/* FIELD UNIQUE TYPES */

#define WQS_not_invert 0
#define WQS_invert 1
#define WQS_unique_invert 2
typedef unsigned short WQS_field_unique_typ;

/* Ordering of the query results */

#define WQS_ascending_order 0
#define WQS_descending_order 1
```

+++ WQS - WorkFlo Queue Services +++

```
typedef unsigned short WQS_sort_order_typ;

#define WQS_delete_none      0
#define WQS_delete_current  1
#define WQS_delete_previous  2
typedef unsigned short      WQS_delete_typ ;

#define WQS_sys_field_pri    0
#define WQS_sys_field_status 1
#define WQS_sys_field_delaytime 2
#define WQS_sys_field_timeout 3
/* the following 2 are really NAMES rather than IDS */
#define WQS_sys_field_userid 4
#define WQS_sys_field_groupid 5
#define WQS_sys_field_entrytime 6

typedef unsigned short      WQS_sys_field_name_typ;

typedef unsigned long       WQS_queue_handle_typ;

typedef unsigned long       WQS_dump_handle_typ;

typedef char                WQS_entry_id_typ [WQS_ROWID_SIZE + 1];

typedef char                WQS_workspace_name_typ [ WQS_max_name_length + 1 ];

typedef char                WQS_queue_name_typ [ WQS_max_name_length + 1 ];

typedef char                WQS_field_name_typ [ WQS_max_field_name_length + 1 ];

/* The Client's view of the definition of a user-defined field.

A field is said to be a rendezvous field if the fld_rendev value is
equal to TRUE.  A queue may contain one and only one such field.
This field must also have fld_unique equal to WQS_unique_invert.
A queue containing a rendezvous field is called a rendezvous queue.

A rendezvous field is used by the WQS_update_entry to identify a
specific queue entry, and WQS_update_entry will either update (if it
already exists) or insert this queue entry.

The name rendezvous queue comes from the use of such queues.  An application
will typically dequeue an entry from one queue, and then enqueue the
the entry to two new queues which are being processed by two different
applications.  After each application dequeues an entry from the queue which
it's processing, it calls WQS_update_entry to do either an insert (first
call for an entry) or update (second call for the same entry) into a
fourth queue, which is a rendezvous queue.  The name rendezvous comes from
```

+++ WQS - WorkFlo Queue Services +++

the fact that the entries in essence get combined when inserted into this fourth queue.

```

*/

typedef struct WQS_prec_scale
{
    unsigned char prec;
    unsigned char scale;
} WQS_prec_scale_typ;

typedef struct WQS_user_field_desc
{
    WQS_field_name_typ    fld_name;        /* name of field */                (pg457.29)
    WQS_queue_field_typ  fld_type;        /* type of field */                (pg456.36)
    unsigned short       fld_length;      /* length of field */
    WQS_field_unique_typ fld_unique;      /* if field contains unique values */ (pg456.43)
    bool                 fld_required;    /* if field is required */         (pg491.22)
    bool                 fld_rendev;     /* if field is for a rendezvous queue */ (pg491.22)
    bool                 fld_display;    /* if field is to be displayed */   (pg491.22)
    WQS_prec_scale_typ  fld_prec_scale; /* for WQS_field_type_decimal */    (pg458.9)
    char                 reserved[12];    /* reserved */
} WQS_user_field_desc_typ;

/* the Client's view of a queue definition */

typedef struct WQS_queue_desc
{
    WQS_workspace_name_typ workspace_name; /* name of workspace */            (pg457.25)
    WQS_queue_name_typ     queue_name;     /* name of queue */                (pg457.27)
    SEC_access_restrictions desc_acc;      /* controls changes to             (pg385.2)
    queue description */
    SEC_access_restrictions content_acc;    /* controls changes to             (pg385.2)
    queue contents */
    char                    *text_desc;    /* text description of q, max
    length must be <=
    WQS_max_text_desc_length */
    unsigned short         num_fields;     /* #fields in array below */
    WQS_user_field_desc_typ *user_field_desc; /* array of user fields */        (pg458.22)
    char                   reserved[12];   /* reserved */
} WQS_queue_desc_typ;

typedef struct WQS_queue_field_choice
{
    WQS_queue_field_typ    qf_type;        (pg456.36)
    union
    {
        char               qf_string_value[ASE_MAX_STR_LEN + 1];
        ASE_selection_typ  qf_selection_value; (pg490.7)
    }
}

```


+++ WQS - WorkFlo Queue Services +++

```

    FP_number          qf_number_value;          (pg211.20)
    int                qf_int_value;
    ASE_time_typ       qf_time_value;          (pg489.31)
    ASE_date_typ       qf_date_value;          (pg489.28)
    ASE_doc_id_typ     qf_docid_value;         (pg483.37)
    ASE_folder_id_typ  qf_folderid_value;      (pg484.13)
    unsigned short     qf_bool_value;
} val;
} WQS_queue_field_choice_typ;

typedef struct WQS_user_field_value
{
    WQS_field_name_typ    field_name;          (pg457.29)
    WQS_queue_field_choice_typ  field_value;   (pg459.9)
} WQS_user_field_value_typ;

typedef struct WQS_sys_field_value
{
    WQS_sys_field_name_typ  sys_field_name;    (pg457.17)
    union
    {
        unsigned short     sf_pri;            /* Priority (WQS_min_priority
        thru WQS_max_priority) */
        bool                sf_status;        /* TRUE=>busy, FALSE=>not busy*/(pg491.22)
        char                sf_user_name[SEC_maxnamelength + 1]; /* User
        associated with the entry */
        char                sf_group_name[SEC_maxnamelength + 1]; /* Group
        associated with the entry. */
        ASE_time_typ        sf_entry_time;    /* Time entry was inserted */ (pg489.31)
        ASE_time_typ        sf_delay_time;    /* Time at which entry should be
        processed (absolute time, not
        offset from current time).
        Queue read routines will
        return queue entries based
        on the value of this
        parameter. See the
        "even_delayed" field in the
        WQS_fetch_spec_typ record. */
        ASE_time_typ        sf_timeout;       /* Time by which this entry
        must be processed. This
        field is set or read
        by the application, but not
        used for anything by WQS. */
    } val;
} WQS_sys_field_value_typ;

/* the true description of the system-defined queue fields kept by

```

+++ WQS - WorkFlo Queue Services +++

```
Queue Services (also matches Oracle's view)
*/

typedef struct WQS_queue_entry_in
{
    unsigned short          num_user_fields;
    WQS_user_field_value_typ *user_field_value;           (pg459.16)
    unsigned short          num_sys_fields;
    WQS_sys_field_value_typ *sys_field_value;             (pg459.46)
} WQS_queue_entry_in_typ;

typedef struct WQS_queue_entry_out
{
    unsigned short          num_user_fields;
    WQS_user_field_value_typ *user_field_value;           (pg459.16)
    unsigned short          num_sys_fields;
    WQS_sys_field_value_typ *sys_field_value;             (pg459.46)
    WQS_entry_id_typ        entry_id;                     (pg457.23)
    char                    reserved[12]; /* reserved */
} WQS_queue_entry_out_typ;

#define WQS_busy_not_ok      0 /* Only fetch entries which are not busy */
#define WQS_busy_ok         1 /* Fetch entry with any busy status. */
#define WQS_busy_must_be_set 2 /* Only fetch entries which are busy */
typedef unsigned short WQS_status_spec_typ;

#define WQS_incomplete_not_ok 0 /* Only fetch entries which are complete */
#define WQS_incomplete_ok    1 /* Fetch entry with any completion status */
#define WQS_incomplete_only  2 /* Only fetch entries which are incomplete */
typedef unsigned short WQS_incomplete_spec_typ;

typedef struct WQS_fetch_spec
{
    ASE_time_typ            deadline; /* Only get entries with           (pg489.31)
    sf_timeout > deadline.
    Use WQS_no_timeout for any */
    unsigned short          min_priority; /* Only get entries which have
    sf_pri >= min_priority. Use
    WQS_min_priority for any */
    unsigned short          max_priority; /* Only get entries which have
    sf_pri <= max_priority. Use
    WQS_max_priority for any */
    char                    group_name [SEC_maxnamelength + 1]; /* Entries
    have only this sf_group_name.
    Use null string for any */
    char                    user_name  [SEC_maxnamelength + 1]; /* Entries
    have only this sf_user_name.
```

+++ WQS - WorkFlo Queue Services +++

```

Use null string for any. */
WQS_status_spec_typ      status_spec;      /* Entry busy status (see above)*/ (pg460.26)
bool                     even_delayed;     /* Fetch entry even if           (pg491.22)
current time < sf_delay_time */
WQS_incomplete_spec_typ incomplete_spec; /* Incomplete status(see above)*/ (pg460.31)
bool                     check_user;      /* Only get entries with the     (pg491.22)
current logged on user being
equal to sf_user_name or being
a member of sf_group_name. */
unsigned short           num_find_fields; /* Number of find_fields below. */
WQS_user_field_value_typ *find_fields;   /* Only get entries which have   (pg459.16)
the values given for the
specified user fields */
WQS_field_name_typ      sort_field;      /* Sort returned records using   (pg457.29)
this user field. Null string
don't want results sorted. */
WQS_sort_order_typ      sort_order;      /* WQS_ascending, WQS_descending */(pg457.1)
char                     reserved[12];   /* reserved */
} WQS_fetch_spec_typ;

/* The following are for statistics returned by WQS_get_server_stats,
WQS_get_queue_stats, and WQS_dump_queue_stats. When adding additional
entrypoints, add lines to the CounterName structure in tool/src/WQS_tool.c */

#define WQS_create_queue_func      1
#define WQS_delete_queue_func     2
#define WQS_get_queue_desc_func   3
#define WQS_update_queue_func     4
#define WQS_open_queue_func       5
#define WQS_close_queue_func      6
#define WQS_count_entries_func    7
#define WQS_read_queue_func       8
#define WQS_read_entry_func       9
#define WQS_insert_entry_func     10
#define WQS_delete_entry_func     11
#define WQS_update_entry_func     12
#define WQS_start_dump_func       13
#define WQS_read_dump_func        14
#define WQS_end_dump_func         15
#define WQS_empty_queue_func      16
#define WQS_create_workspace_func 17
#define WQS_update_workspace_func 18
#define WQS_delete_workspace_func 19
#define WQS_get_workspace_info_func 20
#define WQS_delete_and_insert_entry_func 21
#define WQS_update_selected_func  22
#define WQS_MAX_FUNC              23

```

+++ WQS - WorkFlo Queue Services +++

```
typedef struct WQS_COUNTER
{
    int    Count;
    int    Duration;
    int    OraCount;
    int    OraDuration;
    int    LockCount;
    int    LockDuration;
    char    reserved[12];    /* reserved */
} WQS_COUNTER_typ;

typedef struct WQS_SERVER_STATS
{
    char        ws [WQS_max_name_length + 1];
    char        qn [WQS_max_name_length + 1];
    long        table_id;
    int         user_count;
    bool        is_locked;
    unsigned short trans_typ;    /* WQS_open_queue_func, etc */
    char        reserved[12];    /* reserved */
} WQS_SERVER_STATS_typ;    /* one for each opened queue */

/* Statistical information about a queue */

typedef struct WQS_QUEUE_COUNTER
{
    char        ws [WQS_max_name_length + 1]; /* workspace name */
    char        qn [WQS_max_name_length + 1]; /* queue name */
    long        tbl_id;
    unsigned short num_fields;
    long        result_size;
    int         rendez;
    WQS_COUNTER_typ qcounter[WQS_MAX_FUNC];
    char        reserved[12];    /* reserved */
} WQS_QUEUE_COUNTER_typ;

/*****
**
** ERROR CODES
**
*****/

#define WQS_EXCEPTION 0
#define NOT_WQS_EXCEPTION 1

#define WQS_zother_error 1

/* ERROR VALUES    err_WQS = 151 */
```

(pg491.22)

(pg462.10)

+++ WQS - WorkFlo Queue Services +++

```
#define WQS_err_no_resources err_encode(err_WQS, WQS_EXCEPTION, 1)
/*$M Insufficient resources */
#define WQS_err_invalid_service_handle err_encode(err_WQS, WQS_EXCEPTION, 2)
/*$M Invalid service handle */
#define WQS_err_invalid_queue_handle err_encode(err_WQS, WQS_EXCEPTION, 3)
/*$M Invalid queue handle */
#define WQS_err_invalid_dump_handle err_encode(err_WQS, WQS_EXCEPTION, 4)
/*$M Invalid dump handle */
#define WQS_err_queue_not_empty err_encode(err_WQS, WQS_EXCEPTION, 5)
/*$M Queue is not empty */
#define WQS_err_queue_in_use err_encode(err_WQS, WQS_EXCEPTION, 6)
/*$M Queue is opened by other users. */
#define WQS_err_queue_not_defined err_encode(err_WQS, WQS_EXCEPTION, 7)
/*$M Undefined queue */
#define WQS_err_undefined_user_field err_encode(err_WQS, WQS_EXCEPTION, 8)
/*$M Undefined user field */
#define WQS_err_invalid_sys_field err_encode(err_WQS, WQS_EXCEPTION, 9)
/*$M Invalid system field */
#define WQS_err_invalid_sys_field_data err_encode(err_WQS, WQS_EXCEPTION, 10)
/*$M Invalid system field data - priority value must be between WQS_min_priority
and WQS_max_priority. */
#define WQS_err_field_specified_twice err_encode(err_WQS, WQS_EXCEPTION, 11)
/*$M Field specified more than once */
#define WQS_err_illegal_write_field err_encode(err_WQS, WQS_EXCEPTION, 12)
/*$M Illegal write field - 'status' */
#define WQS_err_required_field_missing err_encode(err_WQS, WQS_EXCEPTION, 13)
/*$M Required field missing */
#define WQS_err_rendez_field_missing err_encode(err_WQS, WQS_EXCEPTION, 14)
/*$M Rendezvous field missing */
#define WQS_err_no_entry_selected err_encode(err_WQS, WQS_EXCEPTION, 15)
/*$M No entry selected */
#define WQS_err_bad_version err_encode(err_WQS, WQS_EXCEPTION, 16)
/*$M Version error */
#define WQS_err_service_not_available err_encode(err_WQS, WQS_EXCEPTION, 17)
/*$M The named service is undefined */
#define WQS_err_queue_already_defined err_encode(err_WQS, WQS_EXCEPTION, 18)
/*$M Queue already defined */
#define WQS_err_queue_not_open err_encode(err_WQS, WQS_EXCEPTION, 19)
/*$M Queue is not open */
#define WQS_err_corrupted_queue_file err_encode(err_WQS, WQS_EXCEPTION, 20)
/*$M Corrupted queue file */
#define WQS_err_26_conversion err_encode(err_WQS, WQS_EXCEPTION, 21)
/*$M Pre-2.6.3 format conversion error */
#define WQS_err_security_violation err_encode(err_WQS, WQS_EXCEPTION, 22)
/*$M Security violation */
#define WQS_err_invalid_field_type err_encode(err_WQS, WQS_EXCEPTION, 23)
/*$M Invalid field type */
```

+++ WQS - WorkFlo Queue Services +++

```
#define WQS_err_db_open err_encode(err_WQS, WQS_EXCEPTION, 24)
/*$M Database open error */
#define WQS_err_db_logon err_encode(err_WQS, WQS_EXCEPTION, 25)
/*$M Database logon error */
#define WQS_err_invalid_entry_id err_encode(err_WQS, WQS_EXCEPTION, 26)
/*$M Invalid entry_id */
#define WQS_err_invalid_find_field err_encode(err_WQS, WQS_EXCEPTION, 27)
/*$M The fetch spec contains an undefined user_field */
#define WQS_err_field_desc_change err_encode(err_WQS,WQS_EXCEPTION, 29)
/*$M Cannot change the type of a field, or reduce the length of a string type
field. If no field has been changed, and you are only adding fields, then
perhaps you neglected to describe the original fields. Make sure to describe
each and every field, and in the original order, whether changed or not.
Fields to be added should follow the originals. */
#define WQS_err_internal_rpc_error err_encode(err_WQS, WQS_EXCEPTION, 30)
/*$M Internal RPC error */
#define WQS_err_table_not_built err_encode(err_WQS, WQS_EXCEPTION, 31)
/*$M Database error - table not build */
#define WQS_err_no_cursor_open err_encode(err_WQS, WQS_EXCEPTION, 32)
/*$M Database error - no cursor opened */
#define WQS_err_entry_not_tagged err_encode(err_WQS, WQS_EXCEPTION, 33)
/*$M Internal error - entry not tagged */
#define WQS_err_cursor_too_large err_encode(err_WQS, WQS_EXCEPTION, 34)
/*$M DB Cursor too large */
#define WQS_err_invalid_workspace err_encode(err_WQS, WQS_EXCEPTION, 35)
/*$M The specified workspace does not exist */
#define WQS_err_load_new_queue_desc err_encode(err_WQS, WQS_EXCEPTION, 36)
/*$M Error while loading new queue desc */
#define WQS_err_workspace_not_created err_encode(err_WQS, WQS_EXCEPTION, 37)
/*$M Create workspace failed */
#define WQS_err_workspace_not_empty err_encode(err_WQS, WQS_EXCEPTION, 38)
/*$M Workspace not deleted - possibly not-empty */
#define WQS_err_invalid_field_value err_encode(err_WQS, WQS_EXCEPTION, 39)
/*$M Invalid field value detected on insertion */
#define WQS_err_dup_unique_val err_encode(err_WQS, WQS_EXCEPTION, 41)
/*$M An entry already exists with the same value for a unique field */
#define WQS_err_invalid_field_length err_encode(err_WQS, WQS_EXCEPTION, 42)
/*$M Fields of type string or selection cannot be 0 length. Fields of type
string also cannot be greater than FN_MAX_STR_LEN (239). */
#define WQS_err_local_func_only err_encode(err_WQS, WQS_EXCEPTION, 43)
/*$M Function can only be run on local server. */
#define WQS_err_wrong_server err_encode(err_WQS, WQS_EXCEPTION, 44)
/*$M Not logged on to the server responsible for the workspace or queue. */
#define WQS_err_row_busy err_encode(err_WQS, WQS_EXCEPTION, 45)
/*$M The selected row is busy (status flag set). */
#define WQS_err_invalid_security_id err_encode(err_WQS, WQS_EXCEPTION, 46)
/*$M An ID in the access restriction list is invalid. */
#define WQS_err_bad_workspace_name err_encode(err_WQS, WQS_EXCEPTION, 47)
```

+++ WQS - WorkFlo Queue Services +++

```
/*$M Workspace names must begin with a letter (upper or lower case). */
#define WQS_err_all_qs_must_be_closed    err_encode(err_WQS, WQS_EXCEPTION, 48)
/*$M All queues in the workspace must be closed for this operation. */
#define WQS_err_illegal_new_name        err_encode(err_WQS, WQS_EXCEPTION, 49)
/*$M <new name> must be NULL when <old name> contains wild-card characters. */
#define WQS_err_queue_def_changed       err_encode(err_WQS, WQS_EXCEPTION, 50)
/*$M Definition of the queue had changed while the session was timed out. */
#define WQS_err_bogus_session_handle    err_encode(err_WQS, WQS_EXCEPTION, 51)
/*$M Invalid session handle - does not match the current active session. */
#define WQS_err_corrupted_ws_desc       err_encode(err_WQS, WQS_EXCEPTION, 52)
/*$M Corrupted workspace description file. */
#define WQS_err_invalid_null_domain     err_encode(err_WQS, WQS_EXCEPTION, 53)
/*$M A domain name parameter cannot be NULL in client libraries. */
#define WQS_err_invalid_area_magic      err_encode(err_WQS, WQS_EXCEPTION, 54)
/*$M The area to be freed has invalid magic or type.
Same area may have been freed twice, or else pointer invalid. */
#define WQS_err_no_user_field           err_encode(err_WQS, WQS_EXCEPTION, 55)
/*$M A queue must have at least 1 user defined field. */
#define WQS_err_illegal_to_drop fld     err_encode(err_WQS, WQS_EXCEPTION, 56)
/*$M The number of user fields in an updated queue description is less than
the original. Make sure to describe each and every field, and in the original
order, whether changed or not. Fields to be added should follow the original
fields. */
#define WQS_err_invalid_sort_field      err_encode(err_WQS, WQS_EXCEPTION, 57)
/*$M The specified sort field is not a valid user-defined field for the queue.*/
#define WQS_err_too_many_opened_queues  err_encode(err_WQS, WQS_EXCEPTION, 58)
/*$M Too many queues opened. */
#define WQS_err_ws_already_defined       err_encode(err_WQS, WQS_EXCEPTION, 59)
/*$M The specified workspace is already defined. */
#define WQS_err_needs_file_to_db_convert err_encode(err_WQS, WQS_EXCEPTION, 60)
/*$M The specified queue has a file based description, and must be converted. */
#define WQS_err_invalid_server_number   err_encode(err_WQS, WQS_EXCEPTION, 61)
/*$M The service number is invalid. Maximum WQS Service number is 999 */
#define WQS_err_invalid_table_name      err_encode(err_WQS, WQS_EXCEPTION, 62)
/*$M The specified or generated table name is too long. */
#define WQS_err_cannot_modify_field     err_encode(err_WQS, WQS_EXCEPTION, 63)
/*$M This database does not support modifying existing fields. */
#define WQS_err_wqss_memory_limit       err_encode(err_WQS, WQS_EXCEPTION, 64)
/*$M WQSS process exiting: exceeded memory limit from environment (WQS_MEM_LIMIT) */
#define WQS_err_invalid_dump_or_queue_handle err_encode(err_WQS, WQS_EXCEPTION, 65)
/*$M Invalid dump or queue handle, cannot find handle in saved_list */
#define WQS_err_session_already_open    err_encode(err_WQS, WQS_EXCEPTION, 66)
/*$M session already open - attempted to create session while one is already active */
#define WQS_err_pid_is_not_lock_owner   err_encode(err_WQS, WQS_EXCEPTION, 67)
/*$M pid is not lock owner */
#define WQS_err_queue_is_currently_opened err_encode(err_WQS, WQS_EXCEPTION, 68)
/*$M queue is currently opened */
#define WQS_err_failed_to_fork_program  err_encode(err_WQS, WQS_EXCEPTION, 69)
```

+++ WQS - WorkFlo Queue Services +++

```
/*$M failed to fork program */
#define WQS_err_failed_to_exec_program      err_encode(err_WQS, WQS_EXCEPTION, 70)
/*$M failed to exec program */
#define WQS_err_invalid_field_unique_type  err_encode(err_WQS, WQS_EXCEPTION, 71)
/*$M Invalid field unique type */
#define WQS_err_cor_abort_msg              err_encode(err_WQS, WQS_EXCEPTION, 72)
/*$M Courier abort message received, see error log for details */
#define WQS_err_cor_reject_msg             err_encode(err_WQS, WQS_EXCEPTION, 73)
/*$M Courier reject message received, see error log for details */
#define WQS_err_sql_overflow                err_encode(err_WQS, WQS_EXCEPTION, 74)
/*$M Internal error: SQL statement overflowed buffer */
#define WQS_err_different_servers          err_encode(err_WQS, WQS_EXCEPTION, 75)
/*$M The two queues specified must reside on the same server */
#define WQS_err_not_rendez                 err_encode(err_WQS, WQS_EXCEPTION, 76)
/*$M Function not allowed on rendezvous queue */
#define WQS_err_cannot_update_two          err_encode(err_WQS, WQS_EXCEPTION, 77)
/*$M WQS_update_selected must specify unique entry - two entries found */
#define WQS_err_actlog_count               err_encode(err_WQS, WQS_EXCEPTION, 78)
/*$M Activity Log count should never be greater than 1 per process.
Abort due to software logic errors. */
#define WQS_err_option_not_allowed         err_encode(err_WQS, WQS_EXCEPTION, 79)
/*$M This option is not allowed because several WQS tables already exist. Use the '-f'
option instead. */
#define WQS_err_queue_table_overflow       err_encode(err_WQS, WQS_EXCEPTION, 80)
/*$M Number of queue table reach maximum */
#define WQS_err_max_db2_col_exceeded       err_encode(err_WQS, WQS_EXCEPTION, 81)
/*$M Number of column exceeded. Max column = 500 (for 4K Page size). Max column = 1012
(for 8K ,16K ,or 32K page size) */
#define WQS_err_upgrade_not_needed         err_encode (err_WQS, WQS_EXCEPTION, 82)
/*$M Upgrade not needed. Tables already have the correct format. */
#define WQS_err_prog_not_support_DB2       err_encode (err_WQS, WQS_EXCEPTION, 83)
/*$M This program or tool is not supported on DB2. */
#define WQS_err_info_only                  err_encode (err_WQS, WQS_EXCEPTION, 84)
/*$M This message is for information and/or debugging purposes only. */
#endif /* MUST BE THE LAST LINE */
```

33. WQS SUBROUTINES

33.1. WQS_close_queue

```
error_typ
WQS_close_queue ( q_handle )
    WQS_queue_handle_typ      q_handle; /* IN: The queue handle. */                (pg457.19)

/* WQS_close_queue closes a queue. The client should call this function when
no further operations on the queue will be performed. The queue handle
is no longer valid upon successful completion of this routine.
```


+++ WQS - WorkFlo Queue Services +++

ERRORS:

WQS_err_invalid_queue_handle

*/

33.2. WQS_count_entries

error_typ

WQS_count_entries (q_handle, fs_p, count_p)

WQS_queue_handle_typ q_handle; /* IN: queue handle */ (pg457.19)

WQS_fetch_spec_typ *fs_p; /* IN: filter condition. If a count of
the total number of entries is
desired, then this parameter
should be NULL. */ (pg461.19)

unsigned long *count_p; /* OUT: Number of entries in the queue */

/* WQS_count_entries returns the number of queue entries in a queue that
satisfy a specified set of filter conditions.

ERRORS:

WQS_err_invalid_queue_handle

WQS_err_invalid_find_field

*/

33.3. WQS_create_queue

error_typ

WQS_create_queue (ims_handle, q_desc_p)

ASE_session_number_typ ims_handle; /* IN: session handle */ (pg483.24)

WQS_queue_desc_typ *q_desc_p; /* IN: queue description */ (pg458.40)

/* WQS_create_queue sets up a new queue by allocating the appropriate
storage, entering the name of the queue into Clearinghouse, and creating
the database table.

ERRORS:

WQS_err_invalid_service_handle

WQS_err_invalid_workspace

WQS_err_queue_already_defined

*/

33.4. WQS_create_workspace

error_typ

WQS_create_workspace (ims_handle, ws_name_p, access_p, text_desc_p)

+++ WQS - WorkFlo Queue Services +++

```
ASE_session_number_typ    ims_handle;    /* IN:  session handle */          (pg483.24)
WQS_workspace_name_typ    ws_name_p;    /* IN:  workspace name */        (pg457.25)
SEC_access_restrictions   *access_p;    /* IN:  access restrictions for   (pg385.2)
                               workspace */
char                       *text_desc_p; /* IN:  text description of
                               workspace */
/* WQS_create_workspace creates a workspace by setting up the necessary
   directories and Clearinghouse entry. The function is considered successful
   if the workspace already exists.

   ERRORS:
       WQS_err_no_resources
       WQS_err_bad_workspace_name
       WQS_err_workspace_not_created
*/
```

33.5. WQS_delete_and_insert_entry

```
error_typ
WQS_delete_and_insert_entry(del_q handle,del_entry_id,ins_q handle,q_entry_in_p)
  WQS_queue_handle_typ del_q handle;    /* IN:  queue handle for delete */  (pg457.19)
  WQS_entry_id_typ     del_entry_id;    /* IN:  single entry id to delete */ (pg457.23)
  WQS_queue_handle_typ ins_q handle;    /* IN:  queue handle for delete */  (pg457.19)
  WQS_queue_entry_in_typ *q_entry_in_p; /* IN:  pointer to entry to insert */ (pg460.11)
/* WQS_delete_and_insert_entry deletes an entry from one queue
   and inserts an entry to another queue. The two queues can be the
   same or different, but must be located on the same server. The
   operations are done in a single database transaction, so neither
   (in case of error) or both (when successful) will occur.

   ERRORS:
       WQS_err_different_servers
       WQS_err_field_specified_twice
       WQS_err_illegal_write_field
       WQS_err_invalid_entry_id
       WQS_err_invalid_field_value
       WQS_err_invalid_queue_handle
       WQS_err_invalid_sys_field
       WQS_err_invalid_sys_field_data
       WQS_err_rendez_field_missing
       WQS_err_required_field_missing
       WQS_err_security_violation
       WQS_err_undefined_user_field
*/
```

+++ WQS - WorkFlo Queue Services +++

33.6. WQS_delete_entry

```
error_typ
WQS_delete_entry( q_handle, num_entries, del_entry_id_p )
  WQS_queue_handle_typ q_handle; /* IN: queue handle */ (pg457.19)
  unsigned short num_entries; /* IN: #entries in del_entry_id_p */
  WQS_entry_id_typ *del_entry_id_p; /* IN: array of entry ids to delete */(pg457.23)

/* WQS_delete_entry deletes one or more entries from a queue. None of the
   entries will be deleted if an error is encountered.

   ERRORS:
   WQS_err_invalid_queue_handle
   WQS_err_invalid_entry_id
   WQS_err_security_violation
*/
```

33.7. WQS_delete_queue

```
error_typ
WQS_delete_queue ( ims_handle, workspace, qname, even_if_full )
  ASE_session_number_typ ims_handle; /* IN: session handle */ (pg483.24)
  WQS_workspace_name_typ workspace; /* IN: name queue's workspace */ (pg457.25)
  WQS_queue_name_typ qname; /* IN: name of queue */ (pg457.27)
  bool even_if_full; /* IN: TRUE=>delete queue even (pg491.22)
   if it contains entries */

/* WQS_delete_queue deletes a queue from the workspace. A queue can only be
   deleted by the Queue service that is responsible for servicing the queue.
   The queue to be deleted must not be opened by any other client.

   ERRORS:
   WQS_err_invalid_service_handle
   WQS_err_invalid_workspace
   WQS_err_queue_not_defined
   WQS_err_queue_in_use
   WQS_err_queue_not_empty
   WQS_err_security_violation
   WQS_err_wrong_server
*/
```

33.8. WQS_delete_workspace

```
error_typ
WQS_delete_workspace (ims_handle, ws_name_p)
  ASE_session_number_typ ims_handle; /* IN: session handle */ (pg483.24)
```

+++ WQS - WorkFlo Queue Services +++

WQS_workspace_name_typ ws_name_p; /* IN: workspace name */ (pg457.25)

/* WQS_delete_workspace deletes a workspace by removing the directories and Clearinghouse entry associated with the workspace. The workspace must not contain any queues or procedures in order to be successfully deleted.

ERRORS:

WQS_err_invalid_service_handle
WQS_err_invalid_workspace
WQS_err_workspace_not_empty

*/

33.9. WQS_empty_queue

error_typ

WQS_empty_queue (q_handle, even_busy)

WQS_queue_handle_typ q_handle; /* IN: The queue handle */ (pg457.19)
bool even_busy; /* IN: TRUE=>delete all entries even (pg491.22)
if some entries are busy.
A busy entry is one with
WQS_STATUS_QF set. */

/* WQS_empty_queue deletes all entries from a queue.

If the even_busy parameter is TRUE, then all entries will be unconditionally deleted.

If the even_busy parameter is FALSE, then all entries will be deleted if there is no entry with the WQS_STATUS_QF set.

The calling process must be the only one that has the queue opened at the time of the call in order for this operation to be successful.

ERRORS:

WQS_err_invalid_queue_handle
WQS_err_queue_in_use
WQS_err_security_violation
WQS_err_row_busy

*/

33.10. WQS_end_dump

error_typ

WQS_end_dump (d_handle)

WQS_dump_handle_typ d_handle; /* IN: dump handle */ (pg457.21)

+++ WQS - WorkFlo Queue Services +++

/* WQS_end_dump terminates a dump. The dump handle is no longer valid after this call.

ERRORS:

WQS_err_invalid_dump_handle

*/

33.11. WQS_flush_act_log

error_typ

WQS_flush_act_log (index_sync)

bool index_sync;

(pg491.22)

/* WQS_flush_act_log - Flushes the shared memory buffers of the activity logs.

ERRORS:

WQS_err_invalid_service_handle

*/

/* RCI 2429 activity logging */

33.12. WQS_get_queue_desc

error_typ

WQS_get_queue_desc (ims_handle, workspace, qname, q_desc_pp)

ASE_session_number_typ ims_handle; /* IN: session handle */

(pg483.24)

WQS_workspace_name_typ workspace; /* IN: queue`s workspace */

(pg457.25)

WQS_queue_name_typ qname; /* IN: queue`s name */

(pg457.27)

WQS_queue_desc_typ **q_desc_pp; /* OUT: Queue description. This pointer points to a service allocated structure which must be freed by calling WQS_retarea (q_desc_pp) */

(pg458.40)

/* WQS_get_queue_desc gets the definition of the specified queue.

ERRORS:

WQS_err_invalid_service_handle

WQS_err_invalid_workspace

WQS_err_queue_not_defined

WQS_err_security_violation

*/

33.13. WQS_get_queue_names

+++ WQS - WorkFlo Queue Services +++

error_typ

WQS_get_queue_names(domain_p,workspace,partial_q_n,num_returned_p,
qname_list_pp)

ASE_domain_name_typ *domain_p; /* IN: domain queues belong to */ (pg486.33)
WQS_workspace_name_typ workspace; /* IN: workspace queues belong to */ (pg457.25)
WQS_queue_name_typ partial_q_n; /* IN: queue name prefix. Only queues (pg457.27)
names beginning with this
string are returned. NULL to
get all queues in workspace */
unsigned short *num_returned_p; /* OUT: number of entries returned */
WQS_queue_name_typ **qname_list_pp; /* OUT: Pointer to array of queue (pg457.27)
names. This pointer points
to a service allocated
array which must be
freed by calling
WQS_retarea (qname_list_pp) */

/* WQS_get_queue_names returns the number of queues in a given workspace and
their names.

ERRORS

WQS_err_invalid_service_handle
WQS_err_invalid_workspace

*/

33.14. WQS_get_server_name

error_typ

WQS_get_server_name (domain_name_p, workspace, qname, server_name_p)

ASE_domain_name_typ *domain_name_p; /* IN: domain where queue resides */ (pg486.33)
WQS_workspace_name_typ workspace; /* IN: workspace where queue (pg457.25)
resides */
WQS_queue_name_typ qname; /* IN: name of queue, can be NULL */ (pg457.27)
ASE_service_name_typ *server_name_p; /* OUT: name of WorkFlo Queue (pg486.28)
Service*/

/* WQS_get_server_name returns the Clearinghouse object name of the
WorkFlo Queue Server responsible for servicing the named workspace
and queue in that workspace.

ERRORS:

WQS_err_invalid_workspace
WQS_err_queue_not_defined

*/

33.15. WQS_get_table_name

+++ WQS - WorkFlo Queue Services +++

error_typ

```
WQS_get_table_name (ims_handle, ws_name_p, qname_p, tbl_name_pp)
  ASE_session_number_typ  ims_handle;    /* IN: session handle */      (pg483.24)
  WQS_workspace_name_typ  ws_name_p;     /* IN: workspace name */    (pg457.25)
  WQS_queue_name_typ     qname_p;       /* IN: queue name */       (pg457.27)
  char                    **tbl_name_pp; /* IN: data base table name. This
                                pointer points to a service
                                allocated structure which
                                must be freed by calling
                                WQS_retarea (q_desc_pp) */
```

/* Retrieves the database table name of the named queue. */

33.16. WQS_get_workspace_info

error_typ

```
WQS_get_workspace_info (ims_handle, ws_name_p, access_p, text_desc)
  ASE_session_number_typ  ims_handle;    /* IN: session handle */      (pg483.24)
  WQS_workspace_name_typ  ws_name_p;     /* IN: workspace name */    (pg457.25)
  SEC_access_restrictions *access_p;     /* OUT: access restrictions of
                                workspace */      (pg385.2)
  char                    text_desc [WQS_max_text_desc_length + 1];
                                /* OUT: text description of
                                workspace */
```

/* WQS_get_workspace_info returns information about a workspace. The information returned are the textual description, and the security access restrictions.

ERRORS:

```
WQS_err_invalid_service_handle
WQS_err_invalid_workspace
WQS_err_bad_workspace_name
```

*/

33.17. WQS_get_workspace_names

error_typ

```
WQS_get_workspace_names (domain_p, partial_ws, num_returned_p, wslist_pp)
  ASE_domain_name_typ     *domain_p;     /* IN: domain of workspaces */ (pg486.33)
  WQS_workspace_name_typ  partial_ws;    /* IN: workspace name prefix.
                                Only names beginning with this
                                string are returned. NULL to
                                get all names. */    (pg457.25)
  unsigned short          *num_returned_p; /* OUT: Number of names returns */
  WQS_workspace_name_typ  **wslist_pp;   /* OUT: Array of workspace names. (pg457.25)
```

+++ WQS - WorkFlo Queue Services +++

This pointer points to a
service allocated array which
must be freed by calling
WQS_retarea (wslist_pp) */

/* WQS_get_workspace_names returns the names of all the workspaces in a given
domain.

ERRORS:

WQS_err_invalid_service_handle

*/

33.18. WQS_insert_entry

error_typ

WQS_insert_entry(q_handle, num_entries, q_entry_in_p)

WQS_queue_handle_typ q_handle; /* IN: queue handle */ (pg457.19)

unsigned short num_entries; /* IN: #entries in q_entry_in_p */

WQS_queue_entry_in_typ *q_entry_in_p; /* IN: array of entries to insert */ (pg460.11)

/* WQS_insert_entry inserts one or more entries into a queue. The insert
operation is a single operation even if more than one entry is to be
inserted. If an error occurs, then none of the entries are successfully
inserted.

ERRORS:

WQS_err_invalid_queue_handle

WQS_err_rendez_field_missing

WQS_err_required_field_missing

WQS_err_illegal_write_field

WQS_err_undefined_user_field

WQS_err_invalid_field_value

WQS_err_invalid_sys_field

WQS_err_invalid_sys_field_data

WQS_err_field_specified_twice

WQS_err_security_violation

*/

33.19. WQS_logoff

error_typ

WQS_logoff (ims_handle)

ASE_session_number_typ ims_handle; /* IN: Session handle */ (pg483.24)

/* Logs off of a WorkFlo Queue Service. The handle is no longer valid
after logoff.

+++ WQS - WorkFlo Queue Services +++

ERRORS:

WQS_err_invalid_service_handle

*/

33.20. WQS_logon

error_typ

WQS_logon (service_name_p, ims_handle_p, timeout_p)

ASE_service_name_typ *service_name_p; /* IN: Name of queue service */ (pg486.28)
ASE_session_number_typ *ims_handle_p; /* IN: Session handle returned */ (pg483.24)
unsigned *timeout_p; /* OUT: obsolete */

/* Logs on to the WorkFlo Queue Service specified. Each queue service may support multiple WorkFlo queues. More than one queue service may exist for a given domain.

ERRORS:

WQS_err_no_resources

WQS_err_service_not_available

WQS_err_bad_version

*/

33.21. WQS_open_queue

error_typ

WQS_open_queue (ims_handle, workspace, qname, return_desc, q_handle_p, num_q_desc_p, q_desc_pp)

ASE_session_number_typ ims_handle; /* IN: session handle */ (pg483.24)
WQS_workspace_name_typ workspace; /* IN: queue's workspace */ (pg457.25)
WQS_queue_name_typ qname; /* IN: queue's name */ (pg457.27)
bool return_desc; /* IN: TRUE=>return description of queue in *q_desc_p. */ (pg491.22)
WQS_queue_handle_typ *q_handle_p; /* OUT: queue handle */ (pg457.19)
unsigned short *num_q_desc_p; /* OUT: 0 or 1. 0 returned if return_desc is FALSE, and 1 returned if return_desc is TRUE and no error. */
WQS_queue_desc_typ **q_desc_pp; /* OUT: Queue description. This pointer points to a service allocated structure which must be freed by calling WQS_retarea (q_desc_pp) */ (pg458.40)

/* WQS_open_queue opens a queue for processing. A queue can only be opened

+++ WQS - WorkFlo Queue Services +++

by the Queue Service that is responsible for servicing the queue. The Queue Service responsible for servicing a queue is usually the Queue Service that creates the queue. A valid queue handle will be returned upon a successful open.

ERRORS:

WQS_err_invalid_service_handle
WQS_err_invalid_workspace
WQS_err_queue_not_defined
WQS_err_no_resources
WQS_err_wrong_server

*/

33.22. WQS_qlogon

error_typ

WQS_qlogon (domain_p, ws, qn, ims_handle_p, timeout_p)
ASE_domain_name_typ *domain_p; /* IN: Domain queue is in */ (pg486.33)
WQS_workspace_name_typ ws; /* IN: Workspace name of queue */ (pg457.25)
WQS_queue_name_typ qn; /* IN: Name of the queue */ (pg457.27)
ASE_session_number_typ *ims_handle_p; /* OUT: Session handle */ (pg483.24)
unsigned *timeout_p; /* OUT: obsolete */

/* Logs on the the WorkFlo Queue Service which the specified queue belongs to. Note that this entry point is not logging on to the queue, it's logging on to a queue service which may contain multiple queues. To use the queue, a subsequent call to open the queue must be performed.

ERRORS:

WQS_err_invalid_workspace
WQS_err_queue_not_defined
WQS_err_service_not_available

*/

33.23. WQS_read_dump

error_typ

WQS_read_dump (d_handle, num_entries, num_read_p, q_entry_pp, finished_p)
WQS_dump_handle_typ d_handle; /* IN: dump handle */ (pg457.21)
unsigned short num_entries; /* IN: #entries to read */
unsigned short *num_read_p; /* IN: #entries actually read */
WQS_queue_entry_out_typ **q_entry_pp; /* OUT: Entries read. This (pg460.21)
pointer points to a service
allocated array which
must be freed by calling

+++ WQS - WorkFlo Queue Services +++

```
bool                               WQS_retarea (q_entry_pp) */
                                   *finished_p; /* OUT: TRUE=>no more entries exist (pg491.22)
                                   FALSE=>more entries for
                                   query. */
/* WQS_read_dump reads one or more entries from a dumped queue.
```

```
ERRORS:
    WQS_err_invalid_dump_handle
*/
```

33.24. WQS_read_entry

```
error_typ
WQS_read_entry(q_handle, entry_id, incomplete, even_busy, set_busy, q_entry_pp)
    WQS_queue_handle_typ    q_handle; /* IN: queue handle */ (pg457.19)
    WQS_entry_id_typ        entry_id; /* IN: id of entry to be read */ (pg457.23)
    bool                    incomplete; /* IN: TRUE=>retrieve entry even if (pg491.22)
                                   incomplete. */
    bool                    even_busy; /* IN: TRUE=>retrieve entry even if (pg491.22)
                                   its status is busy. */
    bool                    set_busy; /* IN: TRUE=>set status to busy */ (pg491.22)
    WQS_queue_entry_out_typ **q_entry_pp; /* OUT: Queue entry read. This (pg460.21)
                                   pointer points to a service
                                   allocated array which
                                   must be freed by calling
                                   WQS_retarea (q_entry_pp) */
```

```
/* WQS_read_entry retrieves a specific entry, the identification of which
is already known.
```

```
ERRORS:
    WQS_err_invalid_queue_handle
    WQS_err_invalid_entry_id
    WQS_err_no_entry_selected
    WQS_err_security_violation
*/
```

33.25. WQS_read_queue

```
error_typ
WQS_read_queue ( q_handle, filter_p, num_entries, set_busy, delete_spec,
                num_read_p, q_entry_pp )
    WQS_queue_handle_typ    q_handle; /* IN: The queue handle */ (pg457.19)
    WQS_fetch_spec_typ      *filter_p; /* IN: A filter condition. Can be (pg461.19)
                                   NULL for no filter, in which
                                   case all entries will
```

+++ WQS - WorkFlo Queue Services +++

```

                                qualify. */
unsigned short      num_entries; /* IN: Number of entries to read */
bool                set_busy;   /* IN: If entries read should be      (pg491.22)
                                set busy or not. */
WQS_delete_typ     delete_spec; /* IN: Specifies which entry, if      (pg457.6)
                                any, is to be deleted. */
unsigned short      *num_read_p; /* OUT: Number of entries read. */
WQS_queue_entry_out_typ **q_entry_pp; /* OUT: Queue entries read. This
                                pointer points to a service
                                allocated array which
                                must be freed by calling
                                WQS_retarea (q_entry_pp) */

/* WQS_read_queue retrieves one or more entries that satisfy a set of filter
conditions from a queue. It can also set the busy status on entries, and
delete entries from the queue.

Setting a queue element busy ("set_busy"=TRUE) allows other readers to
pass in "filter_p->status_spec" equal to "WQS_busy_not_ok" to skip the
entries which are busy.

If "delete_spec" is WQS_delete_none, no entries are deleted from the
queue. If "delete_spec" is WQS_delete_current, the entries currently
read area deleted from the queue and returned to the caller. If
"delete_spec" is WQS_delete_previous, the entries read on the
previous call with the same q_handle are deleted on this call.

ERRORS:
  WQS_err_invalid_queue_handle
  WQS_err_invalid_find_field
  WQS_err_no_entry_selected
  WQS_err_security_violation
*/
```

33.26. WQS_retarea

```

error_typ
WQS_retarea (addr_ptr)
  any_ptr    *addr_ptr; /* IN/OUT: Pointer to pointer to memory to free.      (pg491.24)
                                Set to zero upon return */

/* Returns memory allocated by a prior WQS call. Note that this routine
is called just once, even if an array or hierarchical structure is
being returned. This routine is NOT the equivalent of the Unix
"retarea" call. */
```

+++ WQS - WorkFlo Queue Services +++

33.27. WQS_start_dump

error_typ

WQS_start_dump(q_handle, filter_p, d_handle_p)

WQS_queue_handle_typ q_handle; /* IN: queue handle */ (pg457.19)
WQS_fetch_spec_typ *filter_p; /* IN: records satisfy this filter */ (pg461.19)
WQS_dump_handle_typ *d_handle_p; /* OUT: dump handle returned */ (pg457.21)

/* WQS_start_dump signals the beginning of a queue dump. A number of entries satisfying a filter condition are readied, and returned when the read_dump function is called. A dump_handle will be returned upon a successful start_dump, this dump_handle should be used on subsequent calls to read_dump and end_dump.

ERRORS:

WQS_err_invalid_queue_handle
WQS_err_invalid_find_field
WQS_err_no_entry_selected
WQS_err_security_violation

*/

33.28. WQS_update_entry

error_typ

WQS_update_entry(q_handle, entry_id, q_entry_p, inserted_new_p)

WQS_queue_handle_typ q_handle; /* IN: queue handle */ (pg457.19)
WQS_entry_id_typ entry_id; /* IN: queue entry id */ (pg457.23)
WQS_queue_entry_in_typ *q_entry_p; /* IN: queue entry */ (pg460.11)
bool *inserted_new_p; /* IN/OUT: On input, specifies whether status field can be updated to a specific value. On output, returned TRUE if a new entry has been inserted into a rendezvous queue. */ (pg491.22)

/* WQS_update_entry modifies a specific queue entry.

For an ordinary queue, the entry to be updated is denoted by the entry_id.
For a rendezvous queue, the entry to be updated is determined as follows:

1. If a value is specified for the entry_id, then that entry will be updated. Note that for ordinary queues, the entry_id must be specified.
2. For rendezvous queues, if entry_id is not specified, then the entry with the specified rendezvous field value will be updated. If no entry exists with the specified rendezvous value, a new entry with the

+++ WQS - WorkFlo Queue Services +++

specified rendezvous value will be inserted into the queue and the `inserted_new_p` (see below) parameter will be set.

If no value is specified for the system field `WQS_sys_field_status`, it will be reset. A value can be specified for the field `WQS_sys_field_status` if and only if the parameter `*inserted_new_p_p=1`

ERRORS:

`WQS_err_invalid_queue_handle`
`WQS_err_invalid_entry_id`
`WQS_err_illegal_write_field`
`WQS_err_dup_unique_val`
`WQS_err_required_field_missing`
`WQS_err_undefined_user_field`
`WQS_err_invalid_sys_field`
`WQS_err_invalid_sys_field_data`
`WQS_err_security_violation`

*/

33.29. `WQS_update_queue`

`error_typ`

`WQS_update_queue (q_handle, header_only, q_desc_p)`

`WQS_queue_handle_typ` `q_handle;` /* IN: The queue handle */ (pg457.19)

`bool` `header_only;` /* IN: TRUE=>modify the header only, (pg491.22)
which includes:

1. Text description of queue
2. Name of workspace
3. Name of queue. */

`WQS_queue_desc_typ` `*q_desc_p;` /* IN: New definition of queue. If (pg458.40)
`header_only` is FALSE, then
`q_desc_p` must point to a full
description, with the existing
fields specified in the exact
order as when the queue was
first defined or last updated,
followed by any new fields
to be added. */

/* `WQS_update_queue` enables a client to modify the definition of an existing queue. The allowed modifications include:

- * modify the textual description of the queue.
- * change the name of the workspace (in essence moving the queue from one workspace to another).
- * change the name of the queue.
- * adding new fields
- * enlarge the size of an existing field.
- * adding or deleting indices.

+++ WQS - WorkFlo Queue Services +++

The credentials of the caller will be verified against the description and content access specified for the queue.

ERRORS:

WQS_err_invalid_queue_handle
WQS_err_field_desc_change
WQS_err_queue_in_use
WQS_err_security_violation

*/

33.30. WQS_update_selected

error_typ

```
WQS_update_selected( q_handle, filter_p, entry_id, q_entry_p)
  WQS_queue_handle_typ    q_handle;          /* IN: queue handle */          (pg457.19)
  WQS_fetch_spec_typ      *filter_p;         /* IN: A filter condition. Can be (pg461.19)
                                          NULL for no filter, in which
                                          case all entries will
                                          qualify. */
  WQS_entry_id_typ        entry_id;          /* IN: queue entry id */        (pg457.23)
  WQS_queue_entry_in_typ  *q_entry_p;       /* IN: queue entry */          (pg460.11)
```

/* WQS_update_selected modifies a specific queue entry.
It is not allowed for rendezvous queues.

The entry to be updated is determined by a combination of the entry_id and filter condition, both of which are optional. If no entry_id is given, there must be just one row to satisfy the filter condition. If an entry_id is given, that row must match the filter condition if one is given. If neither is specified, there must be exactly one entry in the queue.

If no value is specified for the system field WQS_sys_field_status, it will be reset.

ERRORS:

WQS_err_invalid_queue_handle
WQS_err_dup_unique_val
WQS_err_required_field_missing
WQS_err_undefined_user_field
WQS_err_invalid_sys_field
WQS_err_invalid_sys_field_data
WQS_err_security_violation
WQS_err_not_rendez
WQS_err_cannot_update_two
WQS_err_no_entry_selected

*/

+++ WQS - WorkFlo Queue Services +++

33.31. WQS_update_workspace

error_typ

```
WQS_update_workspace (ims_handle, ws_name_p, update_name, new_ws_name_p,
                      update_ar, new_access_p, update_text, new_text_p)
ASE_session_number_typ  ims_handle;      /* IN: session handle */      (pg483.24)
WQS_workspace_name_typ  ws_name_p;       /* IN: workspace name */     (pg457.25)
bool                    update_name;     /* IN: if want to update workspace
                                     name */                       (pg491.22)
WQS_workspace_name_typ  new_ws_name_p;   /* IN: new workspace name or NULL
                                     if update_name is FALSE */   (pg457.25)
bool                    update_ar;       /* IN: if want to update access
                                     restrictions */                 (pg491.22)
SEC_access_restrictions *new_access_p;   /* IN: new access restrictions or
                                     NULL if update_ar is FALSE */   (pg385.2)
bool                    update_text;     /* IN: if want to update text
                                     description of workspace */     (pg491.22)
char                    *new_text_p;     /* IN: new text description or NULL
                                     if update_text is FALSE */
```

/* WQS_update_workspace modifies the attributes of a workspace. The caller must have write access permission to the workspace.

The attributes that can be updated are:

1. Name of the workspace. If the name of the workspace is changed, the queues in the workspace will be moved to the new workspace also. In order for the name change to be successful, there should not be an existing workspace with the new name, and the queues in the workspace must all be closed,
2. The access restrictions associated with the workspace, and
3. The textual description of the workspace.

ERRORS:

```
WQS_err_invalid_service_handle
WQS_err_invalid_workspace
WQS_err_security_violation
WQS_err_bad_workspace_name
WQS_err_workspace_not_empty
```

*/

+++ Miscellaneous Declarations +++

34. MISC DECLARATIONS

```
/* Include file common to all clients
 * of Application Services
 */
#ifdef AS_externals_h /* protect against multiple includes */
#define AS_externals_h

/* For IP addresses: struct sockaddr, struct sockaddr_storage,
getaddrinfo(), etc.
*/
#ifdef NT
#include <WinSock2.h>
#else
#include <sys/socket.h>
#include <netdb.h>
#endif
typedef struct sockaddr_storage      sockaddr_storage;

/* Prefix for names is ASE (Application Services Externals) */

typedef unsigned long                ASE_session_number_typ;
#define ASE_INVALID_SESSION_NUMBER  0

/* For those services which take either an IMS name or a service name. */
typedef unsigned short               ASE_nch_name_type_typ;
#define ASE_IMS_NAME                 1
#define ASE_SVC_NAME                  2

/* Image and document definitions */
typedef unsigned long                ASE_image_id_typ;
#define ASE_MIN_IMAGE_ID              73000 /* 100000 for new documents */
#define ASE_MAX_IMAGE_ID              3999999999
#define ASE_INVALID_IMAGE_ID          0
typedef unsigned long                ASE_doc_id_typ;
#define ASE_MIN_DOC_ID                100000
#define ASE_MAX_DOC_ID                3999999999
#define ASE_INVALID_DOC_ID            0
typedef unsigned short               ASE_page_number_typ;
#define ASE_MIN_PAGE_NUMBER           1
#define ASE_MAX_PAGE_NUMBER           1000
#define ASE_INVALID_PAGE_NUMBER       0xffff

/* System serial number definitions */
typedef unsigned long                ASE_ssn_typ;
#define ASE_LOCAL_SSN                 0
```

+++ Miscellaneous Declarations +++

```
#define ASE_DOCCOPY_SSN 0x321 /* reserved for doccopy job */
#define ASE_INVALID_SSN 0xffffffff
#define ASE_MIN_SSN 1000
#define ASE_MAX_SSN 0x7fffffff

/* Server definitions */
#define ASE_MAX_OSAR_SERVERS 64
typedef unsigned short ASE_server_id_typ;
#define ASE_INVALID_SERVER_ID 0
#define ASE_DOC_LOC_SERVER_ID 1

/* Folder definitions */
typedef unsigned long ASE_folder_id_typ;
#define ASE_ROOT_FOLDER_ID 1
#define ASE_INVALID_FOLDER_ID 0

/* The identification of a capability granted to perform a function,
generally used when updating objects. */
typedef unsigned long ASE_capability_typ[5];

/* Tab definitions */
typedef unsigned long ASE_tab_id_typ;
#define ASE_MIN_TAB_ID 10000
#define ASE_MAX_TAB_ID 0xffffffff

/* relational operators: */
#define LSS 0
#define LEQ 1
#define GTR 2
#define GEQ 3
#define EQL 4
#define NEQ 5
#define ASE_HIGHEST_REL_OP 5
typedef unsigned short ASE_relational_op_typ;

/* A page range used in migrating objects. */
typedef struct
{
    ASE_doc_id_typ doc_id; (pg483.37)
    ASE_page_number_typ first_page; (pg483.41)
    ASE_page_number_typ last_page; (pg483.41)
} ASE_page_range_typ;

typedef struct
{
    ASE_doc_id_typ doc_id; (pg483.37)
    ASE_ssn_typ ssn; (pg483.47)
    ASE_page_number_typ first_page; (pg483.41)
```

+++ Miscellaneous Declarations +++

```
ASE_page_number_typ last_page;                                     (pg483.41)
} ASE_page_range_w_ssn_typ;

/* This structure holds an instance of sockaddr_storage, except that
this structure is not guaranteed to be 8-byte aligned as
sockaddr_storage must be. Therefore, users of the network address,
e.g., COR, must use memcpy() to copy this structure to
a local variable of type "struct sockaddr_storage" and then pass that
local variable to C library entry points.
*/
typedef struct ASE_sockaddr_t
{
    unsigned char        addr [sizeof(struct sockaddr_storage)];
    /* NOTE: THE SIZE OF sockaddr_storage IS DIFFERENT ON EVERY PLATFORM */
}
ASE_sockaddr_t;

/* An IP network address */
typedef struct ASE_net_addr_typ
{
    unsigned long        net;
    #define ASE_MAGIC_IADDRESS  0x1ADD9E55 /* MNEMONIC: IADDRESS */
    /* If net==ASE_MAGIC_IADDRESS, client has been recompiled with
the new header files, and, therefore, is using this structure.
If net!=ASE_MAGIC_IADDRESS, client has not been recompiled
using the new header files, and, therefore, is using
ASE_old_NetworkAddress instead of this structure.
So user of the network address (e.g., COR) should convert
the address from the old format (ASE_old_NetworkAddress)
to the new format (this structure).
*/
    unsigned short      host[3];
    unsigned short      socket;
    unsigned int        pad1;
    unsigned int        pad2;
    unsigned short      IPAddrFamily; /* AF_INET=2, AF_INET6=24 */
    unsigned short      IPAddrLen;
    ASE_sockaddr_t      IPAddr; /* Protocol independent IP address */
    /* CANNOT BE USED DIRECTLY. SEE DESCRIPTION OF ASE_sockaddr_t. */
}
ASE_net_addr_typ;

/* THIS STRUCTURE IS FOR INTERNAL USE BY COR AND NCH ONLY!
ALL CLIENTS SHOULD CONTINUE TO USE ASE_net_addr_typ or NCH_NetworkAddress.
*/
typedef struct ASE_old_NetworkAddress
{
    (pg485.17)
```

+++ Miscellaneous Declarations +++

```
    unsigned int          network;
    unsigned short        host[3];
    unsigned short        socket;
}
ASE_old_NetworkAddress;

/* Notification option and requests IDs for document migration requests
and print requests. */

typedef unsigned short    ASE_notify_option_typ;
#define ASE_NOTIFY_ASYNCHRONOUS 1 /* Will not hang caller; block or poll
for completion indication */
#define ASE_NOTIFY_NONE        2 /* Caller does not want notification */
#define ASE_NOTIFY_SYNCHRONOUS 3 /* Caller will hang until done */
typedef unsigned long     ASE_request_id_typ;

/* Clearinghouse style object names.
Each field is null terminated, but no null if the maximum length. */

#define ASE_MAX_ORGANIZATION_LEN 20
#define ASE_MAX_DOMAIN_LEN      20
#define ASE_MAX_OBJECT_LEN      40
typedef struct
{
    char organization[ASE_MAX_ORGANIZATION_LEN];
    char domain[ASE_MAX_DOMAIN_LEN];
    char object[ASE_MAX_OBJECT_LEN];
} ASE_service_name_typ;
typedef struct
{
    char organization[ASE_MAX_ORGANIZATION_LEN];
    char domain[ASE_MAX_DOMAIN_LEN];
} ASE_domain_name_typ;

typedef struct
{
    char organization[ASE_MAX_ORGANIZATION_LEN];
} ASE_organization_name_typ;

#define ASE_DEFAULT_TRANLOG_FAMID 1 /* Transaction logging family ID. */
#define ASE_INVALID_FAMILY_ID     0 /* Invalid family id */
#define ASE_MAX_CURR_SURFS        8 /* Max current surfaces for a fam */
#define ASE_MAX_TRAN_FAMILIES     8 /* Max number of tranlog families */
#define ASE_MAX_FUTURE_SURFS     8 /* Max number of future surfs */
#define ASE_MAX_INTERLEAVED_SURFS 8 /* Max number of interleaved surfs */

typedef unsigned long ASE_surface_id_typ; /* The ID of a surface. */
#define ASE_INVALID_SURFACE_ID 0
```

+++ Miscellaneous Declarations +++

```
typedef unsigned long ASE_surface_offset_typ; /* The offset of a document on
                                              a surface. */
#define ASE_INVALID_SURFACE_OFFSET 0

typedef unsigned short ASE_document_status_typ; /* The status of a copy
                                              of a document on optical disk. */
#define ASE_DOCUMENT_GOOD 0 /* There are no known problems with
                             this copy; this is the default. */
#define ASE_DOCUMENT_BAD 1 /* There are known problems with this copy. */

typedef unsigned short ASE_migrate_status_typ; /* The status of migration of
                                              pages of a document from optical disk. */
#define ASE_ALL_MIGRATED 0 /* All pages are migrated. */
#define ASE_IN_DRIVE 1 /* Some are in a drive. */
#define ASE_IN_SLOT 2 /* Some are in a slot in an OSAR. */
#define ASE_NOT_IN_OSAR 3 /* Disk is out of the storage library */
#define ASE_INTERVENTION_REQ 4 /* Intervention required by operator
                                before migrate can complete. This
                                may be due to a disabled osar, slot,
                                or optical drive. */

#define ASE_MAX_FAM_NAME_LEN 18
typedef char ASE_fam_name_typ [ASE_MAX_FAM_NAME_LEN + 1];
typedef unsigned long ASE_fam_id_typ;

#define ASE_MAX_SDS_NAME_LEN 18
typedef char ASE_sds_name_t [ASE_MAX_SDS_NAME_LEN + 1];
typedef unsigned long ASE_sds_id_t;

/* Optical disk type declarations */

typedef unsigned short ASE_disk_kind;
typedef ASE_disk_kind ASE_disk_type_typ; (pg487.33)
#define ASE_NULL_DISK 0 /* unspecified disk type */
#define ASE_HITACHI12_DISK 1 /* Hitachi 1st generation 12" disk */
#define ASE_HITACHI5_DISK 2 /* Hitachi 1st generation 5" disk */
#define ASE_HITACHI5E_DISK 3 /* Hitachi 1st generation erasable 5" */
#define ASE_OSI12_DISK 4 /* OSI (LSMI) 12" 1st generation */
#define ASE_HITACHI12hc_DISK 5 /* Hitachi 2nd generation 12" disk */
#define ASE_OSI12hc_DISK 6 /* OSI (LMSI) 12" 2nd generation disk */
#define ASE_HP5E_DISK 7 /* HP 5" 650MB erasable disk */
#define ASE_HP5_DISK 8 /* HP 5" 650MB write once disk */
#define ASE_ANSI5Ehc_DISK 9 /* Standard 5" 1.3GB erasable
                             disk, both HP and IBM made. */
#define ASE_ANSI5hc_DISK 10 /* Standard 5" 1.3GB WORM disk,
                             manufactured by both IBM and HP. */
#define ASE_LM6000_DISK 11 /* LMSI LM 6000 WORM disk */
```

+++ Miscellaneous Declarations +++

```

#define ASE_ANSI5Ex4_DISK      12 /* Standard 5" 2.6GB erasable
                                disk, both HP and IBM made. */
#define ASE_ANSI5x4_DISK      13 /* Standard 5" 2.6GB WORM disk,
                                manufactured by both IBM and HP. */
#define ASE_IBM5x4A_DISK      14 /* IBM 5" 2.6GB ablative WORM disk. */
#define ASE_ANSI5Ex8_DISK     15 /* Standard 5" 5.2GB erasable disk,
                                both IBM and HP made. */
#define ASE_ANSI5x8_DISK      16 /* Standard 5" 5.2GB WORM disk,
                                both IBM and HP made. */
#define ASE_IBM5x8A_DISK      17 /* IBM 5" 5.2GB ablative WORM disk. */
#define ASE_LM8000_DISK       18 /* LMSI LM 8000 WORM disk */

#define ASE_ANSI5Ex14_DISK    19 /* Standard 5" 9.1GB erasable
                                disk, HP made for now. */
#define ASE_ANSI5x14_DISK     20 /* Standard 5" 9.1GB WORM disk,
                                HP made for now. */

#define ASE_MSAR_1GB_DISK     21 /* MSAR media, RCI 2472 */
#define ASE_MSAR_2GB_DISK     22
#define ASE_MSAR_4GB_DISK     23
#define ASE_MSAR_8GB_DISK     24
#define ASE_MSAR_16GB_DISK    25
#define ASE_MSAR_32GB_DISK    26

#define ASE_UDO_1_E_DISK      27 /* 5" UDO 30 GB erasable disk */
#define ASE_UDO_1_DISK        28 /* 5" UDO 30 GB worm disk */
#define ASE_UDO_2_E_DISK      29 /* 5" UDO 60 GB erasable disk */
#define ASE_UDO_2_DISK        30 /* 5" UDO 60 GB worm disk */

#define ASE_HP5Ex4_DISK       ASE_ANSI5Ex4_DISK
#define ASE_HP5x4_DISK        ASE_ANSI5x4_DISK

#define ASE_HP5Ex8_DISK       ASE_ANSI5Ex8_DISK
#define ASE_HP5x8_DISK        ASE_ANSI5x8_DISK

#define ASE_HP5Ex14_DISK      ASE_ANSI5Ex14_DISK
#define ASE_HP5x14_DISK       ASE_ANSI5x14_DISK

#define ASE_DEFAULT_DISK_TYPE ASE_HITACHI12_DISK

/*
   The ASE_350DISK_BEGIN define demarcates the beginning of the new
   docheader format released in 3.5.0 release. This was done to
   accommodate multiple sector size support and increased the
   MAX number of sectors per surface.
*/
#define ASE_350DISK_BEGIN      15

```

+++ Miscellaneous Declarations +++

```
#define ASE_MAX_DISK_KINDS          31

#define ASE_DISK_TYPE_HITACHI_12    ASE_HITACHI12_DISK
#define ASE_DISK_TYPE_HITACHI_5     ASE_HITACHI5_DISK

#define ASE_REL_TO_CLOSING          '\0'
#define ASE_REL_TO_ENTRY            '1'

#define ASE_ARCHIVE                  '1'
#define ASE_DELETE                    '\0'

#define ASE_MAX_BIG_O_LEN            18
#define ASE_MAX_IXNAME                ASE_MAX_BIG_O_LEN
#define ASE_MAX_DCNAME_SIZE          ASE_MAX_BIG_O_LEN

#define ASE_MAX_ANYNAMELEN           14
#define ASE_MAX_DCFORMLEN            ASE_MAX_ANYNAMELEN
#define ASE_MAX_WFQUEUEUENAME        ASE_MAX_ANYNAMELEN
#define ASE_MAX_WFSYSNAME            ASE_MAX_ANYNAMELEN

#define ASE_MAX_ANYDESCLEN           30
#define ASE_MAX_IXMASKLEN            ASE_MAX_ANYDESCLEN
#define ASE_MAX_IXDESCLEN            ASE_MAX_ANYDESCLEN
#define ASE_MAX_DCDESCLEN            ASE_MAX_ANYDESCLEN

typedef long      ASE_date_typ; /* # of days since a certain Greenwich Mean
                                Time date. Specifies a date but not a
                                time of day on that date. */
typedef long      ASE_time_typ; /* # of seconds since midnight GMT on a
                                certain date. Specifies a date and time
                                of day on that date. */

#define ASE_MAX_DBCOLS                254

#define ASE_MIN_DATE                  (ASE_date_typ) -2932600 /* 10/25/-6060 */
#define ASE_MAX_DATE                   (ASE_date_typ) 2932600 /* 03/10/9999 */
#define ASE_UNDEF_DATE                 (ASE_date_typ) (-2000000000)

#define ASE_MIN_TIME                   (ASE_time_typ) (-1999999999)
#define ASE_MAX_TIME                    (ASE_time_typ) 2147000000 /*01/14/2038 04:53:20*/
#define ASE_UNDEF_TIME                 (ASE_time_typ) ASE_UNDEF_DATE
/* The following can be used in different services for special purposes,
 * for example, "the current time"
 */
#define ASE_SPECIAL_TIME1              (ASE_time_typ) (-2000000001)
#define ASE_SPECIAL_TIME2              (ASE_time_typ) (-2000000002)
```

+++ Miscellaneous Declarations +++

```
#define ASE_SPECIAL_TIME3      (ASE_time_typ) (-2000000003)
#define ASE_SPECIAL_TIME4      (ASE_time_typ) (-2000000004)

#define ASE_MAX_STR_LEN        239 /* Max length of string index in INX */
#define ASE_UNDEF_DOCCLASS     0

typedef char                    ASE_selection_typ[2];

#define ASE_SLU_TYPE_GENERIC    1
#define ASE_SLU_TYPE_VW        2
typedef unsigned long           ASE_slu_typ;

#define FN_SHOW_UI             0x00000000
#define FN_DONT_SHOW_UI        0x00000001

typedef char ASE_version_number_typ[8];

/***** RETENTION/ACCESS/SECURITY *****/

#define ASE_MAX_RETENTION      16383 /* 2(14) -1 */
#define ASE_MIN_RETENTION      -996

#endif
/* stamp
0I^AXOR5LcFs@m?T4KbE\?VAP;Jsm^>Y?U2_`DZ=XLN1HbDZ>X=M0]^AX;e5LcF]F[=W;KbE\?V9b3JaE_EU>e9IbG\=
T7T1H_WkDS<P0GaAX;R;LcF]@k<g;TbE */
/*
 * FileNet.h - FileNet "C" standards include file
 */

#ifndef FileNet_h
#define FileNet_h

#define not !
#define andif &&
#define orif ||

#define elif else if

#define FN_ENTRY

#ifndef NULL
#define NULL 0
#endif

#define TRUE 1
#define FALSE 0

#if defined (MPE) || defined (HPUX) | defined(SOLARIS)
```


+++ Miscellaneous Declarations +++

```
# define LONG_ALIGN_DIR
#endif

typedef char          FN_int8_t;
typedef unsigned char FN_uint8_t;
typedef short        FN_int16_t;
typedef unsigned short FN_uint16_t;
typedef long         FN_int32_t;
typedef unsigned long FN_uint32_t;
typedef FN_uint8_t   FN_byte_t;           (pg491.5)
typedef FN_uint8_t   FN_bool_t;          (pg491.5)
typedef unsigned short FN_bool16_t;

#if defined(_WIN32)
    typedef __int64          FN_int64_t;
    typedef unsigned __int64 FN_uint64_t;
#else
    typedef long long        FN_int64_t;
    typedef unsigned long long FN_uint64_t;
#endif

#if !defined(BOOL_NAME_CONFLICT) && !defined(BOOL) && !defined(BYTE) && !defined(ANY_PTR)
    #define bool      unsigned short
    #define byte      unsigned char
    #define any_ptr   char *
#endif

/* Declaration of FileNet GUID type */
typedef struct FN_GUID_t
{
    FN_uint32_t    data1;           (pg491.9)
    FN_uint16_t    data2;           (pg491.7)
    FN_uint16_t    data3;           (pg491.7)
    FN_uint8_t     data4 [8];      (pg491.5)
}
FN_GUID_t;

/* String form of GUIDs:

   "{12345678-1234-1234-1234-010203040506}<null>"

   This takes 39 characters including the trailing null.
   Round it up to next multiple of 4 = 40.
*/
typedef struct FN_GUID_char8_t
{
    char          s [40];
}
FN_GUID_char8_t;
```

+++ Miscellaneous Declarations +++

```
/* UTF16 UNICODE form of GUIDs:
   "{12345678-1234-1234-1234-010203040506}<null>"
*/
typedef struct FN_GUID_char16_t
{
    FN_uint16_t          s [40];
}
    FN_GUID_char16_t;

#if defined(_WIN32)

#include <windef.h>

#else

#ifndef FAR
#define FAR
#endif

#ifndef PASCAL
#define PASCAL
#endif

#ifndef BOOL
#define BOOL      bool
#endif

#endif

#ifndef ErrEncode_h
#include <ErrEncode.h>
#endif

#endif /* FileNet_h */
/*****
 *
 * © Copyright IBM Corporation 1985, 2008 All Rights Reserved.
 *
 *****/

/*
 * ErrEncode.h - Provide error numbers for FileNet software.
 *
 */

/*****
```

(pg491.7)

+++ Miscellaneous Declarations +++

```
* NOTE: Error tuples can get passed up through the software and
* into user programs (either WorkFlo, WAL, or Visual WorkFlo).
* The include files used by user programs (if not hard-coded)
* are different files than this one. Therefore error tuple
* values (including err_category, err_function, and err_number)
* cannot be changed without serious justification and
* notification. Changes include renumbering values, splitting
* or combining error numbers, and removing error numbers.
* Adding new values for new types of errors just needs
* notification. Notification should be given to the current
* developers of WorkFlo, WAL, and Visual WorkFlo, at least
* (they in turn need to change files and inform documenters).
* These type of changes should generally only be done on major
* releases (not bug fix releases). In addition, a version of
* these error categories is kept by WFD in pcws.h, so WFD
* developers need to be informed of these changes also.
* NOTE: Many error categories are reused by the PC client software
* (WFD). The categories that are reused are ones which cannot
* be reported by a server to a client. Check with the
* developers of WFD before recycling error categories.
*****/
```

```
#ifndef ErrEncode_h
#define ErrEncode_h

typedef long error_typ;
typedef long severity_typ;

#define err_encode(category, function, number) \
    (((error_typ)((unsigned long)(category)<<24 | \
    (unsigned long)(function)<<16 | (unsigned short)(number)))
#define err_category(value) ((unsigned char)((value)>>24))
#define err_function(value) ((unsigned char)((value)>>16))
#define err_number(value) ((unsigned short)(value))

/* ---- VALUES OF err_category ---- */

#define success ((error_typ) 0)
#define err_Kernel 1 /* General Unix Kernel errors */
#define err_KerNetMM 2 /* Kernel Network S/W Memory Manager */
#define err_KerNetBM 3 /* Kernel Network S/W Buffer Manager */
#define err_KerNetQM 4 /* Kernel Network S/W Queue Manager */
#define err_KerNetSM 5 /* Kernel Network S/W Socket Manager */
#define err_KerNetED 6 /* Kernel Network S/W Ethernet Driver */
#define err_KerNetSYS 7 /* Kernel Network S/W SYS Utilities */
#define err_KerNetIPP 8 /* Kernel Network S/W IPP Process */
#define err_KerNetORP 9 /* Kernel Network S/W ORP Process */
#define err_KerNetEPP 10 /* Kernel Network S/W Echo Packet Protocol */
```

+++ Miscellaneous Declarations +++

```

#define err_KerNetERP 11 /* Kernel Network S/W Error Packet Protocol */
#define err_KerNetIRP 12 /* Kernel Network S/W Internet Routing Protocol */
#define err_DGP 13 /* Datagram Protocol */
#define err_PEP 14 /* Packet Exchange Protocol */
#define err_SPP 15 /* Sequenced Packet Protocol */
#define err_BSP 16 /* Byte Stream Protocol */
#define err_PPM 17 /* Protocol Process Manager */
#define err_RPC 18 /* Remote Procedure Call Protocol */
#define err_Winter 19 /* Windowing Terminal Services */
#define err_AE 20 /* Applications Executive */
#define err_ANO 21 /* Annotations */
#define err_APD 22 /* Archive, Purge, Delete */
#define err_BR 23 /* Backup and Restore */
#define err_BS 24 /* Batch Status services */
#define err_DAM 25 /* Data Access Method */
#define err_DC 26 /* Document Copy Services */
#define err_DD 27 /* Document Display */
#define err_DE 28 /* Document Entry */
#define err_SDC 29 /* Single Document Committal */
#define err_ODX 30 /* Optical Disk Errors */
#define err_AC 31 /* Annotation Copy */
#define err_DR 32 /* Document Retrieval */
#define err_DTM 33 /* Date and Time Manager */
#define err_ERM 34 /* Error Maintenance */
#define err_FW 35 /* Foreign Window */
#define err_FOR 35 /* -- synonym for above */
#define err_HLP 36 /* Help */
#define err_MD 37 /* Modify Document services */
#define err_MF 38 /* Manual File Services */
#define err_PPR 39 /* Print Preparation and Maintenance */
#define err_RMK 40 /* Remote Multi-Keyed-Files (MKF) */
#define err_PUR 41 /* Print User Request */
#define err_RG 42 /* Report Generator */
#define err_SCP 43 /* Server Control Program */
#define err_WP 44 /* Word Processing */
#define err_WQM 45 /* Workflow Queue Manager */
#define err_UFD 46 /* Unfile Documents UI */
#define err_OSI 47 /* OSAR Server Interface */
#define err_GLO 48 /* Process GLOBALS - DS II */
#define err_DBV 49 /* Database Verification */
#define err_CQ 50 /* Committal Queue */
#define err_DBP 51 /* Doc Server Buffer Pool */
#define err_DLS 52 /* Document Locator Service */
#define err_QLG 53 /* Quick Logging */
#define err_BKG 54 /* Document Services Background requests */
#define err_DTR 55 /* Document (object) Transmittal and Retrieval */
#define err_DLI 56 /* Document Locator Interface */
#define err_SLT 57 /* Surface Locator Table */

```

+++ Miscellaneous Declarations +++

```

#define err_DT          58 /* Docs Table */
#define err_FLT        59 /* Family Locator Table */
#define err_FDT        60 /* Family_Disk Table */
#define err_ODD        61 /* Optical Disk Document Descriptor */
#define err_OSS        62 /* OSAR Server Service */
#define err_SNT        63 /* Scalar_Number Table */
#define err_ODT        64 /* Optical Disk Table (DS II) */
#define err_HLT        65 /* High Level Tasks (Doc Services) */
#define err_IST        66 /* Inter-System Transfer */
#define err_SCH        67 /* Scheduler (DS II) */
#define err_ISM        68 /* Inter-System Mover (used by IST) */
#define err_SCA        69 /* Simple CAcheing */
#define err_WRT        70 /* Write_Requests Table */
#define err_SAS        71 /* Server Application Sessions */
#define err_FSM        72 /* FAX Server Manager */
#define err_DED        73 /* Document Entry Display */
#define err_QMA        74 /* DS - Queue Manager Abstract */
#define err_FNT        75 /* FileNet Access Tool (for EDWS) */
#define err_OSA        76 /* DS II - Optical Storage Abstract */
#define err_CSM        77 /* Cache Services Manager */
#define err_PMC        78 /*Page in Memory Cache (PMC) used by DispPgm */
#define err_CNF        79 /* DS - Configuration Abstract */
#define err_DOC        80 /* Document Services */
#define err_WFL        81 /* Workflow maintenance */
#define err_WPL        82 /* Workflow Procedure Language (Compiler) */
#define err_DEO        83 /* Document Entry OCR */
#define err_WFI        84 /* Workflow Interpreter (run time errors) */
#define err_DEW        85 /* Document Entry Workflow Entries abstract */
#define err_CAT        86 /* Document Cataloger */
#define err_PRI        87 /* Print Services */
#define err_BES        88 /* Batch Entry Services */
#define err_FP         89 /* FP_ floating point routines */
#define err_INX        90 /* Index Services */
#define err_DEQ        91 /* Document Entry Queue Manager */
#define err_SCT        92 /* SCT - System Control and Admin. */
#define err_SEC        92 /* SECurity Service */
#define err_CMT        93 /* Document Server Committal Process */
#define err_DTP        94 /* Document Server Data Transfer Process */
#define err_DEE        95 /* Document Entry Editor (manual assembly) */
#define err_DEM        96 /* Document Entry coMmittal */
#define err_FV         97 /* Folder View abstracts */
#define err_SRT        98 /* index record SoRT abstract */
#define err_QMR        99 /* Query Match Report abstract */
#define err_QS         100 /* Query Specification abstract */
#define err_ORK        101 /* 'General' Oracle abstract calls */
#define err_FMT        102 /* ForMaTter for GTI editor */
#define err_OD         103 /* Optical Disk Interface */
#define err_SunIAF     104 /* Sun IAF */

```

+++ Miscellaneous Declarations +++

```

#define err_DEC      105 /* Doc Entry Control - sysnumbers table */
#define err_DCL      106 /* DCL (document class) abstract */
#define err_TAB      107 /* Workstation TAB caching abstract */
#define err_ODS      108 /* Optical Disk Statistics (osar services) */
#define err_IXA      109 /* IXA (indexing fields) abstract */
#define err_IXS      110 /* IXS (indexing synonyms) abstract */
#define err_NLS      111 /* Name Lookup abstract */
#define err_SIT      112 /* Subsystem Interface Tools (Workflow) */
#define err_DEB      113 /* Document Entry Batch status */
#define err_CSA      114 /* Cataloger Server Abstract */
#define err_FS       115 /* Folder Specification abstract */
#define err_DCQ      116 /* Display Command Queue manager abstract */
#define err_SCL      117 /* Scaling images */
#define err_GTI      119 /* GTI Desktop abstrct */
#define err_DAS      120 /* Document attribute display/update */
#define err_GDB      121 /* Generic DataBase interface */
#define err_SYS      122 /* system configuration */
#define err_SIC      123 /* system initialization */
#define err_DSL      124 /* display system log */
#define err_SKF      125 /* Single Keyed File */
#define err_ORA      126 /* Oracle */
#define err_UNIX     127 /* error returned by Unix system subroutine */
#define err_DPD      128 /* Documents to Portable Database abstract */
#define err_CNT      129 /* Performance Counter */
#define err_OPM      130 /* Doc. Srvr. Operator Msg. Passing (DS II) */
#define err_SRF      131 /* Optical Disk Surface Manager (DS II) */
#define err_DIS      132 /* Document Request Distributor (DS II) */
#define err_ARM      133 /* OSAR Arm Manager (DS II) */
#define err_DSA      134 /* Document Services Annotations */
#define err_ADR      135 /* Another Doc Retrieval (for doc svcs) */
#define err_ELA      136 /* Event Logging Abstract */
#define err_GDD      137 /* Generic Document Display */
#define err_GDS      138 /* Generic Document Services */
#define err_GPS      139 /* Generic Page Services */
#define err_WIP      140 /* Work In Progress */
#define err_PDB      141 /* Portable Database abstract */
#define err_NBD      142 /* Network Boot/Dump */
#define err_WSS      143 /* WIP Server Stub */
#define err_AOL      144 /* Applications Oracle Logon - excl or not */
#define err_ASF      145 /* Assign new SurFace - Doc Services */
#define err_ROT      146 /* Rotation of images */
#define err_SQI      147 /* SQL Services */
#define err_FIL      148 /* File Services */
#define err_DEX      149 /* Document Entry Indexing (for wfl) */
#define err_COD      150 /* Code Services */
#define err_WQS      151 /* WorkFlo Queue Service */
#define err_DES      152 /* Document Entry Scanning (for wfl) */
#define err_SCN      153 /* document entry scanning functions */

```

+++ Miscellaneous Declarations +++

```

#define err_FET      154 /* Forms Entry */
#define err_COR      155 /* Courier RPC Protocol */
#define err_NCH      156 /* Network ClearingHouse service */
#define err_SLP      157 /* Session Level Protocol */
#define err_DEU      158 /* Document Entry Utilities */
#define err_CLU      159 /* Index Clustering User */
#define err_DEA      160 /* Document Entry Assembly */
#define err_MKF      161 /* Multi-keyed-files */
#define err_LOG      162 /* Logon */
#define err_IC       163 /* Initialization and Configuration (I & C) */
#define err_DBL      164 /* Doc services DataBase Logon */
#define err_AED      165 /* Am7990 Ethernet driver */
#define err_KerNetRIP 166 /* XNS Routing Information Protocol (RIP) */
#define err_KerNetSSL 167 /* Generic Synch Serial Link Controllers */
#define err_DEV      168 /* Document Entry Verification */
#define err_OLD      169 /* Output to Laser Disk */
#define err_TAP      170 /* TAPE services */
#define err_MSG      171 /* message handling between WorkFlo and FW */
#define err_AIC      172 /* Applications I & C */
#define err_DER      173 /* Document Entry Reports abstract */
#define err_DEP      174 /* Document Entry Property sheets (OPTIONS) */
#define err_EPA      175 /* Easy Page Access */
#define err_CLT      176 /* Cluster map table abstract */
#define err_IAI      177 /* Image Acquisition Interface */
#define err_IBM      178 /* Image Buffer Management */
#define err_DCH      179 /* Dtars ClearingHouse */
#define err_UPD      180 /* User Preference Database */
#define err_PSM      181 /* Print Server Manager */
#define err_PSC      182 /* Print Services Configuration */
#define err_SX       183 /* System Exerciser Diagnostics */
#define err_TEX      184 /* Text EXtraction (COLD documents) */
#define err_PSI      185 /* Presentation Services Interface */
#define err_EDT      186 /* Document EDiTor */
#define err_OFA      187 /* Optical Family Access */
#define err_OCR      188 /* Optical Character Recognition */
#define err_OAP      189 /* OCR Application Program */
#define err_GU       190 /* miscellaneous Graphics Utilities */
#define err_DEZ      191 /* Doc Entry Zone list file abstract */
#define err_TPI      192 /* Transport Interface */
#define err_DAR      193 /* Document Archiving */
#define err_CTF      194 /* Cache To File abstract */
#define err_SU       195 /* System Utilities */
#define err_PA       196 /* Performance Analysis program */
#define err_VP       197 /* Verify Page */
#define err_CT       198 /* Customer Tools */
#define err_CSS      199 /* Contract Special Systems (shared by all) */
#define err_CO       200 /* COLD -- computer output to laser disk */
#define err_IMC      201 /* IMC -- IMAge Conversion abstract */

```

+++ Miscellaneous Declarations +++

```

#define err_SysV      202 /* System 5 shared memory library errors */
#define err_GV       203 /* gv subsystem -- graphics utils for sys V */
#define err_NL       204 /* gl subsystem -- national language translation*/
#define err_PI       205 /* P8 Integration */
#define err_SNM      206 /* Simple Network Management subsystem */
#define err_FFI      207 /* FileNet Formats Interface */
#define err_ICR      208 /* Intelligent Character Recognition */
#define err_RFT      209 /* Remote family table (doc services) */
#define err_NMI      210 /* Network Management Information */
#define err_FTM      211 /* FileNet Task Manager */
#define err_CDB      212 /* Configuration Database */
#define err_VW       213 /* Visual WorkFlo */
#define err_SOD      214 /* SOD - Scsi Optical Device: Driver and Library*/
#define err_ESA      215 /* Externalized Security Access */
#define err_ED       216 /* Encryption/decryption subsystem */
/* #define err_DI    216 // Document Imaging (Redwood) */
#define err_PE       217 /* Personal Edition Visual WorkFlo */
#define err_DB       218 /* Database Maintenance */
#define err_TT       219 /* Tape Transfer System-Images (DDTS) */
#define err_PRG      220 /* Document Purge program (OS) */
#define err_CPT      221 /* Courier Performance Test */
#define err_EBR      222 /* Enterprise Backup/Restore (BR2) */
#define err_CB       223 /* Cache Backup (CB) */
#define err_MON      224 /* System Monitor (Xsysmon) */
#define err_BJC      225 /* Background Job Control */
#define err_GUID     226 /* Globally Unique IDentifiers */
#define err_DVT      227 /* DeVelopment Test programs */
#define err_STO      228 /* Storage Service Generic error */
#define err_DBU      229 /* Database Utility */
#define err_DMS      230 /* DMS 4.x Generic Errors */
#define err_CS       231 /* DMS 4.x Catalog Service */
#define err_SLM      232 /* Software License Management */
#define err_HSM      233 /* HSM Support */
#define err_IMS      234 /* IMS Support */
#define err_ASH      235 /* Application Service Handle */
#define err_LSI      236 /* LSI */
#define err_CATS     237 /* Common Automatic Test Suite */
#define err_SDS      238 /* Single Document Storage */

```

```

/* The last normal error category is 246. After that, when more
error categories are needed, use one of the escape codes for the
error category, and enter a #define into this file for the function
number. This function number will then act as the "real"
error category. Start by using err_ESC1. When the function numbers
up to 254 are used for err_ESC1, then go to err_ESC2, etc. Do not use
function number 255, because that is a wild card to the search
used by msg, etc.
*/

```


+++ Miscellaneous Declarations +++

```
#define err_ESC1      247
#define err_ESC2      248
#define err_ESC3      249
#define err_ESC4      250
#define err_ESC5      251
#define err_ESC6      252
#define err_ESC7      253
#define err_ESC8      254
#define err_LAST      255 /* do not use this error category */
/*                               err_end error categories */

/* Function code defines.
 * The "efn_" series should count down from 254. Number 255 must never
 * be used. It is a wild card to the search for error messages.
 */
#define efn_help      253
#define efn_unix      254
/*                               /* 255 NEVER USE THIS NUMBER */
/* end "efn_" series */

/*
 * This section lists names which are reserved
 * but don't have (or need) numbers.
 */

#define err_FNS      ?? /* FileNet Security Definitions */
#define err_FND      ?? /* FileNet Data Types */

/* This section is reserved function codes to discriminate the
contract special systems. This is necessary, because they all
share category err_CSS).
*/
#define proj_CT      1 /* Central Transport project */
/* end function codes for err_CSS */

#define FN_ERR_INFO      1
/* An informational message being logged to note the
occurrence of an important software event.
Is not an exception and does not require any
operator intervention whatsoever.
Examples are system boot messages, security related
messages... */

#define FN_ERR_WARNING      2
/* A non-fatal exception or low resource condition
which has caused or may soon cause production
```

+++ Miscellaneous Declarations +++

operation of the IMS to become blocked.
This type of error requires prompt attention from
the system administrator, but is not an immediately
blocking problem.
Examples are exceptions which the software can
work around, and low resource conditions. */

#define FN_ERR_OPERATOR 3

/* A normal condition which requires operator
intervention (such as an out of box disk request.)
This type of error may require immediate operator
attention, but it is a normal event, not an
exception. */

#define FN_ERR_SERIOUS 4

/* This is a serious error which indicates that
such a log message should be brought to the
attention of the appropriate service
representative. */

#define FN_ERR_CRITICAL 5

/* An exception or out of resource condition which has
caused, or which may soon cause one or more critical
IMS services to become disabled, or which has caused
or may soon cause the loss of customer data.
This type of error requires immediate attention from
the system administrator. */

/*

Counter IDs used for sys_log2():

Index into syslog_counter[] in SysV overhead_typ.

The bits are OR'd in with the above severity values for sys_log2 calls.

Based on system behavior, some sys_log messages might repeat continuously
thus impacting performance.

sys_log2 provides a method for limiting the number of times a particular
message is output to the elog file - for example, only one message per
thousand might be logged.

To limit a particular sys_log2 call change the code to use one of the
index values below - logically OR the ID with the severity code.

For example:

```
sys_log2(err, SYSLOG_ID_10 | FN_ERR_WARNING, "my message");
```

+++ Miscellaneous Declarations +++

IMPORTANT:

Each index below should only be used for one particular `sys_log2` call. If you are modifying a `sys_log2` call to use one of these indexes, make sure the index you pick is not already in use elsewhere.

See `mv/sysv/src/sys_log.c` for more information.

These indexes are part of the changes made for DR 150484

*/

```
#define SYSLOG_ID_01    0x0100
#define SYSLOG_ID_02    0x0200
#define SYSLOG_ID_03    0x0300
#define SYSLOG_ID_04    0x0400
#define SYSLOG_ID_05    0x0500
#define SYSLOG_ID_06    0x0600
#define SYSLOG_ID_07    0x0700
#define SYSLOG_ID_08    0x0800
#define SYSLOG_ID_09    0x0900
#define SYSLOG_ID_10    0x0A00
#define SYSLOG_ID_11    0x0B00
#define SYSLOG_ID_12    0x0C00
#define SYSLOG_ID_13    0x0D00
#define SYSLOG_ID_14    0x0E00
#define SYSLOG_ID_15    0x0F00
#define SYSLOG_ID_16    0x1000
#define SYSLOG_ID_17    0x1100
#define SYSLOG_ID_18    0x1200
#define SYSLOG_ID_19    0x1300
#define SYSLOG_ID_20    0x1400
```

```
#define SYSLOG_COUNTERS 20
```

```
#endif /* ErrEncode_h */
```

+++ appendix +++

APPENDIX A: INDEX

This section has a list of where all declarations are defined. The format of a location is "xxx.yy", where "xxx" is a page number and "yy" is a line number. Note that locations of only data definitions are listed here; locations of subroutines can be found in the table of contents.

ACCESS_APPROVED	(pg385.38)
ASE_350DISK_BEGIN	(pg488.48)
ASE_ALL_MIGRATED	(pg487.14)
ASE_ANSI5Ehc_DISK	(pg487.44)
ASE_ANSI5Ex14_DISK	(pg488.13)
ASE_ANSI5Ex4_DISK	(pg488.1)
ASE_ANSI5Ex8_DISK	(pg488.6)
ASE_ANSI5hc_DISK	(pg487.46)
ASE_ANSI5x14_DISK	(pg488.15)
ASE_ANSI5x4_DISK	(pg488.3)
ASE_ANSI5x8_DISK	(pg488.8)
ASE_ARCHIVE	(pg489.11)
ASE_DEFAULT_DISK_TYPE	(pg488.40)
ASE_DEFAULT_TRANLOG_FAMID	(pg486.40)
ASE_DELETE	(pg489.12)
ASE_DISK_TYPE_HITACHI_12	(pg489.5)
ASE_DISK_TYPE_HITACHI_5	(pg489.6)
ASE_DOCCOPY_SSN	(pg484.1)
ASE_DOCUMENT_BAD	(pg487.10)
ASE_DOCUMENT_GOOD	(pg487.8)
ASE_DOC_LOC_SERVER_ID	(pg484.10)
ASE_HIGHEST_REL_OP	(pg484.33)
ASE_HITACHI12_DISK	(pg487.36)
ASE_HITACHI12hc_DISK	(pg487.40)
ASE_HITACHI5E_DISK	(pg487.38)
ASE_HITACHI5_DISK	(pg487.37)
ASE_HP5E_DISK	(pg487.42)
ASE_HP5Ex14_DISK	(pg488.37)
ASE_HP5Ex4_DISK	(pg488.31)
ASE_HP5Ex8_DISK	(pg488.34)
ASE_HP5_DISK	(pg487.43)
ASE_HP5x14_DISK	(pg488.38)
ASE_HP5x4_DISK	(pg488.32)
ASE_HP5x8_DISK	(pg488.35)
ASE_IBM5x4A_DISK	(pg488.5)
ASE_IBM5x8A_DISK	(pg488.10)
ASE_IMS_NAME	(pg483.29)
ASE_INTERVENTION_REQ	(pg487.18)
ASE_INVALID_DOC_ID	(pg483.40)
ASE_INVALID_FAMILY_ID	(pg486.41)

+++ appendix +++

ASE_INVALID_FOLDER_ID	(pg484.15)
ASE_INVALID_IMAGE_ID	(pg483.36)
ASE_INVALID_PAGE_NUMBER	(pg483.44)
ASE_INVALID_SERVER_ID	(pg484.9)
ASE_INVALID_SESSION_NUMBER	(pg483.25)
ASE_INVALID_SSN	(pg484.2)
ASE_INVALID_SURFACE_ID	(pg486.48)
ASE_INVALID_SURFACE_OFFSET	(pg487.4)
ASE_IN_DRIVE	(pg487.15)
ASE_IN_SLOT	(pg487.16)
ASE_LM6000_DISK	(pg487.48)
ASE_LM8000_DISK	(pg488.11)
ASE_LOCAL_SSN	(pg483.48)
ASE_MAX_ANYDESCLEN	(pg489.23)
ASE_MAX_ANYNAMELEN	(pg489.18)
ASE_MAX_BIG_O_LEN	(pg489.14)
ASE_MAX_CURR_SURFS	(pg486.42)
ASE_MAX_DATE	(pg489.38)
ASE_MAX_DBCOLS	(pg489.35)
ASE_MAX_DCDESCLEN	(pg489.26)
ASE_MAX_DCFORMLN	(pg489.19)
ASE_MAX_DCNAME SIZE	(pg489.16)
ASE_MAX_DISK_KINDS	(pg489.2)
ASE_MAX_DOC_ID	(pg483.39)
ASE_MAX_DOMAIN_LEN	(pg486.21)
ASE_MAX_FAM_NAME_LEN	(pg487.23)
ASE_MAX_FUTURE_SURFS	(pg486.44)
ASE_MAX_IMAGE_ID	(pg483.35)
ASE_MAX_INTERLEAVED_SURFS	(pg486.45)
ASE_MAX_I XDESCLEN	(pg489.25)
ASE_MAX_I XMASKLEN	(pg489.24)
ASE_MAX_I XNAME	(pg489.15)
ASE_MAX_OBJECT_LEN	(pg486.22)
ASE_MAX_ORGANIZATION_LEN	(pg486.20)
ASE_MAX_OSAR_SERVERS	(pg484.7)
ASE_MAX_PAGE_NUMBER	(pg483.43)
ASE_MAX_RETENTION	(pg490.20)
ASE_MAX_SDS_NAME_LEN	(pg487.27)
ASE_MAX_SSN	(pg484.4)
ASE_MAX_STR_LEN	(pg490.4)
ASE_MAX_TAB_ID	(pg484.24)
ASE_MAX_TIME	(pg489.42)
ASE_MAX_TRAN_FAMILIES	(pg486.43)
ASE_MAX_WFQUEUE NAME	(pg489.20)
ASE_MAX_WFSYSNAME	(pg489.21)
ASE_MIN_DATE	(pg489.37)
ASE_MIN_DOC_ID	(pg483.38)
ASE_MIN_IMAGE_ID	(pg483.34)

+++ appendix +++

ASE_MIN_PAGE_NUMBER	(pg483.42)
ASE_MIN_RETENTION	(pg490.21)
ASE_MIN_SSN	(pg484.3)
ASE_MIN_TAB_ID	(pg484.23)
ASE_MIN_TIME	(pg489.41)
ASE_MSAR_16GB_DISK	(pg488.22)
ASE_MSAR_1GB_DISK	(pg488.18)
ASE_MSAR_2GB_DISK	(pg488.19)
ASE_MSAR_32GB_DISK	(pg488.23)
ASE_MSAR_4GB_DISK	(pg488.20)
ASE_MSAR_8GB_DISK	(pg488.21)
ASE_NOTIFY_ASYNCHRONOUS	(pg486.11)
ASE_NOTIFY_NONE	(pg486.13)
ASE_NOTIFY_SYNCHRONOUS	(pg486.14)
ASE_NOT_IN_OSAR	(pg487.17)
ASE_NULL_DISK	(pg487.35)
ASE_OSI12_DISK	(pg487.39)
ASE_OSI12hc_DISK	(pg487.41)
ASE_REL_TO_CLOSING	(pg489.8)
ASE_REL_TO_ENTRY	(pg489.9)
ASE_ROOT_FOLDER_ID	(pg484.14)
ASE_SLU_TYPE_GENERIC	(pg490.9)
ASE_SLU_TYPE_VW	(pg490.10)
ASE_SPECIAL_TIME1	(pg489.47)
ASE_SPECIAL_TIME2	(pg489.48)
ASE_SPECIAL_TIME3	(pg490.1)
ASE_SPECIAL_TIME4	(pg490.2)
ASE_SVC_NAME	(pg483.30)
ASE_UDO_1_DISK	(pg488.26)
ASE_UDO_1_E_DISK	(pg488.25)
ASE_UDO_2_DISK	(pg488.28)
ASE_UDO_2_E_DISK	(pg488.27)
ASE_UNDEF_DATE	(pg489.39)
ASE_UNDEF_DOCCLASS	(pg490.5)
ASE_UNDEF_TIME	(pg489.43)
ASE_capability_typ	(pg484.19)
ASE_date_typ	(pg489.28)
ASE_disk_kind	(pg487.33)
ASE_disk_type_typ	(pg487.34)
ASE_doc_id_typ	(pg483.37)
ASE_document_status_typ	(pg487.6)
ASE_domain_name_typ	(pg486.33)
ASE_fam_id_typ	(pg487.25)
ASE_fam_name_typ	(pg487.24)
ASE_folder_id_typ	(pg484.13)
ASE_image_id_typ	(pg483.33)
ASE_migrate_status_typ	(pg487.12)
ASE_nch_name_type_typ	(pg483.28)

+++ appendix +++

ASE_net_addr_typ	(pg485.42)
ASE_notify_option_typ	(pg486.10)
ASE_old_NetworkAddress	(pg486.5)
ASE_organization_name_typ	(pg486.38)
ASE_page_number_typ	(pg483.41)
ASE_page_range_typ	(pg484.42)
ASE_page_range_w_ssn_typ	(pg485.2)
ASE_relational_op_typ	(pg484.34)
ASE_request_id_typ	(pg486.15)
ASE_sds_id_t	(pg487.29)
ASE_sds_name_t	(pg487.28)
ASE_selection_typ	(pg490.7)
ASE_server_id_typ	(pg484.8)
ASE_service_name_typ	(pg486.28)
ASE_session_number_typ	(pg483.24)
ASE_slu_typ	(pg490.11)
ASE_sockaddr_t	(pg485.17)
ASE_ssn_typ	(pg483.47)
ASE_surface_id_typ	(pg486.47)
ASE_surface_offset_typ	(pg487.2)
ASE_tab_id_typ	(pg484.22)
ASE_time_typ	(pg489.31)
ASE_version_number_typ	(pg490.16)
AS externals_h	(pg483.9)
BES_APERTURE	(pg26.7)
BES_ARCH	(pg24.10)
BES_ASSEMBLY	(pg25.3)
BES_ASYNCHRONOUS	(pg25.33)
BES_AVAILABLE	(pg24.38)
BES_BAD	(pg26.25)
BES_BATCH	(pg24.4)
BES_BATCH_TOTAL	(pg25.6)
BES_BUSY	(pg24.39)
BES_CATALOG	(pg25.8)
BES_COMMIT	(pg25.7)
BES_COMMIT_QUEUE	(pg25.28)
BES_DDT	(pg24.13)
BES_DEFINE	(pg24.47)
BES_DEL	(pg24.15)
BES_DELETE	(pg26.27)
BES_DEW	(pg24.9)
BES_DONE	(pg25.17)
BES_EQUAL	(pg25.45)
BES_FAST_ASYNCHRONOUS	(pg25.35)
BES_FAX	(pg24.12)
BES_FILTER_ALL	(pg26.38)
BES_FIRST_BATCH_ID	(pg26.41)
BES_FIRST_FOLDER_ID	(pg36.5)

+++ appendix +++

BES_FOLDER	(pg24.14)
BES_FORM	(pg24.6)
BES_GEQ	(pg25.46)
BES_GOOD	(pg26.24)
BES_GT	(pg25.47)
BES_HAS_ERROR	(pg26.33)
BES_HAS_ERRORS	(pg25.19)
BES_IMAGE_VER	(pg25.1)
BES_INDEX	(pg25.4)
BES_INDEX_TYPE_NULL	(pg33.1)
BES_INDEX_TYPE_PACKED	(pg32.48)
BES_INDEX_VER	(pg25.5)
BES_INFO_BATCH	(pg35.14)
BES_INFO_DOC	(pg35.15)
BES_INFO_IDXVAL	(pg35.17)
BES_INFO_IMAGE	(pg35.16)
BES_INPROGRESS_QUEUE	(pg25.27)
BES_INTERRUPTED	(pg25.16)
BES_INVALID_BATCH_ID	(pg35.9)
BES_INVALID_FOLDER_ID	(pg36.6)
BES_INVALID_INFO	(pg35.13)
BES_INVALID_OBJECT_TYPE	(pg36.34)
BES_INVALID_SEQUENCE_ID	(pg36.35)
BES_IN_PROGRESS	(pg25.15)
BES_MANUAL	(pg26.5)
BES_MAX_BATCHDOCS	(pg23.44)
BES_MAX_BATCHPAGES	(pg23.45)
BES_MAX_COMPANION_ID	(pg37.26)
BES_MAX_FOLDER_INFO_LEN	(pg36.9)
BES_MAX_INDEX_LEN	(pg26.42)
BES_MAX_INDICES_PER_DOC	(pg26.40)
BES_MAX_LEVEL	(pg38.4)
BES_MAX_NAME_TYPE	(pg24.17)
BES_MAX_NODE_NAME_LEN	(pg36.8)
BES_MAX_OBJECT_DATA_LEN	(pg36.46)
BES_MAX_PAGES_PER_DOC	(pg26.39)
BES_MAX_TEXT_DATA_LEN	(pg26.43)
BES_MIN_COMPANION_ID	(pg37.25)
BES_MIN_LEVEL	(pg38.3)
BES_NAME_TYPE_ARCH	(pg24.28)
BES_NAME_TYPE_BATCH	(pg24.22)
BES_NAME_TYPE_DDT	(pg24.31)
BES_NAME_TYPE_DEL	(pg24.33)
BES_NAME_TYPE_DEW	(pg24.27)
BES_NAME_TYPE_FAX	(pg24.30)
BES_NAME_TYPE_FOLDER	(pg24.32)
BES_NAME_TYPE_FORM	(pg24.24)
BES_NAME_TYPE_PC	(pg24.29)

+++ appendix +++

BES_NAME_TYPE_QUICK	(pg24.23)
BES_NAME_TYPE_TEST	(pg24.26)
BES_NAME_TYPE_WPS	(pg24.25)
BES_NORMAL_COMMIT	(pg25.39)
BES_NOT_NECESSARY	(pg25.18)
BES_NOT_STARTED	(pg25.14)
BES_NO_MIGRATE	(pg25.40)
BES_NO_QUEUE	(pg25.30)
BES_NUM_PHASES	(pg24.43)
BES_OBJECT_ID_ALL	(pg36.38)
BES_OBJECT_TYPE_ALL	(pg36.37)
BES_OBJECT_TYPE_BATCH	(pg36.41)
BES_OBJECT_TYPE_DOC	(pg36.42)
BES_OBJECT_TYPE_IMAGE	(pg36.43)
BES_OBJECT_TYPE_PHASE	(pg36.44)
BES_OCR	(pg26.6)
BES_OCR_QUEUE	(pg25.26)
BES_OPEN_NORMAL	(pg26.48)
BES_OPEN_READONLY	(pg27.2)
BES_OPTIONAL	(pg25.20)
BES_OVERRIDE	(pg27.1)
BES_PC	(pg24.11)
BES_QUICK	(pg24.5)
BES_RECOMMIT	(pg25.9)
BES_REQUIRED	(pg26.26)
BES_RESCAN	(pg25.2)
BES_REVERSE	(pg27.8)
BES_ROOT_FOLDER_ID	(pg36.4)
BES_SCAN	(pg24.48)
BES_SEQUENCE_ID_ALL	(pg36.39)
BES_SEQUENTIAL	(pg27.7)
BES_SOURCE_UNKNOWN	(pg26.4)
BES_SUCCESS	(pg26.32)
BES_SYNCHRONOUS	(pg25.34)
BES_SYNCHRONOUS_NO_WAIT	(pg25.36)
BES_TEST	(pg24.8)
BES_UNCOMMIT_QUEUE	(pg25.29)
BES_UNKNOWN	(pg26.34)
BES_UNSEEN	(pg26.23)
BES_WPS	(pg24.7)
BES_XIN_PROTOCOL	(pg38.16)
BES_ZOTHER_ERROR	(pg38.15)
BES_batch_cap_typ	(pg28.3)
BES_batch_id_typ	(pg27.19)
BES_batch_name_typ	(pg24.3)
BES_commit_queue_typ	(pg25.25)
BES_companion_desc_typ	(pg37.35)
BES_ctl_desc_typ	(pg34.42)

+++ appendix +++

BES_defs	(pg23.29)
BES_doc_desc_typ	(pg34.30)
BES_doc_list_typ	(pg35.1)
BES_dyn_hdr_desc_typ	(pg31.38)
BES_err_already_in_doc	(pg42.28)
BES_err_already_in_transaction	(pg39.5)
BES_err_bad_batch_name	(pg42.25)
BES_err_bad_batch_total	(pg40.1)
BES_err_bad_handle	(pg40.14)
BES_err_bad_image_batch_id	(pg41.11)
BES_err_bad_index_type	(pg41.37)
BES_err_bad_pages	(pg39.11)
BES_err_bad_relop	(pg39.40)
BES_err_bad_sess_id	(pg38.23)
BES_err_bad_version	(pg38.20)
BES_err_batch_busy	(pg38.38)
BES_err_batch_exists	(pg38.32)
BES_err_batch_for_committal	(pg42.13)
BES_err_batch_name_too_long	(pg40.38)
BES_err_batch_not_found	(pg38.35)
BES_err_batch_not_locked	(pg41.43)
BES_err_batch_not_open	(pg38.41)
BES_err_batch_overridden	(pg42.10)
BES_err_batch_too_large	(pg41.8)
BES_err_buf_size	(pg45.23)
BES_err_commit_batch_total	(pg40.32)
BES_err_commit_failed	(pg41.40)
BES_err_companion_not_found	(pg44.43)
BES_err_connection_already_open	(pg40.44)
BES_err_connection_not_open	(pg40.47)
BES_err_cross_committal	(pg42.1)
BES_err_ctl_not_found	(pg39.32)
BES_err_debug_only	(pg42.40)
BES_err_doc_exists	(pg39.8)
BES_err_doc_not_found	(pg39.14)
BES_err_doc_without_page	(pg42.22)
BES_err_end_of_info_list	(pg43.11)
BES_err_exclusive_use_denied	(pg42.46)
BES_err_folder_bad_path_format	(pg43.20)
BES_err_folder_batch_has_child	(pg44.30)
BES_err_folder_dup_node	(pg43.29)
BES_err_folder_exceed_max_base_name_len	(pg43.47)
BES_err_folder_exceed_max_info_len	(pg43.23)
BES_err_folder_has_changed	(pg44.2)
BES_err_folder_invalid_parameter	(pg44.21)
BES_err_folder_no_children_found	(pg44.14)
BES_err_folder_no_node_found	(pg43.32)
BES_err_folder_no_parent	(pg43.17)

+++ appendix +++

BES_err_folder_node_not_empty	(pg44.11)
BES_err_folder_parent_is_ancestor	(pg45.8)
BES_err_folder_parent_is_batch	(pg43.26)
BES_err_image_companion_exists	(pg45.14)
BES_err_image_companion_invalid_parameter	(pg44.37)
BES_err_image_exists	(pg38.44)
BES_err_image_in_doc	(pg41.16)
BES_err_image_index_exists	(pg41.22)
BES_err_image_not_found	(pg38.47)
BES_err_image_not_verified	(pg40.23)
BES_err_imagebuf_not_alloc	(pg41.46)
BES_err_in_exclusive_use	(pg42.43)
BES_err_index_not_verified	(pg40.35)
BES_err_internal	(pg41.34)
BES_err_internal_rpc_error	(pg39.20)
BES_err_invalid_batch_cap	(pg40.41)
BES_err_invalid_batch_type	(pg39.29)
BES_err_invalid_bulk_data_src	(pg41.2)
BES_err_invalid_companion_id	(pg44.40)
BES_err_invalid_doc_id	(pg44.27)
BES_err_invalid_image_id	(pg42.37)
BES_err_invalid_image_index	(pg41.25)
BES_err_invalid_info_spec	(pg43.4)
BES_err_invalid_info_type	(pg43.8)
BES_err_invalid_last_companion_id	(pg44.46)
BES_err_invalid_level	(pg45.17)
BES_err_invalid_migr_delay_val	(pg42.34)
BES_err_invalid_move_phase	(pg44.17)
BES_err_invalid_name_format	(pg43.14)
BES_err_invalid_object_id	(pg44.5)
BES_err_invalid_object_seq_no	(pg43.35)
BES_err_invalid_object_type	(pg43.41)
BES_err_invalid_queue	(pg40.17)
BES_err_invalid_write	(pg42.16)
BES_err_ixdir_not_found	(pg39.17)
BES_err_ixval_cnt	(pg40.4)
BES_err_ixval_len	(pg42.31)
BES_err_ixval_not_found	(pg39.37)
BES_err_move_doc_same_batch	(pg45.4)
BES_err_no_companion_found	(pg45.1)
BES_err_no_image_index	(pg41.28)
BES_err_no_local_batch_service	(pg45.20)
BES_err_no_required_index	(pg40.29)
BES_err_no_resources	(pg38.29)
BES_err_no_transaction	(pg39.2)
BES_err_non_debugging	(pg39.23)
BES_err_not_create_image	(pg42.19)
BES_err_not_exclusive_logon	(pg43.1)

+++ appendix +++

BES_err_not_logged_on_to_db	(pg39.26)
BES_err_object_data_invalid_parameter	(pg44.34)
BES_err_object_data_not_found	(pg44.8)
BES_err_object_data_too_large	(pg43.38)
BES_err_object_not_in_batch	(pg43.44)
BES_err_page_cnt	(pg40.9)
BES_err_passed_null_ptr	(pg44.24)
BES_err_phase_incomplete	(pg40.20)
BES_err_readonly_batch	(pg42.7)
BES_err_session_busy	(pg41.31)
BES_err_string_too_long	(pg41.5)
BES_err_text_data_size	(pg45.26)
BES_err_too_many_ids	(pg38.26)
BES_err_too_many_images	(pg42.4)
BES_err_too_many_indices	(pg39.46)
BES_err_too_many_pages	(pg39.43)
BES_err_wrong_queue	(pg40.26)
BES_execute_com_typ	(pg25.32)
BES_filter_typ	(pg27.39)
BES_folder_desc_typ	(pg36.25)
BES_folder_node_name_typ	(pg36.11)
BES_generic	(pg38.13)
BES_handle_typ	(pg27.17)
BES_hdr_desc_typ	(pg31.43)
BES_image_desc_typ	(pg32.41)
BES_image_ix_desc_typ	(pg32.45)
BES_index_stat_typ	(pg26.31)
BES_info_rec_typ	(pg35.44)
BES_info_spec_typ	(pg35.36)
BES_info_typ	(pg35.19)
BES_ixdir_desc_typ	(pg32.24)
BES_ixval_desc_typ	(pg33.37)
BES_ixval_spec_typ	(pg35.25)
BES_level_0	(pg37.48)
BES_level_1	(pg38.1)
BES_migrate_com_typ	(pg25.38)
BES_mkf_ixval_desc_typ	(pg33.45)
BES_node_id_typ	(pg36.10)
BES_object_id_typ	(pg37.1)
BES_object_rec_typ	(pg37.17)
BES_object_spec_typ	(pg37.9)
BES_object_type_typ	(pg36.48)
BES_open_options_typ	(pg26.47)
BES_open_status_typ	(pg24.37)
BES_phase_num_typ	(pg24.46)
BES_phase_status_typ	(pg25.13)
BES_relop_typ	(pg25.44)
BES_sequence_id_typ	(pg37.2)

+++ appendix +++

BES_sort_order_typ	(pg27.6)
BES_source_typ	(pg26.3)
BES_stat_hdr_desc_typ	(pg29.4)
BES_verify_stat_typ	(pg26.22)
BES_xmain	(pg38.14)
BOOL	(pg492.27)
CKS_defs	(pg89.14)
CSML_TRACE	(pg93.20)
CSMS_TRACE	(pg93.20)
CSM_ALL_CACHES	(pg92.10)
CSM_ALL_DOCS	(pg92.6)
CSM_ALL_PAGES	(pg92.7)
CSM_ALL_SSNS	(pg92.5)
CSM_CACHE_DELETE	(pg93.11)
CSM_CACHE_READ	(pg93.9)
CSM_CACHE_READDELETE	(pg93.13)
CSM_CACHE_READWRITE	(pg93.12)
CSM_CACHE_READWRITEDELETE	(pg93.15)
CSM_CACHE_WRITE	(pg93.10)
CSM_CACHE_WRITEDELETE	(pg93.14)
CSM_CLIENT_ATTR_ELMNTS	(pg92.8)
CSM_CURRENT_DURATION	(pg92.41)
CSM_FULL	(pg93.19)
CSM_INVALID_CACHE_ID	(pg92.9)
CSM_IO_error	(pg95.34)
CSM_LOCKED_DURATION	(pg92.37)
CSM_MAGIC	(pg99.39)
CSM_MAGIC1	(pg99.40)
CSM_MASK	(pg92.14)
CSM_MAX_PARTITIONS	(pg92.3)
CSM_MAX_TRAN_CACHES	(pg91.47)
CSM_MIGRATE_DURATION	(pg92.39)
CSM_NEAR_FULL	(pg93.19)
CSM_NOT_FULL	(pg93.19)
CSM_OBJECT_READ	(pg92.20)
CSM_OBJECT_WRITE	(pg92.21)
CSM_PREFETCH_DURATION	(pg92.38)
CSM_READ_NOUPDATE	(pg92.22)
CSM_REFRESH_DURATION	(pg92.40)
CSM_ROUND	(pg92.15)
CSM_SECT_SIZE	(pg92.13)
CSM_TEMP_ID_MAX	(pg92.12)
CSM_TEMP_ID_MIN	(pg92.11)
CSM_XIN_PROTOCOL	(pg95.12)
CSM_XNOT_IN_PROTOCOL	(pg95.13)
CSM_ZOTHER_ERROR	(pg95.15)
CSM_already_exists	(pg95.25)
CSM_attr_bad_breakup_object	(pg99.33)

+++ appendix +++

CSM_bad_version	(pg95.22)
CSM_bucket_struct	(pg92.47)
CSM_bucket_typ	(pg93.4)
CSM_bulk_read_err	(pg98.12)
CSM_cache_access_mode	(pg93.8)
CSM_cache_attr_struct	(pg93.30)
CSM_cache_attr_typ	(pg93.47)
CSM_cache_config_struct	(pg94.1)
CSM_cache_config_typ	(pg94.8)
CSM_cache_fullness_typ	(pg93.19)
CSM_cache_id_typ	(pg93.22)
CSM_cache_list_typ	(pg92.45)
CSM_cache_name_typ	(pg93.25)
CSM_config_err	(pg99.12)
CSM_defs	(pg91.31)
CSM_disk_addr_struct	(pg94.39)
CSM_disk_addr_typ	(pg94.47)
CSM_err_cannot_find_named_service	(pg98.24)
CSM_err_internal_rpc_error	(pg98.15)
CSM_err_not_debugging	(pg98.18)
CSM_exim_foreground_only	(pg99.15)
CSM_fatal_err	(pg97.5)
CSM_ilgl_csum_update	(pg99.5)
CSM_illegal_disk_partition	(pg96.26)
CSM_invalid_age_duration	(pg97.20)
CSM_invalid_bulk_data_sink	(pg97.17)
CSM_invalid_bulk_data_source	(pg97.14)
CSM_invalid_cache_handle	(pg97.8)
CSM_invalid_checksum	(pg99.2)
CSM_invalid_close	(pg96.41)
CSM_invalid_io_handle	(pg98.44)
CSM_invalid_maxlength	(pg97.23)
CSM_invalid_mode	(pg96.29)
CSM_invalid_name_type	(pg96.14)
CSM_invalid_object	(pg98.40)
CSM_invalid_object_desc	(pg96.32)
CSM_invalid_object_handle	(pg95.28)
CSM_invalid_object_size	(pg97.11)
CSM_invalid_relop	(pg97.26)
CSM_invalid_sas_handle	(pg97.48)
CSM_invalid_session_handle	(pg95.31)
CSM_invalid_wildcard	(pg97.29)
CSM_io_handle_typ	(pg93.26)
CSM_logon_access_mode	(pg98.27)
CSM_logon_when_connect_open	(pg98.3)
CSM_mult_open	(pg96.38)
CSM_nil_bytes_trans_pointer	(pg96.20)
CSM_nil_cursor	(pg97.34)

+++ appendix +++

CSM_nil_data_pointer	(pg96.17)
CSM_no_ageable_docs	(pg97.37)
CSM_no_local_tape	(pg99.24)
CSM_no_permission	(pg95.37)
CSM_no_refcnts	(pg98.21)
CSM_no_resources	(pg95.40)
CSM_no_resources_in_target_cache	(pg99.36)
CSM_no_such_cache	(pg95.47)
CSM_no_such_known_cache	(pg97.2)
CSM_no_such_object	(pg96.2)
CSM_no_temp_ids	(pg98.9)
CSM_not_a_legal_cache	(pg96.23)
CSM_not_available	(pg96.47)
CSM_not_implemented	(pg98.33)
CSM_object_access_mode	(pg92.19)
CSM_object_attr_struct	(pg94.20)
CSM_object_attr_typ	(pg94.35)
CSM_object_breakup_struct	(pg95.1)
CSM_object_breakup_typ	(pg95.8)
CSM_object_busy	(pg96.5)
CSM_object_desc_struct	(pg94.12)
CSM_object_desc_typ	(pg94.16)
CSM_object_exists_in_target_cache	(pg99.37)
CSM_object_handle_typ	(pg93.23)
CSM_open_access_mode	(pg98.30)
CSM_other_error	(pg96.8)
CSM_out_of_range	(pg96.11)
CSM_overflow_breakup_object	(pg99.27)
CSM_overlap_breakup_object	(pg99.30)
CSM_refcnt_incremented	(pg97.45)
CSM_refcnt_overflow	(pg97.42)
CSM_session_handle_typ	(pg93.24)
CSM_temp_busy	(pg98.36)
CSM_tool_update_info	(pg99.18)
CSM_tools_unavailable	(pg96.35)
CSM_trace_info	(pg99.21)
CSM_trace_typ	(pg93.20)
CSM_undefined_procedure	(pg98.6)
DOC_ASCENDING	(pg126.21)
DOC_BATCH_PAGE_OVERHEAD	(pg129.41)
DOC_BS_BATCHNAME_PFX	(pg130.5)
DOC_COLD_BATCHNAME_PFX	(pg130.6)
DOC_COMMIT_OPTION_MIGRATE_DELAY	(pg130.24)
DOC_DEFAULT_INBOX_PRIORITY	(pg119.20)
DOC_DESCENDING	(pg126.23)
DOC_EXACT_ASCENDING	(pg126.25)
DOC_EXACT_DESCENDING	(pg126.27)
DOC_EXPORT_LOG_LEVEL	(pg131.10)

+++ appendix +++

DOC_IN_DRIVE	(pg119.26)
DOC_IN_OSAR	(pg119.28)
DOC_MAX_ANNOTATION_LEN	(pg128.4)
DOC_MAX_ASYNC_MIGRATES	(pg119.12)
DOC_MAX_BATCHNAME_LEN	(pg129.46)
DOC_MAX_CE_RETURN_REC	(pg128.12)
DOC_MAX_COMMIT_RECS	(pg131.8)
DOC_MAX_DELETIONS	(pg119.8)
DOC_MAX_DOC_SURF_LOCS	(pg119.13)
DOC_MAX_ID_ALLOCATION	(pg119.9)
DOC_MAX_INBOX_PRIORITY	(pg119.21)
DOC_MAX_PAGES_PER_BATCH	(pg119.11)
DOC_MAX_PREFETCHES	(pg119.10)
DOC_MAX_QUEUE_NAME_LEN	(pg126.37)
DOC_NOT_IN_OSAR	(pg119.30)
DOC_NOT_ON_OPT_DISK	(pg119.31)
DOC_NO_QUEUE_NUM	(pg126.36)
DOC_PRIMARY_STATUS_ATTRIBUTE	(pg128.39)
DOC_PRIO_HIGH	(pg126.33)
DOC_PRIO_LOW	(pg126.31)
DOC_PRIO_MID	(pg126.32)
DOC_PSEUDO_REQUEST_ID	(pg119.15)
DOC_SECONDARY_STATUS_ATTRIBUTE	(pg128.40)
DOC_SECURITY_ATTRIBUTE	(pg128.38)
DOC_SYS_BATCHNAME_PFX	(pg130.4)
DOC_XIN_PROTOCOL	(pg131.12)
DOC_XNOT_IN_PROTOCOL	(pg131.13)
DOC_annotation_desc_typ	(pg128.34)
DOC_annotation_id_typ	(pg128.1)
DOC_annotation_typ	(pg128.3)
DOC_async_comp_typ	(pg127.15)
DOC_async_rcv_doc_typ	(pg127.43)
DOC_async_xmit_doc_typ	(pg127.27)
DOC_ce_action_t	(pg128.10)
DOC_clientattr_typ	(pg130.13)
DOC_cmt_opt_typ	(pg130.23)
DOC_commit_option_typ	(pg130.40)
DOC_committal_desc_typ	(pg126.1)
DOC_cursurf_typ	(pg126.7)
DOC_defs	(pg119.1)
DOC_delete_action	(pg128.8)
DOC_doc_attribute_value_typ	(pg129.4)
DOC_doc_location_desc_typ	(pg120.47)
DOC_doc_location_w_sds_desc_typ	(pg121.40)
DOC_doc_surf_loc_typ	(pg120.5)
DOC_err_CFS_annot_DN_not_found	(pg140.8)
DOC_err_CFS_annot_action_failed	(pg140.16)
DOC_err_CFS_annot_doc_not_found	(pg139.44)

+++ appendix +++

DOC_err_CFS_wrong_annot_log_level	(pg140.2)
DOC_err_annotation_already_exists	(pg139.29)
DOC_err_annotation_busy	(pg136.48)
DOC_err_annotation_id_invalid	(pg139.32)
DOC_err_annotation_not_busy	(pg137.9)
DOC_err_annotation_not_found	(pg136.42)
DOC_err_annotation_too_large	(pg136.39)
DOC_err_archive_in_progress	(pg134.13)
DOC_err_bad_attribute_type	(pg133.5)
DOC_err_bad_cache_to_use	(pg137.29)
DOC_err_bad_document_status	(pg133.8)
DOC_err_bad_entry_point	(pg134.7)
DOC_err_bad_handle	(pg135.17)
DOC_err_bad_index_value_type	(pg133.11)
DOC_err_bad_session_type	(pg133.20)
DOC_err_bad_version	(pg132.35)
DOC_err_bad_write	(pg134.23)
DOC_err_badinterleave	(pg133.32)
DOC_err_batch_already_open	(pg134.19)
DOC_err_batch_not_open	(pg134.42)
DOC_err_cache_not_local	(pg137.25)
DOC_err_cancel_detected	(pg137.47)
DOC_err_cannot_find_named_service	(pg133.14)
DOC_err_cant_change_disk_type	(pg134.4)
DOC_err_cant_change_is_primary	(pg134.1)
DOC_err_cant_decrease_preflib_curwrtsurfsnum	(pg139.20)
DOC_err_cant_lock	(pg135.9)
DOC_err_cant_startup_daemon	(pg138.14)
DOC_err_circular_commit	(pg138.37)
DOC_err_commit_not_remote	(pg138.45)
DOC_err_connection_already_open	(pg132.44)
DOC_err_connection_not_open	(pg132.47)
DOC_err_csum_inconsistent	(pg135.12)
DOC_err_document_already_migrated	(pg135.48)
DOC_err_document_not_found	(pg135.36)
DOC_err_dup_in_fast_batch	(pg135.20)
DOC_err_duplicate_doc_id	(pg136.18)
DOC_err_duplicate_family	(pg136.21)
DOC_err_id_wrap_around	(pg133.2)
DOC_err_interleave_single_side	(pg135.27)
DOC_err_interleaveone	(pg133.35)
DOC_err_internal_rpc_error	(pg132.38)
DOC_err_intervention_required	(pg139.16)
DOC_err_invalid_cache	(pg135.45)
DOC_err_invalid_doc_id	(pg134.36)
DOC_err_invalid_duration	(pg138.20)
DOC_err_invalid_family_id	(pg137.15)
DOC_err_invalid_family_name	(pg135.39)

+++ appendix +++

DOC_err_invalid_image_id	(pg136.12)
DOC_err_invalid_index_type	(pg134.48)
DOC_err_invalid_network_address	(pg136.3)
DOC_err_invalid_num_pages	(pg135.3)
DOC_err_invalid_number_of_osars	(pg137.3)
DOC_err_invalid_number_of_surfaces	(pg137.6)
DOC_err_invalid_option	(pg138.17)
DOC_err_invalid_osar_disk_type	(pg133.44)
DOC_err_invalid_osar_id	(pg136.9)
DOC_err_invalid_page_num	(pg134.33)
DOC_err_invalid_relational_operator	(pg133.17)
DOC_err_invalid_request_id	(pg136.6)
DOC_err_invalid_security	(pg136.15)
DOC_err_invalid_session_handle	(pg136.27)
DOC_err_invalid_session_number	(pg133.23)
DOC_err_invalid_surf_id	(pg139.3)
DOC_err_invalid_user_id	(pg136.24)
DOC_err_magic_rm_id_invalid	(pg139.36)
DOC_err_need_disable_surf	(pg139.25)
DOC_err_no_capability	(pg136.45)
DOC_err_no_doc_hdr	(pg134.27)
DOC_err_no_matches	(pg136.30)
DOC_err_no_more_queues	(pg138.5)
DOC_err_no_more_space	(pg134.45)
DOC_err_no_osar	(pg138.40)
DOC_err_no_osar_migrate_off	(pg139.9)
DOC_err_no_permission	(pg135.42)
DOC_err_no_resource	(pg137.12)
DOC_err_not_compatible_ssn	(pg138.29)
DOC_err_not_debugging	(pg132.41)
DOC_err_not_impl_archive	(pg140.23)
DOC_err_not_impl_pdb	(pg140.22)
DOC_err_not_impl_pdb_or_archive	(pg138.23)
DOC_err_not_in_osar	(pg137.18)
DOC_err_not_legal	(pg134.16)
DOC_err_not_on_tranlog	(pg138.33)
DOC_err_other_error	(pg135.33)
DOC_err_page_out_of_range	(pg136.36)
DOC_err_parameter_rm_invalid	(pg139.40)
DOC_err_pgm_err	(pg139.6)
DOC_err_primary_cant_be_tranlog	(pg133.41)
DOC_err_qname_too_long	(pg138.2)
DOC_err_queue_inuse	(pg138.11)
DOC_err_queue_not_allocated	(pg138.8)
DOC_err_req_cancelled	(pg135.6)
DOC_err_server_not_found	(pg133.29)
DOC_err_service_not_local	(pg137.35)
DOC_err_status_not_changeable	(pg137.22)

+++ appendix +++

DOC_err_timeout	(pg137.38)
DOC_err_too_many_commit_families	(pg138.26)
DOC_err_too_many_ids_requested	(pg136.33)
DOC_err_too_many_migrates	(pg137.44)
DOC_err_too_many_pages	(pg134.39)
DOC_err_too_many_prefetches	(pg137.32)
DOC_err_too_many_queues	(pg137.41)
DOC_err_too_many_servers	(pg133.26)
DOC_err_trace_info	(pg139.13)
DOC_err_tranlog_not_defined	(pg133.38)
DOC_err_zero_length_page	(pg135.30)
DOC_export_action	(pg128.7)
DOC_famid_ary	(pg124.20)
DOC_family_commit_typ	(pg131.3)
DOC_family_create_server_typ	(pg124.18)
DOC_family_desc_typ	(pg124.9)
DOC_family_server_desc_typ	(pg123.24)
DOC_inbox_priority_typ	(pg119.22)
DOC_index_value_typ	(pg124.44)
DOC_insert_action	(pg128.6)
DOC_inx_info_typ	(pg130.19)
DOC_iteration_typ	(pg129.19)
DOC_migrate_order_typ	(pg126.20)
DOC_not_cursurf	(pg126.4)
DOC_notify_data_typ	(pg127.3)
DOC_page_cacheloc_typ	(pg129.37)
DOC_page_hdr_typ	(pg129.26)
DOC_page_odloc_struct_typ	(pg129.31)
DOC_page_odloc_typ	(pg129.17)
DOC_page_offset_typ	(pg129.9)
DOC_priority_typ	(pg126.30)
DOC_queue_num_typ	(pg126.35)
DOC_sideA_cursurf	(pg126.5)
DOC_sideB_cursurf	(pg126.6)
DOC_surf_loc_typ	(pg119.34)
DOC_surface_desc_typ	(pg122.37)
DOC_update_action	(pg128.9)
DOC_zCFS_annot_DN_not_found	(pg132.12)
DOC_zCFS_annot_action_failed	(pg132.13)
DOC_zCFS_annot_doc_not_found	(pg132.10)
DOC_zCFS_wrong_annot_log_level	(pg132.11)
DOC_zannotation_already_exists	(pg132.6)
DOC_zannotation_busy	(pg131.42)
DOC_zannotation_id_invalid	(pg132.7)
DOC_zannotation_not_busy	(pg131.45)
DOC_zannotation_not_found	(pg131.40)
DOC_zannotation_too_large	(pg131.39)
DOC_zbad_attribute_type	(pg132.24)

+++ appendix +++

DOC_zbad_cache_to_use	(pg132.2)
DOC_zbad_document_status	(pg132.25)
DOC_zbad_index_value_type	(pg132.26)
DOC_zbad_session_type	(pg132.29)
DOC_zbad_version	(pg132.18)
DOC_zcache_not_local	(pg131.48)
DOC_zcannot_find_named_service	(pg132.27)
DOC_zconnection_already_open	(pg132.21)
DOC_zconnection_not_open	(pg132.22)
DOC_zdocument_already_migrated	(pg131.25)
DOC_zdocument_not_found	(pg131.21)
DOC_zduplicate_doc_id	(pg131.31)
DOC_zduplicate_family	(pg131.32)
DOC_zid_wrap_around	(pg132.23)
DOC_zinvalid_cache	(pg131.24)
DOC_zinvalid_family_id	(pg131.47)
DOC_zinvalid_family_name	(pg131.22)
DOC_zinvalid_image_id	(pg131.29)
DOC_zinvalid_network_address	(pg131.26)
DOC_zinvalid_number_of_osars	(pg131.43)
DOC_zinvalid_number_of_surfaces	(pg131.44)
DOC_zinvalid_osar_id	(pg131.28)
DOC_zinvalid_relational_operator	(pg132.28)
DOC_zinvalid_request_id	(pg131.27)
DOC_zinvalid_security	(pg131.30)
DOC_zinvalid_session_handle	(pg131.34)
DOC_zinvalid_session_number	(pg132.30)
DOC_zinvalid_user_id	(pg131.33)
DOC_zmagic_rm_id_invalid	(pg132.8)
DOC_zno_capability	(pg131.41)
DOC_zno_matches	(pg131.35)
DOC_zno_permission	(pg131.23)
DOC_zno_resource	(pg131.46)
DOC_znot_debugging	(pg132.20)
DOC_znot_impl_pdb	(pg132.5)
DOC_znot_in_osar	(pg131.38)
DOC_zother_error	(pg131.20)
DOC_zpage_out_of_range	(pg131.37)
DOC_zparameter_rm_invalid	(pg132.9)
DOC_zrpc_error	(pg132.19)
DOC_zserver_not_found	(pg132.32)
DOC_zservice_not_local	(pg132.4)
DOC_zstatus_not_changeable	(pg132.1)
DOC_ztoo_many_ids_requested	(pg131.36)
DOC_ztoo_many_prefetches	(pg132.3)
DOC_ztoo_many_servers	(pg132.31)
DTI_28DayRange	(pg180.48)
DTI_29DayRange	(pg181.2)

+++ appendix +++

DTI_AlphaLength	(pg180.34)
DTI_BadMaskDay	(pg180.22)
DTI_BadMaskHour	(pg180.24)
DTI_BadMaskMonth	(pg180.20)
DTI_ConstantFileReadFail	(pg181.26)
DTI_DATE	(pg176.21)
DTI_DATEONLY	(pg176.22)
DTI_DATETIME	(pg176.24)
DTI_DayOfMonthRange	(pg181.4)
DTI_DayOfWeekRange	(pg180.44)
DTI_DayOfYearRange	(pg180.46)
DTI_FNMaskSizeTooSmall	(pg181.36)
DTI_GMT	(pg176.5)
DTI_GeneralRange	(pg181.12)
DTI_GetLocaleInfoFail	(pg181.32)
DTI_HourRange	(pg180.42)
DTI_InconsistantSep	(pg180.26)
DTI_InvalidAmPm	(pg181.14)
DTI_InvalidInput	(pg180.18)
DTI_InvalidMask	(pg180.10)
DTI_InvalidTimeZone	(pg181.16)
DTI_InvalidType	(pg181.10)
DTI_LOCAL	(pg176.4)
DTI_LanguageUnavail	(pg181.18)
DTI_MASKSIZE	(pg179.18)
DTI_MinuteRange	(pg180.40)
DTI_MissingAlpha	(pg180.36)
DTI_MissingNumeric	(pg180.32)
DTI_MonthRange	(pg181.6)
DTI_NODATE	(pg179.2)
DTI_NT2Unix	(pg179.44)
DTI_NoCurrentTime	(pg180.14)
DTI_NoMaskMemory	(pg181.28)
DTI_NoNullTerm	(pg180.28)
DTI_NumericLength	(pg180.30)
DTI_SecondRange	(pg180.38)
DTI_SetConstFail	(pg181.22)
DTI_SetTimeFailed	(pg180.16)
DTI_TIME	(pg176.23)
DTI_TIMEONLY	(pg176.25)
DTI_TooManyMaskItems	(pg180.12)
DTI_TwoWordsReqd	(pg181.30)
DTI_Unix2NT	(pg179.27)
DTI_UnlinkFail	(pg181.24)
DTI_UnsupportedPosixMask	(pg181.34)
DTI_VersionMismatch	(pg181.20)
DTI_YearRange	(pg181.8)
DTI_datetime_typ	(pg176.16)

+++ appendix +++

DTI_defs	(pg175.39)
DTI_dtkind_typ	(pg176.20)
DTI_dtstring_ptr_typ	(pg176.38)
DTI_mask_ptr_typ	(pg178.46)
DTI_tm_struct	(pg179.6)
DTI_tm_struct_typ	(pg179.16)
DTI_zone_typ	(pg176.3)
EQL	(pg484.31)
EXPORT_CFG_FILE_PATH	(pg254.36)
EXPORT_FILENAME	(pg254.37)
ErrEncode_h	(pg493.24)
FALSE	(pg490.46)
FAR	(pg492.19)
FFI_ANNOT_ABS	(pg202.33)
FFI_ANNOT_BRUSH_COLOR	(pg202.35)
FFI_ANNOT_BRUSH_STYLE	(pg202.34)
FFI_ANNOT_CHAR_SET	(pg202.31)
FFI_ANNOT_CHAR_SIZE	(pg202.27)
FFI_ANNOT_LOC	(pg202.26)
FFI_ANNOT_TEXT	(pg202.28)
FFI_ANNOT_TEXT_BCOLOR	(pg202.29)
FFI_ANNOT_TEXT_FCOLOR	(pg202.30)
FFI_ANNOT_VAR	(pg202.32)
FFI_BLACK_COLOR_IS_0	(pg198.17)
FFI_BLACK_COLOR_IS_1	(pg198.18)
FFI_CHAR_SET_8859_1	(pg196.43)
FFI_CHAR_SET_8859_2	(pg196.44)
FFI_CHAR_SET_8859_3	(pg196.45)
FFI_CHAR_SET_8859_4	(pg196.46)
FFI_CHAR_SET_8859_5	(pg196.47)
FFI_CHAR_SET_8859_6	(pg196.48)
FFI_CHAR_SET_8859_7	(pg197.1)
FFI_CHAR_SET_8859_8	(pg197.2)
FFI_CHAR_SET_8859_9	(pg197.3)
FFI_CHAR_SET_ARABIC	(pg196.42)
FFI_CHAR_SET_EUC_J	(pg197.5)
FFI_CHAR_SET_FN_INTERNAT	(pg196.41)
FFI_CHAR_SET_KSC_5601	(pg197.6)
FFI_CHAR_SET_SHIFT_JIS	(pg197.4)
FFI_FILLORDER_LSB_FIRST	(pg198.4)
FFI_FILLORDER_MSB_FIRST	(pg198.3)
FFI_FMT_CALS	(pg195.42)
FFI_FMT_FILENET	(pg195.41)
FFI_FMT_TIFF	(pg195.43)
FFI_HDR_CHAR_SET	(pg200.46)
FFI_ICMP_G3	(pg196.9)
FFI_ICMP_G4	(pg196.11)
FFI_ICMP_MAX	(pg196.14)

+++ appendix +++

FFI_IDECODE_BITMAP_FILLORDER	(pg199.43)
FFI_IDECODE_BLACK_COLOR	(pg199.44)
FFI_IDECODE_REDUCE	(pg199.45)
FFI_IENCODE_BITMAP_FILLORDER	(pg199.3)
FFI_IENCODE_HRES	(pg199.2)
FFI_IENCODE_IMAGE_COMPRESSION	(pg198.46)
FFI_IENCODE_IMAGE_FORMAT	(pg198.45)
FFI_IENCODE_TILED	(pg198.47)
FFI_IENCODE_VRES	(pg199.1)
FFI_IMAGE_DESC_VERSION	(pg196.30)
FFI_IMAGE_PAGE_TYPE	(pg197.27)
FFI_MAX_IMAGE_FORMAT	(pg195.45)
FFI_MIN_PAGE_BYTES	(pg197.30)
FFI_PAGE_DESC_VERSION	(pg197.42)
FFI_TCMP_FILENET	(pg196.36)
FFI_TCMP_MAX	(pg196.38)
FFI_TCMP_NONE	(pg196.35)
FFI_TENCODE_CHAR_SET	(pg200.14)
FFI_TENCODE_TEXT_COMPRESSION	(pg200.15)
FFI_TEXT_DESC_VERSION	(pg197.22)
FFI_TEXT_PAGE_TYPE	(pg197.28)
FFI_annot_attr_typ	(pg202.25)
FFI_annot_brush_typ	(pg202.10)
FFI_annot_color_typ	(pg202.8)
FFI_annot_loc_typ	(pg201.22)
FFI_annot_opt_typ	(pg203.17)
FFI_annot_size_typ	(pg201.31)
FFI_annot_text_typ	(pg201.44)
FFI_black_color_typ	(pg198.16)
FFI_buf_typ	(pg195.34)
FFI_char_set_typ	(pg196.40)
FFI_defs	(pg195.16)
FFI_err_annot_too_long	(pg204.22)
FFI_err_ano_dec	(pg204.31)
FFI_err_ano_free	(pg204.34)
FFI_err_badifacelevel	(pg204.28)
FFI_err_input_underflow	(pg203.23)
FFI_err_invalid_annot_len	(pg204.18)
FFI_err_invalid_annot_option	(pg204.25)
FFI_err_invalid_black	(pg204.6)
FFI_err_invalid_char_set	(pg204.12)
FFI_err_invalid_fillorder	(pg204.3)
FFI_err_invalid_image_comp	(pg204.15)
FFI_err_invalid_image_format	(pg203.48)
FFI_err_invalid_option	(pg203.45)
FFI_err_invalid_reduce_option	(pg203.39)
FFI_err_invalid_tcmp_option	(pg203.42)
FFI_err_invalid_text_comp	(pg204.9)

+++ appendix +++

FFI_err_output_overflow	(pg203.27)
FFI_err_unknown_page_type	(pg203.31)
FFI_err_wrong_page_type	(pg203.35)
FFI_fillorder_typ	(pg198.2)
FFI_hdr_attr_typ	(pg200.45)
FFI_hdr_opt_typ	(pg201.8)
FFI_idecode_attr_typ	(pg199.42)
FFI_idecode_opt_typ	(pg200.9)
FFI_iencode_attr_typ	(pg198.44)
FFI_iencode_opt_typ	(pg199.22)
FFI_image_comp_typ	(pg196.8)
FFI_image_desc_typ	(pg196.28)
FFI_image_format_typ	(pg195.40)
FFI_pad_typ	(pg198.10)
FFI_page_desc_typ	(pg197.40)
FFI_page_type_typ	(pg197.26)
FFI_quick_reduce_typ	(pg199.38)
FFI_rect_loc_typ	(pg198.28)
FFI_rect_size_typ	(pg198.34)
FFI_rectangle_typ	(pg198.40)
FFI_resolution_typ	(pg196.1)
FFI_tdecode_attr_typ	(pg200.32)
FFI_tdecode_opt_typ	(pg200.41)
FFI_tencode_attr_typ	(pg200.13)
FFI_tencode_opt_typ	(pg200.26)
FFI_text_comp_typ	(pg196.34)
FFI_text_desc_typ	(pg197.20)
FFI_var_data_typ	(pg202.20)
FN_DONT_SHOW_UI	(pg490.14)
FN_ENTRY	(pg490.39)
FN_ENTRY	(pg490.39)
FN_ERR_CRITICAL	(pg500.21)
FN_ERR_INFO	(pg499.38)
FN_ERR_OPERATOR	(pg500.8)
FN_ERR_SERIOUS	(pg500.15)
FN_ERR_WARNING	(pg499.46)
FN_GUID_char16_t	(pg492.10)
FN_GUID_char8_t	(pg491.48)
FN_GUID_t	(pg491.35)
FN_SHOW_UI	(pg490.13)
FN_bool16_t	(pg491.12)
FN_bool_t	(pg491.11)
FN_byte_t	(pg491.10)
FN_int16_t	(pg491.6)
FN_int32_t	(pg491.8)
FN_int64_t	(pg491.17)
FN_int8_t	(pg491.4)
FN_uint16_t	(pg491.7)

+++ appendix +++

FN_uint32_t	(pg491.9)
FN_uint64_t	(pg491.18)
FN_uint8_t	(pg491.5)
FP_BadAlign	(pg213.28)
FP_BadDigits	(pg213.18)
FP_BadMask	(pg213.10)
FP_BadNum	(pg213.22)
FP_BadStyle	(pg213.14)
FP_DOMESTIC	(pg211.37)
FP_EUROPEAN	(pg211.38)
FP_IlglVers	(pg213.35)
FP_IllegalAlign	(pg213.29)
FP_IllegalFmt	(pg213.6)
FP_IllegalNum	(pg213.23)
FP_InvalidDigits	(pg213.19)
FP_InvalidMask	(pg213.11)
FP_InvalidStyle	(pg213.15)
FP_Overflow	(pg212.41)
FP_Undefined	(pg212.45)
FP_badVersion	(pg213.36)
FP_bcd_style_typ	(pg212.35)
FP_defs	(pg211.6)
FP_expchars	(pg211.27)
FP_expoffset	(pg211.26)
FP_ilglfmt	(pg213.5)
FP_manchars	(pg211.25)
FP_manoffset	(pg211.24)
FP_mode_typ	(pg211.36)
FP_num_abs	(pg213.47)
FP_num_add	(pg213.42)
FP_num_dbl2num	(pg214.13)
FP_num_divide	(pg213.45)
FP_num_long2num	(pg214.2)
FP_num_multiply	(pg213.44)
FP_num_neg	(pg213.46)
FP_num_num2dbl	(pg214.12)
FP_num_num2long	(pg214.1)
FP_num_num2mask	(pg214.11)
FP_num_num2ora	(pg214.8)
FP_num_num2sci	(pg214.10)
FP_num_num2str	(pg214.4)
FP_num_num2unslong	(pg214.14)
FP_num_ora2num	(pg214.9)
FP_num_round	(pg214.5)
FP_num_rounddisp	(pg214.6)
FP_num_setmode	(pg214.7)
FP_num_str2num	(pg214.3)
FP_num_subtract	(pg213.43)

+++ appendix +++

FP_num_trunc	(pg213.48)
FP_num_unslong2num	(pg214.15)
FP_number	(pg211.20)
FP_numlongs	(pg211.19)
FP_oflo	(pg212.40)
FP_pointer_typ	(pg211.21)
FP_strchars	(pg211.28)
FP_string	(pg211.29)
FP_undeflong	(pg211.48)
FP_undefopnd	(pg212.44)
FP_xmain	(pg213.40)
FileNet_h	(pg490.31)
GEQ	(pg484.30)
GTR	(pg484.29)
GUID_REGISTRY_FORMAT_LENGTH	(pg239.4)
IMS_bes_svc_typ	(pg223.33)
IMS_csm_svc_typ	(pg223.29)
IMS_defs	(pg223.20)
IMS_doc_svc_typ	(pg223.30)
IMS_err_invalid_IMS_handle	(pg223.39)
IMS_err_invalid_IMS_service_type	(pg223.42)
IMS_err_not_impl	(pg224.3)
IMS_err_release_failed	(pg223.45)
IMS_err_reserve_failed	(pg223.48)
IMS_inx_svc_typ	(pg223.31)
IMS_num_svc_typs	(pg223.28)
IMS_pri_svc_typ	(pg223.32)
IMS_sec_svc_typ	(pg223.35)
IMS_sqi_svc_typ	(pg223.36)
IMS_svc_typ	(pg223.27)
IMS_wqs_svc_typ	(pg223.34)
INXD_err_lock_conflict	(pg263.40)
INX_ANNOT_DOC	(pg234.18)
INX_BEGINNING_OF_FOLDER	(pg246.27)
INX_BOOLOP_AND	(pg249.26)
INX_BOOLOP_NOT	(pg249.28)
INX_BOOLOP_OR	(pg249.27)
INX_CAP_AVAIL	(pg240.13)
INX_CAP_CLU	(pg240.18)
INX_CAP_DCL	(pg240.17)
INX_CAP_DCL_UNLOCK	(pg240.22)
INX_CAP_DOC	(pg240.14)
INX_CAP_FLD	(pg240.15)
INX_CAP_FORM_MENU	(pg240.20)
INX_CAP_FORM_VALTAB	(pg240.21)
INX_CAP_IDX	(pg240.16)
INX_CAP_TAB	(pg240.19)
INX_CEID_MIN	(pg240.5)

+++ appendix +++

INX_CEID_UNDEFINED	(pg240.4)
INX_CE_GUID_LENGTH	(pg253.17)
INX_CE_NULL_GUID	(pg253.18)
INX_CL_ACTIVE	(pg248.24)
INX_CL_ALL	(pg248.26)
INX_CL_CLOSED	(pg248.25)
INX_DICT_LEVEL	(pg233.44)
INX_DOC_DESC_LEVEL	(pg240.43)
INX_DUPS_FEW	(pg238.41)
INX_DUPS_MANY	(pg238.40)
INX_DUPS_NONE	(pg238.42)
INX_END_OF_FOLDER	(pg246.28)
INX_EXPORT_LOG_LEVEL	(pg254.4)
INX_FOLDER_SEPARATOR	(pg247.14)
INX_FORM_DOC	(pg234.15)
INX_IMAGE_DOC	(pg234.13)
INX_INVALID_CLUSTER_KEY	(pg243.41)
INX_INVALID_CLUSTER_SPACE	(pg243.34)
INX_INVALID_INDEX_ID	(pg235.35)
INX_IN_ACCESSRIGHTS	(pg241.42)
INX_IN_ACCESSRIGHTS_AX	(pg241.48)
INX_IN_ACCESSRIGHTS_RD	(pg241.46)
INX_IN_ACCESSRIGHTS_WR	(pg241.47)
INX_IN_ANNOTATIONFLAG	(pg241.34)
INX_IN_ARCHIVEDATE	(pg241.35)
INX_IN_CE_OSID	(pg241.45)
INX_IN_CLOSED	(pg242.2)
INX_IN_DELETEDATE	(pg241.36)
INX_IN_DOCCLASSNUMBER	(pg241.32)
INX_IN_DOCFORMAT	(pg241.43)
INX_IN_DOCLOCATION	(pg241.44)
INX_IN_DOCNUMBER	(pg241.31)
INX_IN_DOCTYPE	(pg241.41)
INX_IN_ENTRYDATE	(pg241.33)
INX_IN_FIRST_USER	(pg242.7)
INX_IN_LAST_SYS	(pg242.5)
INX_IN_PAGES	(pg241.40)
INX_IN_RETENTBASE	(pg241.37)
INX_IN_RETENTDISP	(pg241.38)
INX_IN_RETENTOFFSET	(pg241.39)
INX_KEY_RANGE	(pg248.33)
INX_KEY_SIMPLE	(pg248.32)
INX_LEN_BOOLEAN	(pg236.41)
INX_LEN_BYTE	(pg236.42)
INX_LEN_DATE	(pg237.3)
INX_LEN_FPNUM	(pg236.48)
INX_LEN_LENGTH	(pg237.2)
INX_LEN_LONG	(pg236.46)

+++ appendix +++

INX_LEN_MENU	(pg237.5)
INX_LEN_NULL	(pg237.6)
INX_LEN_SHORT	(pg236.44)
INX_LEN_TIME	(pg237.4)
INX_LEN_UN_ BYTE	(pg236.43)
INX_LEN_UN_ LONG	(pg236.47)
INX_LEN_UN_ SHORT	(pg236.45)
INX_LEN_VARIABLE	(pg237.7)
INX_LONGER_STRING	(pg237.9)
INX_LONG_STRING	(pg237.8)
INX_MAXOP	(pg249.30)
INX_MAX_APERCARDFILE_LEN	(pg245.43)
INX_MAX_APERCARDIXLOC_LEN	(pg245.47)
INX_MAX_CE_DIR_LOG_SIZE	(pg253.15)
INX_MAX_CE_DOMAIN_NAME	(pg253.12)
INX_MAX_CE_HOST_ID	(pg253.13)
INX_MAX_CE_OBJECT_STORE_NAME	(pg253.14)
INX_MAX_CE_RETURN_REC	(pg253.16)
INX_MAX_DBCOLS	(pg244.42)
INX_MAX_DCL_INDEXES	(pg244.40)
INX_MAX_DOC_DCL_ID_REC	(pg252.44)
INX_MAX_FAM_NAME_LEN	(pg243.23)
INX_MAX_FILTER_LEN	(pg252.46)
INX_MAX_FLD_COMP_NAME	(pg246.25)
INX_MAX_FLD_LEVELS	(pg246.24)
INX_MAX_FLD_NAME	(pg246.26)
INX_MAX_FORM_MENU_DESC_LEN	(pg251.35)
INX_MAX_FORM_MENU_ITEM_DESC_LEN	(pg251.34)
INX_MAX_FORM_VALTAB_DESC_LEN	(pg252.14)
INX_MAX_FORM_VALTAB_NAME_LEN	(pg252.13)
INX_MAX_IINVALIDATION_LEN	(pg235.37)
INX_MAX_KEY_COND	(pg249.4)
INX_MAX_KEY_INDEXES	(pg242.17)
INX_MAX_KEY_LEN	(pg252.45)
INX_MAX_MENU_ITEM_NAME_LEN	(pg238.47)
INX_MAX_MENU_NAME_LEN	(pg238.44)
INX_MAX_OPT_DE	(pg244.10)
INX_MAX_TAB_LENGTH	(pg251.14)
INX_MAX_VALTAB_ITEM_CODE_LEN	(pg252.15)
INX_MAX_VALTAB_ITEM_DESC_LEN	(pg252.16)
INX_MISCOP_HALT	(pg249.29)
INX_MIXED_DOC	(pg234.16)
INX_NULL_DOC	(pg234.12)
INX_OPT_DE_BATCH_TOTAL	(pg244.15)
INX_OPT_DE_VFY_IMAGES	(pg244.13)
INX_OPT_DE_VFY_INDEXES	(pg244.14)
INX_OTHER_DOC	(pg234.19)
INX_OT_CHAR	(pg236.20)

+++ appendix +++

INX_OT_DATE	(pg236.28)
INX_OT_FIXED	(pg236.30)
INX_OT_FPNUM	(pg236.29)
INX_OT_NUM_DEPRECATED	(pg236.21)
INX_OT_OTHER	(pg236.19)
INX_P8_MODEL_DOC	(pg234.20)
INX_QUERY_FORWARD	(pg248.10)
INX_REC_BEGINNING	(pg248.16)
INX_REC_ENDING	(pg248.17)
INX_RELOP_EQ	(pg249.21)
INX_RELOP_GEQ	(pg249.22)
INX_RELOP_GTR	(pg249.23)
INX_RELOP_LEQ	(pg249.20)
INX_RELOP_LIKE	(pg249.25)
INX_RELOP_LSS	(pg249.19)
INX_RELOP_NEQ	(pg249.24)
INX_ROOT_FOLDER_NAME	(pg247.13)
INX_SEP_SHEET	(pg234.17)
INX_SERVICE_STR_LEN	(pg249.8)
INX_SRC_APERTURE	(pg244.38)
INX_SRC_MANUAL	(pg244.36)
INX_SRC_OCR	(pg244.37)
INX_SRC_UNKNOWN	(pg244.35)
INX_TEXT_DOC	(pg234.14)
INX_VALOP_CONST	(pg249.18)
INX_VALOP_INDEX	(pg249.17)
INX_VAL_ALL	(pg251.10)
INX_VAL_DCL	(pg251.9)
INX_VAL_DEF	(pg251.8)
INX_VARYING	(pg233.31)
INX_VT_ASCII	(pg236.9)
INX_VT_BOOLEAN	(pg236.1)
INX_VT_BYTE	(pg236.2)
INX_VT_DATE	(pg236.10)
INX_VT_FPNUM	(pg236.8)
INX_VT_LONG	(pg236.6)
INX_VT_MENU	(pg236.12)
INX_VT_NULL	(pg236.13)
INX_VT_SHORT	(pg236.4)
INX_VT_TIME	(pg236.11)
INX_VT_UNUS_BYTE	(pg236.3)
INX_VT_UNUS_LONG	(pg236.7)
INX_VT_UNUS_SHORT	(pg236.5)
INX_XIN_PROTOCOL	(pg255.21)
INX_ZOTHER_ERROR	(pg255.22)
INX_aperfile_desc_typ	(pg246.11)
INX_aperfile_name_typ	(pg245.45)
INX_aperixloc_data_typ	(pg245.48)

+++ appendix +++

INX_aperixloc_desc_typ	(pg246.18)
INX_capability_typ	(pg240.30)
INX_ce_action_delete	(pg253.38)
INX_ce_action_delete_doctaba_export	(pg253.37)
INX_ce_action_export	(pg253.35)
INX_ce_action_insert	(pg253.34)
INX_ce_action_t	(pg253.33)
INX_ce_action_update	(pg253.36)
INX_ce_complete_t	(pg254.2)
INX_ce_dcl_info_guid_t	(pg254.32)
INX_ce_domain_name_t	(pg253.20)
INX_ce_host_id_t	(pg253.21)
INX_ce_id_t	(pg240.6)
INX_ce_log_seq_t	(pg253.30)
INX_ce_new_dir_offset	(pg254.14)
INX_ce_objectstore_name_t	(pg253.22)
INX_ce_os_info_t	(pg254.24)
INX_ce_rec_list_t	(pg254.12)
INX_closed_typ	(pg248.23)
INX_cluster_desc_typ	(pg244.3)
INX_cluster_id_typ	(pg243.32)
INX_cluster_key_typ	(pg243.39)
INX_cluster_space_typ	(pg243.31)
INX_dcl_apercardfile_typ	(pg245.44)
INX_dcl_apercardixloc_typ	(pg246.5)
INX_dcl_desc_typ	(pg245.41)
INX_dcl_id_typ	(pg235.41)
INX_dcl_index_desc_typ	(pg244.32)
INX_dcl_name_typ	(pg239.47)
INX_defs	(pg233.11)
INX_dict_id_typ	(pg234.5)
INX_dir_offset	(pg242.14)
INX_dir_size	(pg238.20)
INX_dir_typ	(pg238.18)
INX_doc_class_export_t	(pg255.4)
INX_doc_dcl_id_typ	(pg253.3)
INX_doc_desc_typ	(pg241.16)
INX_doc_format_typ	(pg234.24)
INX_doc_index_rec_typ	(pg238.17)
INX_doc_location_typ	(pg234.25)
INX_doc_type_typ	(pg234.11)
INX_dups_typ	(pg238.39)
INX_err	(pg255.19)
INX_err_IA_config_map	(pg268.37)
INX_err_IA_doc_not_processed	(pg268.42)
INX_err_actlog_count	(pg265.28)
INX_err_already_in_folder	(pg256.13)
INX_err_already_inverted	(pg259.31)

+++ appendix +++

INX_err_arc_in_progress (pg261.25)
INX_err_bad_cap_type (pg258.22)
INX_err_bad_current_rec (pg257.40)
INX_err_bad_db (pg262.40)
INX_err_bad_direction (pg257.37)
INX_err_bad_folder_atts (pg258.31)
INX_err_bad_folder_name (pg258.37)
INX_err_bad_fork (pg262.37)
INX_err_bad_menu_name (pg261.7)
INX_err_bad_menu_numitems (pg261.22)
INX_err_bad_move_sys_col (pg263.19)
INX_err_bad_net_data (pg262.31)
INX_err_bad_new_row_buf (pg263.22)
INX_err_bad_pages (pg257.10)
INX_err_bad_row_buf (pg260.16)
INX_err_bad_user_type (pg259.10)
INX_err_bad_valtab_name (pg261.13)
INX_err_bad_valtab_numitems (pg261.19)
INX_err_bad_value_type (pg257.31)
INX_err_bad_version (pg263.1)
INX_err_cannot_change_dcl (pg258.19)
INX_err_cannot_convert (pg262.25)
INX_err_ce_DIR_too_big (pg265.34)
INX_err_ce_append_status_export_xml (pg268.11)
INX_err_ce_array_out_of_bound (pg266.22)
INX_err_ce_bad_signal (pg266.31)
INX_err_ce_completion_recs_not_consistent_w_info (pg267.4)
INX_err_ce_dcl_default_os_not_config (pg268.18)
INX_err_ce_default_IS_not_found (pg266.1)
INX_err_ce_default_IS_value_not_found (pg266.4)
INX_err_ce_delete_default_os (pg267.33)
INX_err_ce_delete_export_xml (pg268.8)
INX_err_ce_doc_class_failed (pg266.34)
INX_err_ce_domain_or_object_store_name_change (pg267.30)
INX_err_ce_duplicate_export_range (pg266.28)
INX_err_ce_export_file_not_found (pg265.46)
INX_err_ce_export_xml_not_found (pg267.47)
INX_err_ce_id_not_found (pg267.9)
INX_err_ce_import_agent_info_timeout (pg267.18)
INX_err_ce_index_server_address_not_found (pg266.10)
INX_err_ce_index_server_desc_not_found (pg266.7)
INX_err_ce_index_server_not_local (pg266.13)
INX_err_ce_info_doc_already_exported (pg266.37)
INX_err_ce_invalid_action (pg267.15)
INX_err_ce_invalid_arguments (pg265.43)
INX_err_ce_invalid_domain_name (pg268.25)
INX_err_ce_invalid_guid (pg267.43)
INX_err_ce_invalid_parameter (pg267.12)

+++ appendix +++

INX_err_ce_invalid_update	(pg267.21)
INX_err_ce_invalid_xml_tag_value	(pg266.25)
INX_err_ce_no_cached_info	(pg266.43)
INX_err_ce_no_completion_recs	(pg266.47)
INX_err_ce_no_export_data	(pg268.5)
INX_err_ce_object_guid_duplicate	(pg267.26)
INX_err_ce_only_one_ce_export	(pg265.37)
INX_err_ce_open_export_xml	(pg268.2)
INX_err_ce_os_dcl_map_not_found	(pg267.38)
INX_err_ce_os_guid_not_config	(pg266.40)
INX_err_ce_rm_bad_magic	(pg268.22)
INX_err_ce_rm_filter_off	(pg268.14)
INX_err_ce_signals_undefined	(pg266.16)
INX_err_ce_timezone_error	(pg268.31)
INX_err_ce_unexpected_xml_format	(pg266.19)
INX_err_ce_user_is_not_admin	(pg265.40)
INX_err_change_dcl_id	(pg259.1)
INX_err_change_dcl_name	(pg259.4)
INX_err_cluster_exists	(pg259.13)
INX_err_cluster_in_use	(pg261.40)
INX_err_cluster_not_zero_base	(pg261.28)
INX_err_cor_reject_msg	(pg262.14)
INX_err_date_conflict	(pg256.46)
INX_err_dcl_index_twice	(pg257.13)
INX_err_descendent_dest	(pg256.22)
INX_err_dict_level_wrong	(pg264.12)
INX_err_dict_not_found	(pg263.16)
INX_err_doc_id_range	(pg257.7)
INX_err_doc_is_filed	(pg257.34)
INX_err_doc_is_tabbed	(pg259.43)
INX_err_double_bg	(pg263.31)
INX_err_dup_dcl	(pg259.7)
INX_err_dup_record	(pg255.40)
INX_err_duplicate_aperfile	(pg260.46)
INX_err_duplicate_aperixloc	(pg261.1)
INX_err_duplicate_index	(pg259.25)
INX_err_duplicate_validation	(pg260.43)
INX_err_exceed_page_size	(pg264.1)
INX_err_folder_closed	(pg258.10)
INX_err_h	(pg255.17)
INX_err_have_conn	(pg263.7)
INX_err_import_buf_len	(pg263.34)
INX_err_incomplete_dcl	(pg259.22)
INX_err_index_aperture_differ	(pg260.22)
INX_err_index_in_rec_twice	(pg257.25)
INX_err_index_not_in_dcl	(pg258.16)
INX_err_info	(pg262.17)
INX_err_internal	(pg262.43)

+++ appendix +++

INX_err_interrupted	(pg258.13)
INX_err_inv_cluster_col	(pg258.43)
INX_err_inv_dcl_name	(pg258.7)
INX_err_inv_ora_typ_idx	(pg260.19)
INX_err_inv_proc_num	(pg263.10)
INX_err_inv_query	(pg256.16)
INX_err_inv_query_archive	(pg260.31)
INX_err_inv_record	(pg256.28)
INX_err_inv_retent_base	(pg257.1)
INX_err_inv_retent_disp	(pg257.28)
INX_err_inv_rpc_msg	(pg263.13)
INX_err_inv_source	(pg257.46)
INX_err_inv_sys_col	(pg257.19)
INX_err_invalid_cap_for_id	(pg259.16)
INX_err_invalid_duplicate	(pg259.40)
INX_err_invalid_handle	(pg255.31)
INX_err_invalid_index_name	(pg260.10)
INX_err_invalid_page_size	(pg264.4)
INX_err_invalid_security	(pg260.13)
INX_err_keycondtype	(pg264.33)
INX_err_len_for_ascii_only	(pg260.7)
INX_err_logon_use_sct_names	(pg261.47)
INX_err_mask_not_allowed	(pg260.4)
INX_err_menu_not_allowed	(pg260.1)
INX_err_menu_ref_by_index	(pg262.8)
INX_err_no_aperfile	(pg260.37)
INX_err_no_aperixloc	(pg260.40)
INX_err_no_capability	(pg256.25)
INX_err_no_cluster	(pg258.46)
INX_err_no_conn	(pg263.4)
INX_err_no_dcl	(pg256.31)
INX_err_no_dict_param	(pg262.46)
INX_err_no_folder	(pg256.7)
INX_err_no_form_menu	(pg261.10)
INX_err_no_form_valtab	(pg261.16)
INX_err_no_index	(pg256.34)
INX_err_no_init	(pg262.28)
INX_err_no_key	(pg256.40)
INX_err_no_menu	(pg256.4)
INX_err_no_more_cols	(pg258.28)
INX_err_no_permission	(pg255.34)
INX_err_no_query	(pg256.19)
INX_err_no_record	(pg255.43)
INX_err_no_service	(pg263.25)
INX_err_no_sys_in_dcl	(pg258.40)
INX_err_no_validation	(pg260.34)
INX_err_not_clu_cap	(pg259.19)
INX_err_not_empty	(pg258.34)

+++ appendix +++

INX_err_not_idx_cap	(pg259.28)
INX_err_not_impl	(pg262.22)
INX_err_not_impl_archive	(pg263.43)
INX_err_not_imported	(pg257.4)
INX_err_not_in_folder	(pg256.10)
INX_err_not_inverted	(pg259.37)
INX_err_not_pdb	(pg258.1)
INX_err_old_ims_def	(pg263.37)
INX_err_old_service_def	(pg263.28)
INX_err_other	(pg255.28)
INX_err_pdb_badfcons	(pg265.15)
INX_err_pdb_badoffset	(pg264.24)
INX_err_pdb_badtype	(pg265.9)
INX_err_pdb_fconssz	(pg265.12)
INX_err_pdb_fconser	(pg264.15)
INX_err_pdb_form_menu	(pg264.39)
INX_err_pdb_form_valtab	(pg265.21)
INX_err_pdb_header	(pg264.18)
INX_err_pdb_negchunks	(pg264.21)
INX_err_pdb_nostopper	(pg265.18)
INX_err_pdb_numindexes	(pg264.30)
INX_err_pdb_numpoolsz	(pg264.45)
INX_err_pdb_pushwrong	(pg265.6)
INX_err_pdb_shortread	(pg264.27)
INX_err_pdb_stackoflo	(pg265.3)
INX_err_pdb_stackuflo	(pg264.48)
INX_err_pdb_stgpoolsz	(pg264.42)
INX_err_pdb_tab	(pg264.36)
INX_err_prec_scale_overflow	(pg261.44)
INX_err_query_in_progress	(pg261.37)
INX_err_query_non_index	(pg258.4)
INX_err_query_overflow	(pg263.46)
INX_err_record_busy	(pg255.46)
INX_err_rename_duplicate_menu	(pg262.2)
INX_err_renum_ordinal	(pg261.34)
INX_err_reqd_is_null	(pg256.37)
INX_err_run_out_of_cluster_num	(pg261.31)
INX_err_sct_name_too_long	(pg262.5)
INX_err_session_busy	(pg255.37)
INX_err_share_cnt_ovfl	(pg260.25)
INX_err_share_cnt_udfl	(pg260.28)
INX_err_string_too_long	(pg262.34)
INX_err_sys_idx_not_allowed	(pg259.46)
INX_err_table_busy	(pg259.34)
INX_err_too_deep	(pg258.25)
INX_err_unk_sys_col	(pg257.22)
INX_err_unknown_op	(pg257.43)
INX_err_validation_not_allowed	(pg261.4)

+++ appendix +++

INX_err_valtab_ref_by_index	(pg262.11)
INX_err_wrong_handle	(pg256.43)
INX_err_xs_dcl_indexes	(pg257.16)
INX_fake_sys_col	(pg242.6)
INX_fam_id_typ	(pg243.25)
INX_fam_name_typ	(pg243.24)
INX_fc_doc_ord_item_typ	(pg248.3)
INX_filter_op_code_typ	(pg249.16)
INX_filter_op_typ	(pg250.13)
INX_filter_typ	(pg250.33)
INX_folder_content_item_typ	(pg247.46)
INX_folder_desc_typ	(pg247.41)
INX_folder_name_typ	(pg246.40)
INX_folder_posn_typ	(pg247.21)
INX_folder_spec_typ	(pg247.11)
INX_form_menu_desc_typ	(pg252.11)
INX_form_menu_description_typ	(pg251.39)
INX_form_menu_item_code_typ	(pg251.37)
INX_form_menu_item_desc_typ	(pg251.38)
INX_form_menu_item_typ	(pg251.44)
INX_form_name_typ	(pg243.17)
INX_form_valtab_desc_typ	(pg252.39)
INX_form_valtab_description_typ	(pg252.18)
INX_form_valtab_item_code_typ	(pg252.19)
INX_form_valtab_item_desc_typ	(pg252.20)
INX_form_valtab_item_typ	(pg252.25)
INX_guid_list_typ	(pg239.9)
INX_guids_typ	(pg239.5)
INX_index_choice_typ	(pg238.34)
INX_index_desc_typ	(pg239.34)
INX_index_id_typ	(pg235.34)
INX_index_name_typ	(pg235.31)
INX_index_size	(pg253.9)
INX_index_validation_typ	(pg235.38)
INX_index_value_typ	(pg238.8)
INX_key_comp_typ	(pg248.39)
INX_key_cond_typ	(pg248.45)
INX_key_desc_typ	(pg242.30)
INX_key_id_typ	(pg242.16)
INX_key_name_typ	(pg235.32)
INX_library_errors	(pg265.25)
INX_mask_typ	(pg239.2)
INX_menu_desc_typ	(pg243.13)
INX_menu_id_typ	(pg251.46)
INX_menu_item_name_typ	(pg238.48)
INX_menu_item_typ	(pg243.3)
INX_menu_name_typ	(pg238.45)
INX_menu_typ	(pg243.15)

+++ appendix +++

INX_ora_type_typ	(pg236.17)
INX_query_direct_typ	(pg248.9)
INX_query_match_typ	(pg242.13)
INX_query_miss_doc_typ	(pg251.30)
INX_query_typ	(pg251.4)
INX_relational_op_typ	(pg234.7)
INX_retent_base_typ	(pg234.46)
INX_retent_disp_typ	(pg235.25)
INX_retent_offset_typ	(pg235.4)
INX_session_handle_typ	(pg234.3)
INX_tab_desc_typ	(pg251.25)
INX_validation_desc_typ	(pg239.41)
INX_validation_name_typ	(pg235.39)
INX_valtab_id_typ	(pg252.27)
INX_valtab_name_typ	(pg252.28)
INX_value_type_typ	(pg235.48)
LEQ	(pg484.28)
LOCAL_TRAN	(pg389.10)
LONG_ALIGN_DIR	(pg491.1)
LSS	(pg484.27)
MAX_DMA_DISPLAY_NAME_LENGTH	(pg233.45)
MAX_NUMBER_OF_GUIDS	(pg239.11)
NEQ	(pg484.32)
NOT_WQS_EXCEPTION	(pg462.44)
NO_WINDOWS	(pg211.9)
NULL	(pg490.42)
PASCAL	(pg492.23)
PRI_CLASS_ARRAY	(pg312.11)
PRI_CLASS_LONG	(pg312.9)
PRI_CLASS_SHORT	(pg312.8)
PRI_CLASS_STRING	(pg312.10)
PRI_CLASS_UNKNOWN	(pg312.7)
PRI_DEFAULT_EJECT_TRAY	(pg301.47)
PRI_DEFAULT_FM	(pg301.10)
PRI_DEFAULT_PRIORITY	(pg298.33)
PRI_ET_BIN1	(pg302.6)
PRI_ET_BIN10	(pg302.15)
PRI_ET_BIN2	(pg302.7)
PRI_ET_BIN3	(pg302.8)
PRI_ET_BIN4	(pg302.9)
PRI_ET_BIN5	(pg302.10)
PRI_ET_BIN6	(pg302.11)
PRI_ET_BIN7	(pg302.12)
PRI_ET_BIN8	(pg302.13)
PRI_ET_BIN9	(pg302.14)
PRI_ET_DEFAULT	(pg301.48)
PRI_ET_LOWER	(pg302.3)
PRI_ET_LOWER_OFFSET	(pg302.4)

+++ appendix +++

PRI_ET_SIDE	(pg302.5)
PRI_ET_UPPER	(pg302.1)
PRI_ET_UPPER_OFFSET	(pg302.2)
PRI_FM_300_DPI	(pg301.4)
PRI_FM_400_DPI	(pg301.5)
PRI_FM_COARSE	(pg301.1)
PRI_FM_FINE	(pg301.3)
PRI_FM_FNDATA	(pg301.6)
PRI_FM_OTHERDATA	(pg301.7)
PRI_HOLD_PRIORITY	(pg298.31)
PRI_MAX_DOCS_PER_REQ	(pg302.35)
PRI_MAX_EJECT_TRAY	(pg302.16)
PRI_MAX_FM	(pg301.9)
PRI_MAX_FORM_NAME_LEN	(pg302.31)
PRI_MAX_HEADLINE_LEN	(pg302.34)
PRI_MAX_MEMO_LEN	(pg302.37)
PRI_MAX_NOTE_LEN	(pg302.32)
PRI_MAX_OPTION	(pg303.33)
PRI_MAX_OP_STATUS	(pg297.9)
PRI_MAX_ORIENT	(pg300.28)
PRI_MAX_OVERLAY	(pg301.36)
PRI_MAX_PAPER_SIZE	(pg298.5)
PRI_MAX_PD	(pg300.40)
PRI_MAX_PHONE_EXT_LEN	(pg302.36)
PRI_MAX_PHONE_NUM_LEN	(pg302.33)
PRI_MAX_PRIORITY	(pg298.34)
PRI_MAX_PT	(pg301.23)
PRI_MAX_REQ_STATUS	(pg296.41)
PRI_MAX_SCALE	(pg300.2)
PRI_MAX_SERVICE_TYPE	(pg306.36)
PRI_MAX_SUBSTATUS	(pg310.5)
PRI_MIN_PRIORITY	(pg298.32)
PRI_MULTI_OVERLAY	(pg301.34)
PRI_NO_OVERLAY	(pg301.32)
PRI_NO_SERVICE_ID	(pg302.25)
PRI_ONE_OVERLAY	(pg301.33)
PRI_OPTION_ANNOTATIONS	(pg303.16)
PRI_OPTION_COLLATE	(pg303.8)
PRI_OPTION_COPIES	(pg303.10)
PRI_OPTION_COVER_DOC	(pg303.30)
PRI_OPTION_COVER_SSN	(pg303.31)
PRI_OPTION_DOC_HEADERS	(pg303.18)
PRI_OPTION_EJECT_TRAY	(pg303.27)
PRI_OPTION_FAX_MODE	(pg303.24)
PRI_OPTION_FORM_NAME	(pg303.14)
PRI_OPTION_HEADLINE	(pg303.23)
PRI_OPTION_MEMO	(pg303.29)
PRI_OPTION_NOTE	(pg303.15)

+++ appendix +++

PRI_OPTION_ORIENTATION	(pg303.20)
PRI_OPTION_OVERLAY	(pg303.21)
PRI_OPTION_PAGE_FOOTNOTE	(pg303.25)
PRI_OPTION_PAPER_SIZE	(pg303.5)
PRI_OPTION_PHONE_EXT	(pg303.28)
PRI_OPTION_PHONE_NUM	(pg303.22)
PRI_OPTION_PRINTER_NAME	(pg303.7)
PRI_OPTION_PRINT_TIME	(pg303.11)
PRI_OPTION_PRIORITY	(pg303.6)
PRI_OPTION_REQUEST_HEADER	(pg303.17)
PRI_OPTION_SCALING	(pg303.19)
PRI_OPTION_SECURITY	(pg303.9)
PRI_OPTION_STAPLE	(pg303.12)
PRI_OPTION_TIME_FOOTNOTE	(pg303.26)
PRI_OPTION_TWO_SIDED	(pg303.13)
PRI_ORIENT_DEFAULT	(pg300.12)
PRI_ORIENT_LANDSCAPE	(pg300.14)
PRI_ORIENT_NO_ROTATE	(pg300.24)
PRI_ORIENT_PORTRAIT	(pg300.20)
PRI_PD_BACKWARD	(pg300.38)
PRI_PD_FORWARD	(pg300.37)
PRI_POS_AVAILABLE	(pg297.5)
PRI_POS_DISABLED	(pg297.4)
PRI_POS_NEEDS_ATTN	(pg297.6)
PRI_POS_NEEDS_SVC	(pg297.7)
PRI_POS_UNKNOWN	(pg297.3)
PRI_PS_10x14	(pg298.2)
PRI_PS_18x24	(pg297.43)
PRI_PS_A0	(pg297.35)
PRI_PS_A1	(pg297.36)
PRI_PS_A2	(pg297.37)
PRI_PS_A3	(pg297.38)
PRI_PS_A4	(pg297.39)
PRI_PS_A5	(pg297.40)
PRI_PS_B	(pg297.31)
PRI_PS_B4	(pg297.41)
PRI_PS_B5	(pg297.42)
PRI_PS_BEST_AVAIL	(pg298.1)
PRI_PS_BOTTOM	(pg297.45)
PRI_PS_C	(pg297.32)
PRI_PS_D	(pg297.33)
PRI_PS_DONT_CARE	(pg297.47)
PRI_PS_E	(pg297.34)
PRI_PS_EXECUTIVE	(pg298.4)
PRI_PS_GLOBAL_DEFAULT	(pg298.3)
PRI_PS_HALF_LETTER	(pg297.48)
PRI_PS_LEGAL	(pg297.30)
PRI_PS_LETTER	(pg297.29)

+++ appendix +++

PRI_PS_THIRD	(pg297.46)
PRI_PS_TOP	(pg297.44)
PRI_PS_UNKNOWN	(pg297.28)
PRI_PT_FAX	(pg301.21)
PRI_PT_PRINTER	(pg301.20)
PRI_PT_UNKNOWN	(pg301.19)
PRI_RS_CANCELLED	(pg296.36)
PRI_RS_COMPLETE	(pg296.34)
PRI_RS_FAILED	(pg296.35)
PRI_RS_FETCHING	(pg296.38)
PRI_RS_PRINTING	(pg296.33)
PRI_RS_QUEUED	(pg296.32)
PRI_RS_SUSPENDED	(pg296.39)
PRI_RS_UNKNOWN	(pg296.30)
PRI_RS_WAITING	(pg296.37)
PRI_SCALE_APPROX	(pg299.20)
PRI_SCALE_CENTER	(pg299.41)
PRI_SCALE_CLIPBOTH	(pg299.7)
PRI_SCALE_ENHANCED_EXACT	(pg299.45)
PRI_SCALE_EXACT	(pg299.13)
PRI_SCALE_NORMAL	(pg298.43)
PRI_SCALE_ORIGINAL	(pg299.28)
PRI_ST_CACHE_SERVICE	(pg306.34)
PRI_ST_DOC_SERVICE	(pg306.33)
PRI_SUBSTATUS_CONVERSION	(pg309.48)
PRI_SUBSTATUS_PRINTING	(pg310.4)
PRI_SUBSTATUS_RETRY	(pg310.3)
PRI_SUBSTATUS_SPOOLING	(pg310.1)
PRI_SUBSTATUS_TRANSMIT	(pg310.2)
PRI_SUBSTATUS_UNINITIALIZED	(pg309.46)
PRI_SUBSTATUS_WAITING_CONV	(pg309.47)
PRI_TYPE_UNKNOWN	(pg311.43)
PRI_WorkFlo_Printer_or_FAX_error	(pg319.28)
PRI_XIN_PROTOCOL	(pg312.42)
PRI_XNOT_IN_PROTOCOL	(pg312.43)
PRI_class_typ	(pg312.6)
PRI_defs	(pg296.14)
PRI_dev_err	(pg313.4)
PRI_doc_spec_id_typ	(pg312.30)
PRI_doc_specifier_struct	(pg306.38)
PRI_doc_specifier_typ	(pg307.14)
PRI_eject_tray_typ	(pg301.45)
PRI_err_address_to_string	(pg319.37)
PRI_err_bad_cache	(pg318.48)
PRI_err_bad_doc	(pg319.5)
PRI_err_bad_file	(pg316.38)
PRI_err_bad_ims	(pg319.10)
PRI_err_bad_migrate	(pg316.30)

+++ appendix +++

PRI_err_bad_version	(pg316.6)
PRI_err_bulk_trans_unexpected	(pg315.48)
PRI_err_cache_timeout	(pg319.30)
PRI_err_del_ageable	(pg317.9)
PRI_err_dest_redirected	(pg316.41)
PRI_err_duplicate_doc	(pg317.38)
PRI_err_duplicate_option	(pg317.35)
PRI_err_duplicate_request	(pg317.32)
PRI_err_duplicate_service	(pg317.47)
PRI_err_empty_object	(pg316.27)
PRI_err_extra_papersize	(pg318.30)
PRI_err_internal_rpc_error	(pg316.9)
PRI_err_invalid_NCH_name	(pg318.2)
PRI_err_invalid_eject_tray	(pg319.16)
PRI_err_invalid_fax_collate	(pg313.42)
PRI_err_invalid_fax_copies	(pg314.3)
PRI_err_invalid_fax_doc_headers	(pg315.18)
PRI_err_invalid_fax_eject_tray	(pg315.6)
PRI_err_invalid_fax_form	(pg313.45)
PRI_err_invalid_fax_headline	(pg314.39)
PRI_err_invalid_fax_memo	(pg315.32)
PRI_err_invalid_fax_mode	(pg314.45)
PRI_err_invalid_fax_note	(pg315.21)
PRI_err_invalid_fax_orientation	(pg314.18)
PRI_err_invalid_fax_overlay	(pg314.24)
PRI_err_invalid_fax_phone	(pg314.33)
PRI_err_invalid_fax_phone_ext	(pg315.38)
PRI_err_invalid_fax_scaling	(pg314.12)
PRI_err_invalid_fax_staple	(pg315.12)
PRI_err_invalid_fax_two_sided	(pg315.15)
PRI_err_invalid_note	(pg314.9)
PRI_err_invalid_num_pages	(pg317.16)
PRI_err_invalid_option	(pg313.15)
PRI_err_invalid_page_ordering	(pg315.24)
PRI_err_invalid_page_range	(pg313.27)
PRI_err_invalid_paper_size	(pg317.2)
PRI_err_invalid_parameter	(pg313.30)
PRI_err_invalid_pointer	(pg316.21)
PRI_err_invalid_pri_copies	(pg314.6)
PRI_err_invalid_pri_cover_doc	(pg315.41)
PRI_err_invalid_pri_footnote	(pg314.48)
PRI_err_invalid_pri_form	(pg313.48)
PRI_err_invalid_pri_headline	(pg314.36)
PRI_err_invalid_pri_memo	(pg315.29)
PRI_err_invalid_pri_mode	(pg314.42)
PRI_err_invalid_pri_orientation	(pg314.21)
PRI_err_invalid_pri_overlay	(pg314.27)
PRI_err_invalid_pri_phone	(pg314.30)

+++ appendix +++

PRI_err_invalid_pri_phone_ext	(pg315.35)
PRI_err_invalid_pri_scaling	(pg314.15)
PRI_err_invalid_printer_num	(pg319.13)
PRI_err_invalid_priority	(pg313.39)
PRI_err_invalid_security_setting	(pg315.44)
PRI_err_invalid_service_type	(pg318.22)
PRI_err_invalid_session_handle	(pg313.8)
PRI_err_invalid_staple	(pg313.33)
PRI_err_invalid_status	(pg315.3)
PRI_err_invalid_two_sided	(pg313.36)
PRI_err_lock_busy	(pg316.35)
PRI_err_migrate_timeout	(pg317.21)
PRI_err_multiple_notify	(pg317.44)
PRI_err_must_delete_pri_cache	(pg318.11)
PRI_err_no_notify_waiting	(pg316.12)
PRI_err_no_permission	(pg313.12)
PRI_err_no_printer_by_that_name	(pg316.47)
PRI_err_no_printers_for_options	(pg317.13)
PRI_err_no_such_doc	(pg317.41)
PRI_err_no_such_option	(pg318.5)
PRI_err_no_such_printer	(pg316.18)
PRI_err_no_such_request	(pg313.18)
PRI_err_no_such_service	(pg313.24)
PRI_err_not_in_pri_cache	(pg317.24)
PRI_err_notify_strlen	(pg319.34)
PRI_err_notify_timeout	(pg316.24)
PRI_err_old_request	(pg318.25)
PRI_err_other_error	(pg313.21)
PRI_err_print_data_too_short	(pg315.9)
PRI_err_request_cancelled	(pg316.15)
PRI_err_string_to_address	(pg319.40)
PRI_err_too_many_docs	(pg318.16)
PRI_err_too_many_options	(pg318.19)
PRI_err_too_many_pages	(pg317.5)
PRI_err_too_many_papersizes	(pg318.39)
PRI_err_unexpected_exit	(pg319.20)
PRI_fax_mode_typ	(pg300.47)
PRI_option_typ	(pg303.3)
PRI_orientation_typ	(pg300.10)
PRI_overlay_typ	(pg301.30)
PRI_page_rec_typ	(pg312.36)
PRI_paper_size_typ	(pg297.26)
PRI_position_typ	(pg311.31)
PRI_print_direction_typ	(pg300.35)
PRI_print_option_struct	(pg303.45)
PRI_print_option_typ	(pg305.43)
PRI_printer_attr	(pg307.40)
PRI_printer_attr2_typ	(pg308.43)

+++ appendix +++

PRI_printer_attr_typ	(pg308.8)
PRI_printer_op_status_typ	(pg297.1)
PRI_printer_status	(pg306.4)
PRI_printer_status_typ	(pg306.22)
PRI_printer_type_typ	(pg301.17)
PRI_priority_typ	(pg298.29)
PRI_pro_err	(pg313.3)
PRI_request_desc2_typ	(pg311.22)
PRI_request_desc_struct	(pg309.4)
PRI_request_desc_typ	(pg309.33)
PRI_request_filter_typ	(pg312.26)
PRI_request_status_typ	(pg296.28)
PRI_scaling_typ	(pg298.41)
PRI_service_id_typ	(pg302.23)
PRI_service_status_struct	(pg307.22)
PRI_service_status_typ	(pg307.33)
PRI_service_type_typ	(pg306.31)
PRI_sub_status_typ	(pg309.44)
PRI_trace_log_tuple	(pg319.25)
PRI_type_typ	(pg311.42)
PRI_zother_error	(pg312.45)
PRS_err	(pg330.14)
PRS_err_2manykws	(pg331.29)
PRS_err_MAXRESWD	(pg336.9)
PRS_err_badwhere	(pg332.9)
PRS_err_bugsimp	(pg330.37)
PRS_err_caryoflo	(pg334.1)
PRS_err_coltype	(pg335.45)
PRS_err_constsize	(pg336.41)
PRS_err_dclnamemis	(pg336.45)
PRS_err_deallocon	(pg334.13)
PRS_err_doctypebug1	(pg337.33)
PRS_err_doctypekludge	(pg338.1)
PRS_err_doctypemis	(pg337.25)
PRS_err_doctypstglen	(pg337.37)
PRS_err_endarea	(pg336.25)
PRS_err_expropmis	(pg332.5)
PRS_err_exprtypemis	(pg332.1)
PRS_err_exptype	(pg336.1)
PRS_err_findexpected	(pg332.25)
PRS_err_fpfail	(pg334.37)
PRS_err_idenexpected	(pg332.45)
PRS_err_idstate	(pg336.33)
PRS_err_ilgldateop	(pg334.17)
PRS_err_ilgldeclnameop	(pg337.1)
PRS_err_ilgldef	(pg337.5)
PRS_err_ilgldoctypeop	(pg337.29)
PRS_err_ilgldoctypepstgval	(pg337.41)

+++ appendix +++

PRS_err_ilglfpconst	(pg333.29)
PRS_err_ilglhexconst	(pg333.33)
PRS_err_ilglkconlist	(pg333.17)
PRS_err_ilglkwlist	(pg331.37)
PRS_err_ilgltype	(pg337.21)
PRS_err_ilglmenuop	(pg335.5)
PRS_err_ilglqparam	(pg333.25)
PRS_err_ilglkeycon	(pg332.17)
PRS_err_ilgltimeop	(pg334.21)
PRS_err_ilglwhere	(pg331.45)
PRS_err_iltlkconslst	(pg332.21)
PRS_err_kcon	(pg336.17)
PRS_err_kconop	(pg336.29)
PRS_err_kcons	(pg332.13)
PRS_err_keycflo	(pg335.9)
PRS_err_keycondexp	(pg333.5)
PRS_err_keyconstsz	(pg335.25)
PRS_err_keyconstype	(pg335.33)
PRS_err_keyeofexp	(pg336.37)
PRS_err_keyidexpected	(pg332.41)
PRS_err_keyopexp	(pg333.1)
PRS_err_keywlcflo	(pg335.13)
PRS_err_kwdsexp	(pg331.21)
PRS_err_likeexp	(pg337.13)
PRS_err_lparenexp	(pg331.25)
PRS_err_menulen	(pg335.29)
PRS_err_menumis	(pg335.1)
PRS_err_menumnotfound	(pg337.9)
PRS_err_misrtquote	(pg334.9)
PRS_err_mustbeint	(pg331.13)
PRS_err_mustbeposint	(pg331.17)
PRS_err_nconspoflo	(pg330.25)
PRS_err_neqstgl	(pg337.45)
PRS_err_ni_arithop	(pg333.45)
PRS_err_ni_kwlst	(pg336.13)
PRS_err_ni_logop	(pg335.37)
PRS_err_ni_mpkeys	(pg335.17)
PRS_err_ni_qparam	(pg333.37)
PRS_err_ni_unop	(pg334.41)
PRS_err_nkeyitems	(pg336.21)
PRS_err_nomem	(pg330.17)
PRS_err_noncondatestg	(pg334.25)
PRS_err_nonconnumstg	(pg334.33)
PRS_err_noncontimstg	(pg334.29)
PRS_err_notdbcol	(pg334.45)
PRS_err_nottableid	(pg332.29)
PRS_err_oparyoflo	(pg333.41)
PRS_err_ormoofexp	(pg331.41)

+++ appendix +++

PRS_err_poolx	(pg335.41)
PRS_err_pqlen	(pg335.21)
PRS_err_rang1	(pg330.41)
PRS_err_rang1_2	(pg330.45)
PRS_err_rang2	(pg333.13)
PRS_err_rang3	(pg331.1)
PRS_err_rang3_2	(pg331.5)
PRS_err_rang4	(pg331.9)
PRS_err_rangeexp	(pg333.9)
PRS_err_stackoflo	(pg333.21)
PRS_err_stgconsexp	(pg331.33)
PRS_err_stgpoflo	(pg330.29)
PRS_err_syntaboflo	(pg330.33)
PRS_err_tableidexp	(pg332.33)
PRS_err_tokenize	(pg330.21)
PRS_err_unkconstyp	(pg334.5)
PRS_err_unkop	(pg337.17)
PRS_err_viaexpected	(pg332.37)
PRS_err_what	(pg336.5)
PRS_z2manykwds	(pg331.28)
PRS_zMAXRESWD	(pg336.8)
PRS_zbadwhere	(pg332.8)
PRS_zbugsimp	(pg330.36)
PRS_zcaryoflo	(pg333.48)
PRS_zcoltype	(pg335.44)
PRS_zconstsize	(pg336.40)
PRS_zdclnamemis	(pg336.44)
PRS_zdealloccon	(pg334.12)
PRS_zdoctypebug1	(pg337.32)
PRS_zdoctypekludge	(pg337.48)
PRS_zdoctypemis	(pg337.24)
PRS_zdoctypstglen	(pg337.36)
PRS_zendarea	(pg336.24)
PRS_zexpropmis	(pg332.4)
PRS_zexprtypemis	(pg331.48)
PRS_zexptype	(pg335.48)
PRS_zfindexpected	(pg332.24)
PRS_zfpfail	(pg334.36)
PRS_zidenexpected	(pg332.44)
PRS_zidstate	(pg336.32)
PRS_zilgldateop	(pg334.16)
PRS_zilglldclnameop	(pg336.48)
PRS_zilgldef	(pg337.4)
PRS_zilgldoctypeop	(pg337.28)
PRS_zilgldoctypepstgval	(pg337.40)
PRS_zilglfpconst	(pg333.28)
PRS_zilglhexconst	(pg333.32)
PRS_zilglkconlist	(pg333.16)

+++ appendix +++

PRS_zilglkwlist	(pg331.36)
PRS_zilgltype	(pg337.20)
PRS_zilglmenuop	(pg335.4)
PRS_zilglqparam	(pg333.24)
PRS_zilglskeycon	(pg332.16)
PRS_zilgltimeop	(pg334.20)
PRS_zilglwhere	(pg331.44)
PRS_ziltlkconslst	(pg332.20)
PRS_zkcon	(pg336.16)
PRS_zkconop	(pg336.28)
PRS_zkcons	(pg332.12)
PRS_zkeycoflo	(pg335.8)
PRS_zkeycondexp	(pg333.4)
PRS_zkeyconstsz	(pg335.24)
PRS_zkeyconstype	(pg335.32)
PRS_zkeyeofexp	(pg336.36)
PRS_zkeyidexpected	(pg332.40)
PRS_zkeyopexp	(pg332.48)
PRS_zkeywlcflo	(pg335.12)
PRS_zkwdsexp	(pg331.20)
PRS_zlikeexp	(pg337.12)
PRS_zlparenexp	(pg331.24)
PRS_zmenulen	(pg335.28)
PRS_zmenumis	(pg334.48)
PRS_zmenunmnotfound	(pg337.8)
PRS_zmisrtquote	(pg334.8)
PRS_zmustbeint	(pg331.12)
PRS_zmustbeposint	(pg331.16)
PRS_znconspoflo	(pg330.24)
PRS_zneqstg1	(pg337.44)
PRS_zni_arithop	(pg333.44)
PRS_zni_kwlst	(pg336.12)
PRS_zni_logop	(pg335.36)
PRS_zni_mpkeys	(pg335.16)
PRS_zni_qparam	(pg333.36)
PRS_zni_unop	(pg334.40)
PRS_znkeyitems	(pg336.20)
PRS_znomem	(pg330.16)
PRS_znoncondatestg	(pg334.24)
PRS_znonconnumstg	(pg334.32)
PRS_znoncontimstg	(pg334.28)
PRS_znotdbcol	(pg334.44)
PRS_znottableid	(pg332.28)
PRS_zoparyoflo	(pg333.40)
PRS_zormoofexp	(pg331.40)
PRS_zpoolx	(pg335.40)
PRS_zpqlen	(pg335.20)
PRS_zrang1	(pg330.40)

+++ appendix +++

PRS_zrang1_2	(pg330.44)
PRS_zrang2	(pg333.12)
PRS_zrang3	(pg330.48)
PRS_zrang3_2	(pg331.4)
PRS_zrang4	(pg331.8)
PRS_zrangeexp	(pg333.8)
PRS_zstackoflo	(pg333.20)
PRS_zstgconsexp	(pg331.32)
PRS_zstgpoflo	(pg330.28)
PRS_zsymtaboflo	(pg330.32)
PRS_ztableidexp	(pg332.32)
PRS_ztokensize	(pg330.20)
PRS_zunkconstyp	(pg334.4)
PRS_zunkop	(pg337.16)
PRS_zviaexpected	(pg332.36)
PRS_zwhat	(pg336.4)
RETRY_OPEN_COUNT	(pg389.8)
SEC_ACTIVATED	(pg371.18)
SEC_ADDR_LENGTH	(pg370.48)
SEC_ANYONE_ID	(pg373.41)
SEC_ANYONE_NAME	(pg375.13)
SEC_ARABIC_ID	(pg372.21)
SEC_ARABIC_NAME	(pg372.48)
SEC_AUDIT_GRP_ID	(pg374.47)
SEC_AUDIT_GRP_NAME	(pg375.28)
SEC_BAD_CHARS	(pg375.5)
SEC_BAD_TERM_CHARS	(pg375.6)
SEC_BRITISH_ID	(pg372.13)
SEC_BRITISH_NAME	(pg372.40)
SEC_CN_LENGTH	(pg371.13)
SEC_COMMENTS_LENGTH	(pg370.38)
SEC_CONVERTED	(pg373.4)
SEC_DAEMON_IDLE_TIME	(pg389.12)
SEC_DANISH_ID	(pg372.15)
SEC_DANISH_NAME	(pg372.42)
SEC_DB2_ACCT_NAME_LENGTH	(pg371.9)
SEC_DB2_ACCT_NAME_MAX_LENGTH	(pg371.48)
SEC_DB2_CLEAR_SCND_PWD	(pg371.44)
SEC_DB2_MAGIC_NUM	(pg371.46)
SEC_DB2_MOVE_SCND_PWD	(pg371.43)
SEC_DB2_PRIMARY_PWD	(pg371.40)
SEC_DB2_PWD_MAX_LENGTH	(pg371.47)
SEC_DB2_SECONDARY_PWD	(pg371.41)
SEC_DBMAP_FOPEN_ID	(pg382.36)
SEC_DBMAP_FOPEN_NAME	(pg382.40)
SEC_DBMAP_FOPEN_PWD	(pg382.41)
SEC_DBMAP_LAST_RESERVED_ID	(pg382.37)
SEC_DBMAP_NOMAP_ID	(pg382.35)

+++ appendix +++

SEC_DBMAP_NOMAP_NAME	(pg382.39)
SEC_DBNAME_LENGTH	(pg371.3)
SEC_DBPWD_LENGTH	(pg371.5)
SEC_DEVCLASS_FAX	(pg376.6)
SEC_DEVCLASS_PRINTER	(pg376.3)
SEC_DEVCLASS_SCANNER	(pg376.4)
SEC_DEVCLASS_TAPE	(pg376.5)
SEC_DEVCLASS_TERMINAL	(pg376.2)
SEC_DEVCLASS_UNKNOWN	(pg376.1)
SEC_DEVICE_DEFAULTS_ID	(pg374.26)
SEC_DEVICE_DEFAULTS_NAME	(pg375.22)
SEC_DICT_NAME_LENGTH	(pg370.40)
SEC_DN_LENGTH	(pg371.11)
SEC_DOM_LENGTH	(pg370.42)
SEC_DUTCH_ID	(pg372.20)
SEC_DUTCH_NAME	(pg372.47)
SEC_EDB2PWD_LENGTH	(pg371.10)
SEC_EDBPWD_LENGTH	(pg371.4)
SEC_ENGLISH_ID	(pg372.8)
SEC_ENGLISH_NAME	(pg372.35)
SEC_EPWD_LENGTH	(pg370.36)
SEC_EQL	(pg376.17)
SEC_EXPIRED	(pg377.35)
SEC_FIELD_SVC_GRP_ID	(pg374.5)
SEC_FIELD_SVC_GRP_NAME	(pg375.15)
SEC_FIELD_SVC_USR_ID	(pg374.41)
SEC_FIELD_SVC_USR_NAME	(pg375.25)
SEC_FRENCH_ID	(pg372.10)
SEC_FRENCH_NAME	(pg372.37)
SEC_FULL_OBJECT_LENGTH	(pg370.45)
SEC_FUNCLIST_FUNCTION_ADMIN	(pg386.2)
SEC_FUNCLIST_GROUP_ADMIN	(pg385.48)
SEC_FUNCLIST_PARAM_ADMIN	(pg386.5)
SEC_FUNCLIST_PASSWORD_ADMIN	(pg386.6)
SEC_FUNCLIST_SECURITY_ADMIN	(pg386.3)
SEC_FUNCLIST_SERVICE_ADMIN	(pg386.7)
SEC_FUNCLIST_SITE_ADMIN	(pg386.1)
SEC_FUNCLIST_TIME_ADMIN	(pg386.4)
SEC_FUN_LENGTH	(pg370.44)
SEC_GEQ	(pg376.18)
SEC_GERMAN_ID	(pg372.9)
SEC_GERMAN_NAME	(pg372.36)
SEC_GET	(pg388.36)
SEC_GROUP_DEFAULTS_ID	(pg374.25)
SEC_GROUP_DEFAULTS_NAME	(pg375.21)
SEC_GROUP_OBJECT	(pg371.19)
SEC_GTR	(pg376.19)
SEC_GUID_REGISTRY_FORMAT_LENGTH	(pg386.25)

+++ appendix +++

SEC_INFOTYPE_DATE_TIME	(pg386.20)
SEC_INFOTYPE_DEFAULTS	(pg386.19)
SEC_INFOTYPE_FUNCTION	(pg386.17)
SEC_INFOTYPE_FUNCTION_MEMBER	(pg386.18)
SEC_INFOTYPE_GROUP	(pg386.23)
SEC_INFOTYPE_SERVICE_PROCESS	(pg386.21)
SEC_INFOTYPE_TERMINAL	(pg386.15)
SEC_INFOTYPE_TERMINAL_MEMBER	(pg386.16)
SEC_INFOTYPE_UPDATE_TIME	(pg386.22)
SEC_INFOTYPE_USER	(pg386.13)
SEC_INFOTYPE_USER_MEMBER	(pg386.14)
SEC_INVALID_ID	(pg385.40)
SEC_ITALIAN_ID	(pg372.11)
SEC_ITALIAN_NAME	(pg372.38)
SEC_LANGUAGE_LENGTH	(pg371.1)
SEC_LAST_RESERVED_ID	(pg374.48)
SEC_MAXDB_OPTIONS	(pg382.47)
SEC_MAXOBJ_OPTIONS	(pg377.25)
SEC_MAXSYS_OPTIONS	(pg381.15)
SEC_MAX_DEVICE_CLASS	(pg376.9)
SEC_MAX_DMASK_LENGTH	(pg380.31)
SEC_MAX_DOCS_TO_DECLARE	(pg387.34)
SEC_MAX_DWEEK	(pg380.10)
SEC_MAX_FORM_LENGTH	(pg380.29)
SEC_MAX_FUNCLASS	(pg380.26)
SEC_MAX_GROUP_ID	(pg385.39)
SEC_MAX_HOURS	(pg380.8)
SEC_MAX_LANGUAGES	(pg373.2)
SEC_MAX_LANGUAGE_LENGTH	(pg380.34)
SEC_MAX_MAX_SESSIONS	(pg380.14)
SEC_MAX_MINUTES	(pg380.6)
SEC_MAX_MMASK_LENGTH	(pg380.33)
SEC_MAX_NMASK_LENGTH	(pg380.32)
SEC_MAX_OBJ_CLASS	(pg375.42)
SEC_MAX_OLD_FUNCTIONS	(pg385.37)
SEC_MAX_PWD_ATTEMPTS	(pg380.18)
SEC_MAX_PWD_FAILURE_MINS	(pg380.24)
SEC_MAX_PWD_GRACE_PERIOD	(pg380.22)
SEC_MAX_PWD_LENGTH	(pg380.27)
SEC_MAX_PWD_MIN_LENGTH	(pg380.16)
SEC_MAX_PWD_RENEWAL_DAYS	(pg380.20)
SEC_MAX_SYS_OVERRIDE	(pg380.12)
SEC_MAX_TITLE_LENGTH	(pg380.28)
SEC_MAX_TMASK_LENGTH	(pg380.30)
SEC_MEMLIST_SUBSCRIBER	(pg386.10)
SEC_MEMLIST_SUBSCRIPTION	(pg386.9)
SEC_MIN_DEVICE_CLASS	(pg376.8)
SEC_MIN_DWEEK	(pg380.9)

+++ appendix +++

SEC_MIN_FUNCLASS	(pg380.25)
SEC_MIN_HOURS	(pg380.7)
SEC_MIN_MAX_SESSIONS	(pg380.13)
SEC_MIN_MINUTES	(pg380.5)
SEC_MIN_OBJ_CLASS	(pg375.41)
SEC_MIN_PWD_ATTEMPTS	(pg380.17)
SEC_MIN_PWD_FAILURE_MINS	(pg380.23)
SEC_MIN_PWD_GRACE_PERIOD	(pg380.21)
SEC_MIN_PWD_MIN_LENGTH	(pg380.15)
SEC_MIN_PWD_RENEWAL_DAYS	(pg380.19)
SEC_MIN_SYS_OVERRIDE	(pg380.11)
SEC_NORWEGIAN_ID	(pg372.16)
SEC_NORWEGIAN_NAME	(pg372.43)
SEC_NO_EXPIRATION	(pg377.34)
SEC_NULL_GROUP_ID	(pg373.40)
SEC_NULL_GROUP_NAME	(pg375.12)
SEC_OBJCLASS_DEVICE	(pg375.38)
SEC_OBJCLASS_GROUP	(pg375.36)
SEC_OBJCLASS_SYSTEM	(pg375.35)
SEC_OBJCLASS_USER	(pg375.37)
SEC_OBJ_LENGTH	(pg370.43)
SEC_OPERATOR_GRP_ID	(pg374.6)
SEC_OPERATOR_GRP_NAME	(pg375.16)
SEC_OPERATOR_USR_ID	(pg374.42)
SEC_OPERATOR_USR_NAME	(pg375.26)
SEC_OPTDB_NAME	(pg382.45)
SEC_OPTDB_PASSWORD	(pg382.46)
SEC_OPTOBJ_ADMIN_CLASS_GROUP	(pg376.46)
SEC_OPTOBJ_ADMIN_CLASS_NOT	(pg376.48)
SEC_OPTOBJ_ADMIN_CLASS_PASSWORD	(pg376.47)
SEC_OPTOBJ_ADMIN_CLASS_PRINC	(pg376.45)
SEC_OPTOBJ_ADMIN_CLASS_SUPERV	(pg376.44)
SEC_OPTOBJ_ADMIN_GROUP	(pg377.4)
SEC_OPTOBJ_CLASS_KEY	(pg377.24)
SEC_OPTOBJ_COMMENTS	(pg377.21)
SEC_OPTOBJ_DEFAULT_DBMAP_ID	(pg377.23)
SEC_OPTOBJ_DEVICE_CLASS	(pg376.43)
SEC_OPTOBJ_DEVICE_SECURITY	(pg377.1)
SEC_OPTOBJ_END_DWEEK	(pg377.11)
SEC_OPTOBJ_END_HOUR	(pg377.10)
SEC_OPTOBJ_END_MIN	(pg377.9)
SEC_OPTOBJ_EXPIRATION_TIME	(pg377.12)
SEC_OPTOBJ_FAILED_ERR	(pg377.15)
SEC_OPTOBJ_FAILED_WHERE	(pg377.14)
SEC_OPTOBJ_LANGUAGE	(pg377.2)
SEC_OPTOBJ_LOG_DB_UPDATES	(pg377.18)
SEC_OPTOBJ_LOG_FAILED_LOGON	(pg377.17)
SEC_OPTOBJ_LOG_SUCCESS_LOGON	(pg377.16)

+++ appendix +++

SEC_OPTOBJ_MAX_SESSIONS	(pg377.20)
SEC_OPTOBJ_OBJECT_CLASS	(pg376.42)
SEC_OPTOBJ_OBJECT_NAME	(pg376.41)
SEC_OPTOBJ_PASSWORD	(pg377.22)
SEC_OPTOBJ_PRIM_GROUP	(pg377.3)
SEC_OPTOBJ_SESS_GROUP	(pg377.5)
SEC_OPTOBJ_START_DWEEK	(pg377.8)
SEC_OPTOBJ_START_HOUR	(pg377.7)
SEC_OPTOBJ_START_MIN	(pg377.6)
SEC_OPTOBJ_SUCCESS_WHERE	(pg377.13)
SEC_OPTOBJ_SYSTEM_OVERRIDE	(pg377.19)
SEC_OPTSYS_ALLOW_OVERRIDE	(pg381.14)
SEC_OPTSYS_DEVICE_SECURITY	(pg380.42)
SEC_OPTSYS_END_DWEEK	(pg381.2)
SEC_OPTSYS_END_HOUR	(pg381.1)
SEC_OPTSYS_END_MIN	(pg380.48)
SEC_OPTSYS_LANGUAGE	(pg381.13)
SEC_OPTSYS_LOG_DB_UPDATES	(pg381.12)
SEC_OPTSYS_LOG_FAILED_LOGON	(pg381.11)
SEC_OPTSYS_LOG_SUCCESS_LOGON	(pg381.10)
SEC_OPTSYS_MAX_SESSIONS	(pg381.3)
SEC_OPTSYS_NO_FUNC_DEF_OK	(pg380.43)
SEC_OPTSYS_PWD_ATTEMPTS	(pg381.8)
SEC_OPTSYS_PWD_FAILURE_MINS	(pg381.9)
SEC_OPTSYS_PWD_GRACE_PERIOD	(pg381.7)
SEC_OPTSYS_PWD_MIN_LEN	(pg381.5)
SEC_OPTSYS_PWD_RENEWAL_DAYS	(pg381.6)
SEC_OPTSYS_PWD_SPEC_CHAR	(pg381.4)
SEC_OPTSYS_START_DWEEK	(pg380.47)
SEC_OPTSYS_START_HOUR	(pg380.46)
SEC_OPTSYS_START_MIN	(pg380.45)
SEC_OPTSYS_SYSTEM_ID	(pg380.41)
SEC_OPTSYS_UPDATE_TIME	(pg380.44)
SEC_ORGDOM_LENGTH	(pg370.39)
SEC_ORG_LENGTH	(pg370.41)
SEC_PORTUGUESE_ID	(pg372.14)
SEC_PORTUGUESE_NAME	(pg372.41)
SEC_PWD_EXPIRED	(pg377.47)
SEC_PWD_LENGTH	(pg370.37)
SEC_PWD_RESET	(pg377.48)
SEC_RM_DECLARE_PERM_ANNOT_ONLY	(pg387.37)
SEC_RM_DECLARE_PERM_NO_CHANGE	(pg387.38)
SEC_RM_DECLARE_PERM_READ_ONLY	(pg387.36)
SEC_RM_DEFAULT_PURGE_LEVEL	(pg387.41)
SEC_RM_DEFAULT_RM_GROUP_ID	(pg387.39)
SEC_RM_DEFAULT_SCRUB_LEVEL	(pg387.42)
SEC_RM_DEFAULT_SYSADMIN_LOCK	(pg387.45)
SEC_RM_INVALID_RM_GROUP_ID	(pg387.40)

+++ appendix +++

SEC_RM_LOG_LEVEL_ERR_ONLY	(pg387.43)
SEC_RM_LOG_LEVEL_VERBOSE	(pg387.44)
SEC_RM_SYSTEM_LEVEL_CONFIG_KEY	(pg387.35)
SEC_SCTPASSWORD_LENGTH	(pg371.2)
SEC_SCT_PROTOCOL	(pg388.40)
SEC_SERVPROC_ID	(pg374.14)
SEC_SERVPROC_NAME	(pg375.17)
SEC_SERVPROC_TERM_ID	(pg374.43)
SEC_SERVPROC_TERM_NAME	(pg375.27)
SEC_SET	(pg388.35)
SEC_SLU_ACCOUNTING_APPROX	(pg386.31)
SEC_SLU_ACCOUNTING_STRICT	(pg386.30)
SEC_SLU_NO_POOLING	(pg386.29)
SEC_SLU_duplicate	(pg405.31)
SEC_SLU_hard_limit_reached	(pg405.34)
SEC_SLU_not_found	(pg405.40)
SEC_SPANISH_ID	(pg372.12)
SEC_SPANISH_NAME	(pg372.39)
SEC_SWEDISH_ID	(pg372.17)
SEC_SWEDISH_NAME	(pg372.44)
SEC_SWISSGERMAN_ID	(pg372.18)
SEC_SWISSGERMAN_NAME	(pg372.45)
SEC_SWISSROMANCE_ID	(pg372.19)
SEC_SWISSROMANCE_NAME	(pg372.46)
SEC_SYSADMIN_GRP_ID	(pg373.48)
SEC_SYSADMIN_GRP_NAME	(pg375.14)
SEC_SYSADMIN_USR_ID	(pg374.37)
SEC_SYSADMIN_USR_NAME	(pg375.24)
SEC_SYSTEM_DEFAULTS_ID	(pg374.31)
SEC_SYSTEM_DEFAULTS_NAME	(pg375.23)
SEC_TERM_LENGTH	(pg370.46)
SEC_UI_only_supported_on_WAL	(pg406.1)
SEC_UNDEFINED_ID	(pg374.18)
SEC_UNDEFINED_NAME	(pg375.20)
SEC_UNIFIED_LOGON_BAD_CHARS	(pg375.7)
SEC_UNIXPWD_LENGTH	(pg370.47)
SEC_UNKNOWN_FUNC	(pg385.47)
SEC_UPDATE_SVCPRCS_NAME	(pg385.27)
SEC_UPDATE_SVCPRCS_PASSWORD	(pg385.28)
SEC_UPDATE_USER_FORLOGON	(pg385.26)
SEC_UPDATE_USER_LANGUAGE	(pg385.24)
SEC_UPDATE_USER_NAME	(pg385.29)
SEC_UPDATE_USER_PASSWORD	(pg385.23)
SEC_UPDATE_USER_PRIMARY	(pg385.25)
SEC_USER_DEFAULTS_ID	(pg374.24)
SEC_USER_DEFAULTS_NAME	(pg375.19)
SEC_WANT_AX	(pg376.29)
SEC_WANT_READ	(pg376.27)

+++ appendix +++

SEC_WANT_WRITE	(pg376.28)
SEC_WILDCARD_TIME	(pg373.13)
SEC_XIN_PROTOCOL	(pg388.38)
SEC_XNOT_IN_PROTOCOL	(pg388.39)
SEC_access_restrictions	(pg385.2)
SEC_access_restrictions_typ	(pg385.2)
SEC_access_wanted_typ	(pg376.25)
SEC_admin_class_typ	(pg384.25)
SEC_cn_typ	(pg371.38)
SEC_comments_typ	(pg371.29)
SEC_conditional_typ	(pg376.15)
SEC_db2_acct_typ	(pg371.35)
SEC_dbinfo_desc_typ	(pg383.16)
SEC_dbinfo_list_typ	(pg383.22)
SEC_dbinfo_opt_typ	(pg383.9)
SEC_dbname_typ	(pg371.32)
SEC_dbpwd_typ	(pg371.34)
SEC_debug_message	(pg406.4)
SEC_debug_options_typ	(pg370.29)
SEC_defs	(pg370.21)
SEC_despwd_typ	(pg371.30)
SEC_dev_err	(pg388.46)
SEC_device_class_typ	(pg375.47)
SEC_dn_typ	(pg371.37)
SEC_eDB2Pwd_typ	(pg371.36)
SEC_edbpwd_typ	(pg371.33)
SEC_endpt_addr_typ	(pg371.27)
SEC_err_CreateProcess_failed	(pg404.15)
SEC_err_QMA_null_pointer	(pg404.27)
SEC_err_SLU_not_found	(pg406.15)
SEC_err_access_denied	(pg390.39)
SEC_err_admingrp_not_found	(pg399.23)
SEC_err_already_logon	(pg406.42)
SEC_err_ax_denied	(pg389.41)
SEC_err_bad_filename	(pg398.2)
SEC_err_bad_import_format	(pg398.18)
SEC_err_bad_import_version	(pg398.6)
SEC_err_bad_magic	(pg392.36)
SEC_err_bad_service	(pg392.48)
SEC_err_batch_update_dn_by_id	(pg400.7)
SEC_err_batch_update_prin_by_cn	(pg404.45)
SEC_err_cannot_specify_dbmap	(pg401.19)
SEC_err_cannot_specify_primary	(pg397.18)
SEC_err_cannot_specify_sessgrp	(pg400.27)
SEC_err_cannot_update_primary	(pg396.21)
SEC_err_cant_decode_AR	(pg390.1)
SEC_err_cant_remove_inheritance	(pg390.48)
SEC_err_ce_dom_not_found	(pg399.48)

+++ appendix +++

SEC_err_cn_size_too_short (pg404.48)
SEC_err_conflicting_obj_class (pg398.32)
SEC_err_convert_bad_character (pg403.24)
SEC_err_convert_duplicate_name (pg403.27)
SEC_err_convert_name_exceed_39_characters (pg403.30)
SEC_err_create_README (pg402.10)
SEC_err_create_SEC_anl_success_file (pg404.18)
SEC_err_create_tls_index (pg406.24)
SEC_err_credential_not_in_environment (pg390.22)
SEC_err_data_segment_alloc (pg404.24)
SEC_err_database_level_not_match (pg402.1)
SEC_err_db2_acctname_max_length_reached (pg402.25)
SEC_err_db2_not_found (pg402.16)
SEC_err_db2_pwd_max_length_reached (pg402.22)
SEC_err_db2_time_out_of_range (pg402.28)
SEC_err_db_add_denied (pg401.10)
SEC_err_db_delete_denied (pg401.4)
SEC_err_db_update_denied (pg401.7)
SEC_err_dbinfo_not_found (pg400.34)
SEC_err_dbmap_denied (pg401.13)
SEC_err_debug_info_only (pg403.18)
SEC_err_deleted_not_found (pg394.6)
SEC_err_device_access_denied (pg392.27)
SEC_err_device_expired (pg393.15)
SEC_err_dipl_copyright_failed (pg406.39)
SEC_err_dn_size_too_short (pg393.3)
SEC_err_duplicate_ce_dom (pg400.4)
SEC_err_duplicate_db2 (pg402.19)
SEC_err_duplicate_dbinfo (pg400.40)
SEC_err_duplicate_deleted (pg394.12)
SEC_err_duplicate_funcmbr (pg394.39)
SEC_err_duplicate_function (pg394.36)
SEC_err_duplicate_group (pg394.9)
SEC_err_duplicate_logon (pg390.42)
SEC_err_duplicate_map_prin (pg399.45)
SEC_err_duplicate_namemap (pg400.37)
SEC_err_duplicate_object (pg391.24)
SEC_err_duplicate_rm_config (pg402.34)
SEC_err_duplicate_system (pg394.15)
SEC_err_exec_failed (pg404.9)
SEC_err_export_denied (pg398.46)
SEC_err_ext_authent_init_library_failed (pg402.46)
SEC_err_ext_authent_invalid_parameters (pg403.4)
SEC_err_ext_authent_lib_interface_missing (pg405.10)
SEC_err_ext_authent_missing_map_entry (pg405.3)
SEC_err_ext_authent_mode_undetermined (pg403.7)
SEC_err_ext_authent_validation_failed (pg403.1)
SEC_err_ext_validation_pwd_failed (pg404.34)

+++ appendix +++

SEC_err_fmbr_add_denied	(pg396.6)
SEC_err_fmbr_delete_denied	(pg396.9)
SEC_err_fmbr_read_denied	(pg397.41)
SEC_err_fork_failed	(pg404.12)
SEC_err_fun_read_denied	(pg397.37)
SEC_err_func_add_denied	(pg395.48)
SEC_err_func_delete_denied	(pg396.3)
SEC_err_funcmbr_not_found	(pg394.33)
SEC_err_funcmbr_out_of_sync	(pg394.42)
SEC_err_function_not_found	(pg394.30)
SEC_err_functionality_not_supported	(pg404.30)
SEC_err_get_NT_credential	(pg406.48)
SEC_err_get_process_access_token	(pg401.37)
SEC_err_get_tls_value	(pg406.30)
SEC_err_group_expired	(pg393.18)
SEC_err_group_not_found	(pg394.3)
SEC_err_groups_out_of_sync	(pg396.18)
SEC_err_grp_read_denied	(pg397.33)
SEC_err_illegal_instruction	(pg393.6)
SEC_err_import_denied	(pg399.4)
SEC_err_in_GetTokenInfo	(pg401.43)
SEC_err_incompatible_client	(pg403.13)
SEC_err_insufficient_token_buffer	(pg401.40)
SEC_err_invalid_admin_class	(pg391.12)
SEC_err_invalid_admin_group_opt	(pg396.33)
SEC_err_invalid_admin_opt	(pg397.3)
SEC_err_invalid_admingrp	(pg399.33)
SEC_err_invalid_admingrp_opt	(pg399.28)
SEC_err_invalid_allow_override	(pg400.24)
SEC_err_invalid_annot_security	(pg394.21)
SEC_err_invalid_call	(pg395.6)
SEC_err_invalid_char_set	(pg398.11)
SEC_err_invalid_comments	(pg391.30)
SEC_err_invalid_comments_opt	(pg396.48)
SEC_err_invalid_comments_search	(pg392.39)
SEC_err_invalid_conditional	(pg393.39)
SEC_err_invalid_db_filter	(pg400.43)
SEC_err_invalid_dbname	(pg401.1)
SEC_err_invalid_dbopt	(pg401.16)
SEC_err_invalid_dbpassword	(pg400.46)
SEC_err_invalid_dev_class_opt	(pg397.9)
SEC_err_invalid_dev_security	(pg391.42)
SEC_err_invalid_device_class	(pg391.9)
SEC_err_invalid_device_name	(pg392.33)
SEC_err_invalid_dn_length	(pg392.45)
SEC_err_invalid_domain	(pg398.29)
SEC_err_invalid_dweek_range	(pg395.3)
SEC_err_invalid_epassword	(pg399.36)

+++ appendix +++

SEC_err_invalid_error_opt	(pg396.45)
SEC_err_invalid_event_type	(pg401.46)
SEC_err_invalid_exp_time_opt	(pg396.36)
SEC_err_invalid_failed_opt	(pg396.42)
SEC_err_invalid_failure_mins	(pg393.27)
SEC_err_invalid_func_def	(pg391.45)
SEC_err_invalid_function_name	(pg393.9)
SEC_err_invalid_grace_period	(pg393.24)
SEC_err_invalid_handle	(pg395.21)
SEC_err_invalid_handle_ssn	(pg395.9)
SEC_err_invalid_hour_range	(pg394.48)
SEC_err_invalid_import_param	(pg398.26)
SEC_err_invalid_info_type	(pg390.30)
SEC_err_invalid_key_num	(pg392.18)
SEC_err_invalid_language	(pg394.27)
SEC_err_invalid_language_opt	(pg397.6)
SEC_err_invalid_log	(pg391.39)
SEC_err_invalid_logon_time	(pg393.12)
SEC_err_invalid_max_sess_opt	(pg397.15)
SEC_err_invalid_max_sessions	(pg391.21)
SEC_err_invalid_member_deletion	(pg392.21)
SEC_err_invalid_min_range	(pg394.45)
SEC_err_invalid_name	(pg390.33)
SEC_err_invalid_nbr_bytes	(pg401.25)
SEC_err_invalid_nbr_options	(pg391.3)
SEC_err_invalid_obj_class_opt	(pg397.12)
SEC_err_invalid_object_class	(pg391.6)
SEC_err_invalid_object_filter	(pg391.33)
SEC_err_invalid_object_name	(pg391.27)
SEC_err_invalid_objopt	(pg393.45)
SEC_err_invalid_option_in_fnfork_exec	(pg404.6)
SEC_err_invalid_param	(pg390.36)
SEC_err_invalid_password	(pg389.25)
SEC_err_invalid_password_opt	(pg396.30)
SEC_err_invalid_pc_admin	(pg400.13)
SEC_err_invalid_pointer	(pg401.22)
SEC_err_invalid_primary	(pg393.36)
SEC_err_invalid_pwd_aging	(pg392.6)
SEC_err_invalid_pwd_attempts	(pg392.9)
SEC_err_invalid_pwd_chg_upon_reset	(pg394.24)
SEC_err_invalid_pwd_memory	(pg392.15)
SEC_err_invalid_pwd_min_length	(pg392.3)
SEC_err_invalid_pwd_renewal_days	(pg393.21)
SEC_err_invalid_pwd_spec_char	(pg391.48)
SEC_err_invalid_pwd_suspending	(pg392.12)
SEC_err_invalid_relogon	(pg399.16)
SEC_err_invalid_rm_config_data	(pg402.37)
SEC_err_invalid_search_param	(pg393.30)

+++ appendix +++

SEC_err_invalid_service_logon	(pg390.19)
SEC_err_invalid_service_name	(pg398.40)
SEC_err_invalid_sess_override	(pg391.18)
SEC_err_invalid_sessgrp	(pg400.21)
SEC_err_invalid_session_nbr	(pg390.27)
SEC_err_invalid_success_opt	(pg396.39)
SEC_err_invalid_sysopt	(pg393.42)
SEC_err_invalid_system_filter	(pg391.36)
SEC_err_invalid_system_options	(pg394.18)
SEC_err_invalid_tag_type	(pg395.24)
SEC_err_invalid_termid	(pg396.15)
SEC_err_invalid_terminal_name	(pg396.27)
SEC_err_invalid_time	(pg391.15)
SEC_err_invalid_time_range	(pg397.21)
SEC_err_invalid_username	(pg406.33)
SEC_err_language_not_found	(pg390.9)
SEC_err_logonui	(pg406.36)
SEC_err_logonui_cancel	(pg406.45)
SEC_err_make_log_directory	(pg402.7)
SEC_err_map_cn_duplicate	(pg404.42)
SEC_err_map_cn_not_found	(pg404.39)
SEC_err_map_prin_not_found	(pg399.42)
SEC_err_max_sessions	(pg390.45)
SEC_err_maximum_logons_reached	(pg396.24)
SEC_err_mbr_read_denied	(pg397.30)
SEC_err_member_add_denied	(pg395.30)
SEC_err_member_delete_denied	(pg395.42)
SEC_err_missing_license	(pg399.39)
SEC_err_namemap_not_found	(pg400.31)
SEC_err_no_SLU_init	(pg405.43)
SEC_err_no_logon	(pg390.5)
SEC_err_no_special_char	(pg395.18)
SEC_err_no_such_service	(pg392.24)
SEC_err_nomap	(pg401.28)
SEC_err_nomemory_for_terminal_listing	(pg401.31)
SEC_err_not_member	(pg389.29)
SEC_err_not_user	(pg392.30)
SEC_err_obj_add_denied	(pg395.45)
SEC_err_obj_delete_denied	(pg395.33)
SEC_err_obj_read_denied	(pg397.24)
SEC_err_obj_update_denied	(pg395.27)
SEC_err_object_not_found	(pg389.45)
SEC_err_open_key_failed	(pg406.18)
SEC_err_other_error	(pg389.21)
SEC_err_password_expired	(pg395.15)
SEC_err_permission_denied_not_SysAdmin	(pg403.48)
SEC_err_permission_denied_on_NT	(pg403.42)
SEC_err_permission_denied_on_UNIX	(pg403.45)

+++ appendix +++

SEC_err_pid_not_found	(pg400.10)
SEC_err_primary_not_found	(pg393.33)
SEC_err_process_create_time	(pg401.34)
SEC_err_putenv_failed	(pg404.3)
SEC_err_pwd_attempts_exceeded	(pg395.12)
SEC_err_pwd_len_out_of_range	(pg397.45)
SEC_err_pwd_update_denied	(pg395.39)
SEC_err_query_key_failed	(pg406.21)
SEC_err_read_denied	(pg389.33)
SEC_err_rm_config_access	(pg402.40)
SEC_err_rm_config_group_not_found	(pg402.43)
SEC_err_rm_config_not_found	(pg402.31)
SEC_err_run_SEC_convert	(pg403.21)
SEC_err_run_SEC_init_on_MPE	(pg403.39)
SEC_err_run_SEC_init_on_NT	(pg403.33)
SEC_err_run_SEC_init_on_UNIX	(pg403.36)
SEC_err_sec_map_ext_auth_access	(pg405.7)
SEC_err_service_is_remote	(pg402.4)
SEC_err_sessgrp_not_found	(pg400.17)
SEC_err_set_tls_value	(pg406.27)
SEC_err_signals_undefined	(pg403.10)
SEC_err_stale_session	(pg399.10)
SEC_err_sys_read_denied	(pg397.27)
SEC_err_system_not_found	(pg393.48)
SEC_err_terminate_logon_denied	(pg395.36)
SEC_err_too_many_termids	(pg396.12)
SEC_err_undefined_object	(pg390.12)
SEC_err_user_expired	(pg392.42)
SEC_err_write_README	(pg402.13)
SEC_err_write_denied	(pg389.37)
SEC_err_zero_session	(pg404.21)
SEC_func_list_typ	(pg385.45)
SEC_function_list_typ	(pg383.44)
SEC_function_members_typ	(pg384.4)
SEC_function_name_typ	(pg371.28)
SEC_get_language	(pg372.27)
SEC_guid_typ	(pg386.26)
SEC_id_typ	(pg373.32)
SEC_identification_typ	(pg385.10)
SEC_info_type_typ	(pg386.12)
SEC_information_message	(pg406.7)
SEC_invalid_SLU_typ	(pg406.10)
SEC_language_id_typ	(pg372.6)
SEC_language_typ	(pg372.33)
SEC_leftoff_typ	(pg384.16)
SEC_logged_in_by_tlcred	(pg405.15)
SEC_logon_leftoff_typ	(pg385.15)
SEC_maxnamelength	(pg371.6)

+++ appendix +++

SEC_maxpasswordlength	(pg371.8)
SEC_maxterminallength	(pg371.7)
SEC_member_list_typ	(pg383.34)
SEC_memlist_typ	(pg387.19)
SEC_memlist_type_typ	(pg387.9)
SEC_name_struct_typ	(pg386.35)
SEC_name_typ	(pg371.25)
SEC_object_class_typ	(pg375.33)
SEC_object_desc_typ	(pg379.38)
SEC_object_info_typ	(pg388.18)
SEC_object_list_typ	(pg379.48)
SEC_object_opt_typ	(pg378.45)
SEC_oldpassword_typ	(pg371.31)
SEC_only_supported_on_NT	(pg405.46)
SEC_optdbinfo_typ	(pg382.43)
SEC_optobj_typ	(pg376.34)
SEC_optsys_typ	(pg380.39)
SEC_password_typ	(pg371.24)
SEC_pro_err	(pg388.44)
SEC_rm_config_typ	(pg387.32)
SEC_sct_err	(pg388.45)
SEC_security_defaults_typ	(pg388.9)
SEC_security_info_typ	(pg386.44)
SEC_sess_proc_rec_typ	(pg388.32)
SEC_slu_accounting_typ	(pg386.28)
SEC_stats_desc_typ	(pg384.40)
SEC_system_desc_typ	(pg382.30)
SEC_system_opt_typ	(pg382.4)
SEC_terminal_typ	(pg371.26)
SEC_time_not_available	(pg405.37)
SEC_time_range_typ	(pg373.23)
SEC_tlcred_access_error	(pg405.18)
SEC_tlcred_name_invalid	(pg405.28)
SEC_tlcred_unavailable	(pg405.21)
SEC_update_info_typ	(pg387.7)
SEC_update_typ	(pg385.21)
SEC_user_logon_list_typ	(pg384.14)
SEC_user_tlcred_unknown	(pg405.25)
SQIDEFS	(pg439.16)
SQI_ASCII	(pg440.20)
SQI_BAD_DB_VERSION	(pg442.43)
SQI_BAD_USER	(pg442.36)
SQI_BAD_VERSION	(pg443.19)
SQI_BIND_RESULT_MISSING	(pg442.42)
SQI_BLOB_ALLOC_LEN	(pg442.1)
SQI_BLOB_DATA_OFFSET	(pg441.48)
SQI_BLOB_TYPE	(pg442.3)
SQI_BLOB_TYP_LEN	(pg442.2)

+++ appendix +++

<u>SQI_CANCELLED</u>	(pg442.33)
<u>SQI_CONNECTION_BROKEN</u>	(pg443.13)
<u>SQI_CONVERSION_ERROR_INDICATOR</u>	(pg439.27)
<u>SQI_COR_MSG_FUNC_CODE</u>	(pg442.20)
<u>SQI_DATA_TYPE_NOT_SUPPORTED</u>	(pg442.46)
<u>SQI_DB_CANCELLED</u>	(pg443.35)
<u>SQI_DB_MSG_FUNC_CODE</u>	(pg442.21)
<u>SQI_DB_SERVER_NOT_AVAIL</u>	(pg442.35)
<u>SQI_DEFAULT_BLOB_BUF</u>	(pg442.4)
<u>SQI_DUMMY_BLOB_LEN</u>	(pg442.9)
<u>SQI_DUPLICATE_COLUMN_NAME</u>	(pg443.32)
<u>SQI_END_OF_FETCH</u>	(pg443.33)
<u>SQI_ERR_BAD_DB_VERSION</u>	(pg444.30)
<u>SQI_ERR_BAD_USER</u>	(pg444.9)
<u>SQI_ERR_BAD_VERSION</u>	(pg445.42)
<u>SQI_ERR_BIND_RESULT_MISSING</u>	(pg444.27)
<u>SQI_ERR_CANCELLED</u>	(pg443.48)
<u>SQI_ERR_CONNECTION_BROKEN</u>	(pg445.24)
<u>SQI_ERR_DATA_TYPE_NOT_SUPPORTED</u>	(pg444.18)
<u>SQI_ERR_DB_CANCELLED</u>	(pg446.29)
<u>SQI_ERR_DB_SERVER_NOT_AVAIL</u>	(pg444.6)
<u>SQI_ERR_DUPLICATE_COLUMN_NAME</u>	(pg446.20)
<u>SQI_ERR_END_OF_FETCH</u>	(pg446.23)
<u>SQI_ERR_FETCH_WITHOUT_EXECUTION</u>	(pg444.33)
<u>SQI_ERR_INT_INVALID_FIELD_SIZE</u>	(pg445.15)
<u>SQI_ERR_INVALID_BIND_RESULT</u>	(pg444.24)
<u>SQI_ERR_INVALID_BIND_VALUE</u>	(pg444.21)
<u>SQI_ERR_INVALID_COLUMN_NAME</u>	(pg446.8)
<u>SQI_ERR_INVALID_CURSOR_HANDLE</u>	(pg444.15)
<u>SQI_ERR_INVALID_DATA_TYPE</u>	(pg444.39)
<u>SQI_ERR_INVALID_FIELD_POS</u>	(pg445.33)
<u>SQI_ERR_INVALID_LOGON_CALL</u>	(pg445.48)
<u>SQI_ERR_INVALID_MESSAGE</u>	(pg445.18)
<u>SQI_ERR_INVALID_PROCEDURE_CALL</u>	(pg445.21)
<u>SQI_ERR_INVALID_SESSION_HANDLE</u>	(pg444.12)
<u>SQI_ERR_INVALID_SQLDA_SIZE</u>	(pg445.36)
<u>SQI_ERR_INVALID_TABLE_NAME</u>	(pg446.14)
<u>SQI_ERR_LOGON_TWICE</u>	(pg444.42)
<u>SQI_ERR_NOT_A_SELECT_QUERY</u>	(pg445.39)
<u>SQI_ERR_NO_CONFIG_FILE</u>	(pg445.27)
<u>SQI_ERR_NO_CURRENT_QUERY</u>	(pg444.45)
<u>SQI_ERR_NO_LICENSE</u>	(pg444.48)
<u>SQI_ERR_NO_RESOURCES</u>	(pg444.3)
<u>SQI_ERR_NO_SUCH_DBMS</u>	(pg445.45)
<u>SQI_ERR_OUTPUT_WARNING</u>	(pg446.26)
<u>SQI_ERR_POS_CONFIG_FILE</u>	(pg445.30)
<u>SQI_ERR_SQLCA_MISSING</u>	(pg445.6)
<u>SQI_ERR_SQLDA_MISSING</u>	(pg445.9)

+++ appendix +++

<u>SQI_ERR_SQL_SERVICE_NOT_AVAIL</u>	(pg445.12)
<u>SQI_ERR_SYNTAX_ERROR</u>	(pg446.11)
<u>SQI_ERR_TABLE_ALREADY_EXIST</u>	(pg446.17)
<u>SQI_ERR_USER_NOT_ALLOWED</u>	(pg446.3)
<u>SQI_ERR_VALUE_OUT_OF_RANGE</u>	(pg444.36)
<u>SQI_FETCH_WITHOUT_EXECUTION</u>	(pg442.44)
<u>SQI_FPnum</u>	(pg440.19)
<u>SQI_GOOD_RESULT_INDICATOR</u>	(pg439.25)
<u>SQI_INT_INVALID_FIELD_SIZE</u>	(pg443.10)
<u>SQI_INT_MSG_FUNC_CODE</u>	(pg442.22)
<u>SQI_INVALID_BIND_RESULT</u>	(pg442.41)
<u>SQI_INVALID_BIND_VALUE</u>	(pg442.40)
<u>SQI_INVALID_COLUMN_NAME</u>	(pg443.28)
<u>SQI_INVALID_CURSOR_HANDLE</u>	(pg442.38)
<u>SQI_INVALID_DATA_TYPE</u>	(pg442.39)
<u>SQI_INVALID_FIELD_POS</u>	(pg443.16)
<u>SQI_INVALID_LOGON_CALL</u>	(pg443.21)
<u>SQI_INVALID_MESSAGE</u>	(pg443.11)
<u>SQI_INVALID_PROCEDURE_CALL</u>	(pg443.12)
<u>SQI_INVALID_SESSION_HANDLE</u>	(pg442.37)
<u>SQI_INVALID_SQLDA_SIZE</u>	(pg443.17)
<u>SQI_INVALID_TABLE_NAME</u>	(pg443.30)
<u>SQI_LOGON_TWICE</u>	(pg442.47)
<u>SQI_MAX_SQL_ERRD_SIZE</u>	(pg440.41)
<u>SQI_MAX_SQL_ERRP_SIZE</u>	(pg440.40)
<u>SQI_MAX_SQL_EXT_SIZE</u>	(pg440.42)
<u>SQI_MAX_SQL_ID_SIZE</u>	(pg440.39)
<u>SQI_MAX_SQL_MSG_SIZE</u>	(pg440.38)
<u>SQI_MAX_SQL_NAME_SIZE</u>	(pg440.44)
<u>SQI_NOT_A_SELECT_QUERY</u>	(pg443.18)
<u>SQI_NO_CONFIG_FILE</u>	(pg443.14)
<u>SQI_NO_CURRENT_QUERY</u>	(pg442.48)
<u>SQI_NO_LICENSE</u>	(pg443.1)
<u>SQI_NO_RESOURCES</u>	(pg442.34)
<u>SQI_NO_SUCH_DBMS</u>	(pg443.20)
<u>SQI_NULL_INDICATOR</u>	(pg439.26)
<u>SQI_OTHER_ERROR</u>	(pg442.32)
<u>SQI_OUTPUT_WARNING</u>	(pg443.34)
<u>SQI_POS_CONFIG_FILE</u>	(pg443.15)
<u>SQI_SQLCA_ID</u>	(pg439.21)
<u>SQI_SQLCA_MISSING</u>	(pg443.7)
<u>SQI_SQLDA_ID</u>	(pg439.22)
<u>SQI_SQLDA_MISSING</u>	(pg443.8)
<u>SQI_SQL_SERVICE_NOT_AVAIL</u>	(pg443.9)
<u>SQI_SQL_WARNING</u>	(pg439.23)
<u>SQI_SYNTAX_ERROR</u>	(pg443.29)
<u>SQI_TABLE_ALREADY_EXIST</u>	(pg443.31)
<u>SQI_USER_NOT_ALLOWED</u>	(pg443.22)

+++ appendix +++

<u>SQI_VALUE_OUT_OF_RANGE</u>	(pg442.45)
<u>SQI_binary_object</u>	(pg440.25)
<u>SQI_blob_typ</u>	(pg441.46)
<u>SQI_boolean</u>	(pg440.12)
<u>SQI_byte</u>	(pg440.13)
<u>SQI_cor_msg</u>	(pg443.41)
<u>SQI_cursor_handle_typ</u>	(pg439.30)
<u>SQI_data_type_typ</u>	(pg440.27)
<u>SQI_date</u>	(pg440.21)
<u>SQI_db_msg</u>	(pg443.43)
<u>SQI_dbms_allbase</u>	(pg439.38)
<u>SQI_dbms_db2</u>	(pg439.40)
<u>SQI_dbms_info_typ</u>	(pg440.6)
<u>SQI_dbms_mssql</u>	(pg439.39)
<u>SQI_dbms_oracle</u>	(pg439.36)
<u>SQI_dbms_sybase</u>	(pg439.37)
<u>SQI_dbms_typ</u>	(pg439.35)
<u>SQI_dbms_version_typ</u>	(pg439.46)
<u>SQI_int_msg</u>	(pg443.42)
<u>SQI_long</u>	(pg440.17)
<u>SQI_menu</u>	(pg440.23)
<u>SQI_null</u>	(pg440.26)
<u>SQI_session_handle_typ</u>	(pg439.29)
<u>SQI_short</u>	(pg440.15)
<u>SQI_sqlca_typ</u>	(pg441.16)
<u>SQI_sqlda_typ</u>	(pg441.35)
<u>SQI_sqlvar_typ</u>	(pg441.26)
<u>SQI_text_object</u>	(pg440.24)
<u>SQI_time</u>	(pg440.22)
<u>SQI_unsigned_byte</u>	(pg440.14)
<u>SQI_unsigned_long</u>	(pg440.18)
<u>SQI_unsigned_short</u>	(pg440.16)
<u>SYSLOG_COUNTERS</u>	(pg501.34)
<u>SYSLOG_ID_01</u>	(pg501.13)
<u>SYSLOG_ID_02</u>	(pg501.14)
<u>SYSLOG_ID_03</u>	(pg501.15)
<u>SYSLOG_ID_04</u>	(pg501.16)
<u>SYSLOG_ID_05</u>	(pg501.17)
<u>SYSLOG_ID_06</u>	(pg501.18)
<u>SYSLOG_ID_07</u>	(pg501.19)
<u>SYSLOG_ID_08</u>	(pg501.20)
<u>SYSLOG_ID_09</u>	(pg501.21)
<u>SYSLOG_ID_10</u>	(pg501.22)
<u>SYSLOG_ID_11</u>	(pg501.23)
<u>SYSLOG_ID_12</u>	(pg501.24)
<u>SYSLOG_ID_13</u>	(pg501.25)
<u>SYSLOG_ID_14</u>	(pg501.26)
<u>SYSLOG_ID_15</u>	(pg501.27)

+++ appendix +++

SYSLOG_ID_16	(pg501.28)
SYSLOG_ID_17	(pg501.29)
SYSLOG_ID_18	(pg501.30)
SYSLOG_ID_19	(pg501.31)
SYSLOG_ID_20	(pg501.32)
TRUE	(pg490.45)
WQS_COUNTER	(pg462.2)
WQS_COUNTER_typ	(pg462.10)
WQS_EXCEPTION	(pg462.43)
WQS_FN_FIELDS	(pg456.10)
WQS_MAX_FUNC	(pg461.47)
WQS_QUEUE_COUNTER	(pg462.26)
WQS_QUEUE_COUNTER_typ	(pg462.35)
WQS_ROWID_SIZE	(pg456.20)
WQS_SERVER_STATS	(pg462.13)
WQS_SERVER_STATS_typ	(pg462.21)
WQS_ascending_order	(pg456.47)
WQS_busy_must_be_set	(pg460.25)
WQS_busy_not_ok	(pg460.23)
WQS_busy_ok	(pg460.24)
WQS_close_queue_func	(pg461.30)
WQS_count_entries_func	(pg461.31)
WQS_create_queue_func	(pg461.25)
WQS_create_workspace_func	(pg461.41)
WQS_current_time	(pg456.4)
WQS_db2_index_tbl_space	(pg455.47)
WQS_db2_max_col	(pg456.17)
WQS_db2_user_tbl_space	(pg455.46)
WQS_default_priority	(pg456.9)
WQS_defs	(pg455.27)
WQS_delete_and_insert_entry_func	(pg461.45)
WQS_delete_current	(pg457.4)
WQS_delete_entry_func	(pg461.35)
WQS_delete_none	(pg457.3)
WQS_delete_previous	(pg457.5)
WQS_delete_queue_func	(pg461.26)
WQS_delete_typ	(pg457.6)
WQS_delete_workspace_func	(pg461.43)
WQS_descending_order	(pg456.48)
WQS_dump_handle_typ	(pg457.21)
WQS_empty_queue_func	(pg461.40)
WQS_end_dump_func	(pg461.39)
WQS_entry_id_typ	(pg457.23)
WQS_err_26_conversion	(pg463.43)
WQS_err_actlog_count	(pg466.18)
WQS_err_all_qs_must_be_closed	(pg465.2)
WQS_err_bad_version	(pg463.33)
WQS_err_bad_workspace_name	(pg464.48)

+++ appendix +++

WQS_err_bogus_session_handle	(pg465.8)
WQS_err_cannot_modify_field	(pg465.36)
WQS_err_cannot_update_two	(pg466.16)
WQS_err_cor_abort_msg	(pg466.6)
WQS_err_cor_reject_msg	(pg466.8)
WQS_err_corrupted_queue_file	(pg463.41)
WQS_err_corrupted_ws_desc	(pg465.10)
WQS_err_cursor_too_large	(pg464.23)
WQS_err_db_logon	(pg464.3)
WQS_err_db_open	(pg464.1)
WQS_err_different_servers	(pg466.12)
WQS_err_dup_unique_val	(pg464.35)
WQS_err_entry_not_tagged	(pg464.21)
WQS_err_failed_to_exec_program	(pg466.2)
WQS_err_failed_to_fork_program	(pg465.48)
WQS_err_field_desc_change	(pg464.9)
WQS_err_field_specified_twice	(pg463.23)
WQS_err_illegal_new_name	(pg465.4)
WQS_err_illegal_to_drop fld	(pg465.19)
WQS_err_illegal_write_field	(pg463.25)
WQS_err_info_only	(pg466.31)
WQS_err_internal_rpc_error	(pg464.15)
WQS_err_invalid_area_magic	(pg465.14)
WQS_err_invalid_dump_handle	(pg463.8)
WQS_err_invalid_dump_or_queue_handle	(pg465.40)
WQS_err_invalid_entry_id	(pg464.5)
WQS_err_invalid_field_length	(pg464.37)
WQS_err_invalid_field_type	(pg463.47)
WQS_err_invalid_field_unique_type	(pg466.4)
WQS_err_invalid_field_value	(pg464.33)
WQS_err_invalid_find_field	(pg464.7)
WQS_err_invalid_null_domain	(pg465.12)
WQS_err_invalid_queue_handle	(pg463.6)
WQS_err_invalid_security_id	(pg464.46)
WQS_err_invalid_server_number	(pg465.32)
WQS_err_invalid_service_handle	(pg463.4)
WQS_err_invalid_sort_field	(pg465.24)
WQS_err_invalid_sys_field	(pg463.18)
WQS_err_invalid_sys_field_data	(pg463.20)
WQS_err_invalid_table_name	(pg465.34)
WQS_err_invalid_workspace	(pg464.25)
WQS_err_load_new_queue_desc	(pg464.27)
WQS_err_local_func_only	(pg464.40)
WQS_err_max_db2_col_exceeded	(pg466.25)
WQS_err_needs_file_to_db_convert	(pg465.30)
WQS_err_no_cursor_open	(pg464.19)
WQS_err_no_entry_selected	(pg463.31)
WQS_err_no_resources	(pg463.2)

+++ appendix +++

WQS_err_no_user_field	(pg465.17)
WQS_err_not_rendez	(pg466.14)
WQS_err_option_not_allowed	(pg466.21)
WQS_err_pid_is_not_lock_owner	(pg465.44)
WQS_err_prog_not_support_DB2	(pg466.29)
WQS_err_queue_already_defined	(pg463.37)
WQS_err_queue_def_changed	(pg465.6)
WQS_err_queue_in_use	(pg463.12)
WQS_err_queue_is_currently_opened	(pg465.46)
WQS_err_queue_not_defined	(pg463.14)
WQS_err_queue_not_empty	(pg463.10)
WQS_err_queue_not_open	(pg463.39)
WQS_err_queue_table_overflow	(pg466.23)
WQS_err_rendez_field_missing	(pg463.29)
WQS_err_required_field_missing	(pg463.27)
WQS_err_row_busy	(pg464.44)
WQS_err_security_violation	(pg463.45)
WQS_err_service_not_available	(pg463.35)
WQS_err_session_already_open	(pg465.42)
WQS_err_sql_overflow	(pg466.10)
WQS_err_table_not_built	(pg464.17)
WQS_err_too_many_opened_queues	(pg465.26)
WQS_err_undefined_user_field	(pg463.16)
WQS_err_upgrade_not_needed	(pg466.27)
WQS_err_workspace_not_created	(pg464.29)
WQS_err_workspace_not_empty	(pg464.31)
WQS_err_wqss_memory_limit	(pg465.38)
WQS_err_wrong_server	(pg464.42)
WQS_err_ws_already_defined	(pg465.28)
WQS_fetch_spec	(pg460.34)
WQS_fetch_spec_typ	(pg461.19)
WQS_field_name_typ	(pg457.29)
WQS_field_type_access	(pg456.32)
WQS_field_type_boolean	(pg456.33)
WQS_field_type_date	(pg456.31)
WQS_field_type_decimal	(pg456.35)
WQS_field_type_document	(pg456.28)
WQS_field_type_folder	(pg456.29)
WQS_field_type_int	(pg456.30)
WQS_field_type_null	(pg456.34)
WQS_field_type_number	(pg456.24)
WQS_field_type_selection	(pg456.27)
WQS_field_type_string	(pg456.25)
WQS_field_type_time	(pg456.26)
WQS_field_unique_typ	(pg456.43)
WQS_get_queue_desc_func	(pg461.27)
WQS_get_workspace_info_func	(pg461.44)
WQS_incomplete_not_ok	(pg460.28)

+++ appendix +++

WQS_incomplete_ok	(pg460.29)
WQS_incomplete_only	(pg460.30)
WQS_incomplete_spec_typ	(pg460.31)
WQS_insert_entry_func	(pg461.34)
WQS_invert	(pg456.41)
WQS_max_field_name_length	(pg456.13)
WQS_max_name_length	(pg456.12)
WQS_max_priority	(pg456.8)
WQS_max_queue_table	(pg456.16)
WQS_max_sys_fields	(pg456.11)
WQS_max_text_desc_length	(pg456.14)
WQS_min_priority	(pg456.7)
WQS_no_delay	(pg456.2)
WQS_no_timeout	(pg456.3)
WQS_not_invert	(pg456.40)
WQS_open_queue_func	(pg461.29)
WQS_prec_scale	(pg458.6)
WQS_prec_scale_typ	(pg458.9)
WQS_queue_desc	(pg458.27)
WQS_queue_desc_typ	(pg458.40)
WQS_queue_entry_in	(pg460.5)
WQS_queue_entry_in_typ	(pg460.11)
WQS_queue_entry_out	(pg460.14)
WQS_queue_entry_out_typ	(pg460.21)
WQS_queue_field_choice	(pg458.43)
WQS_queue_field_choice_typ	(pg459.9)
WQS_queue_field_typ	(pg456.36)
WQS_queue_handle_typ	(pg457.19)
WQS_queue_name_typ	(pg457.27)
WQS_read_dump_func	(pg461.38)
WQS_read_entry_func	(pg461.33)
WQS_read_queue_func	(pg461.32)
WQS_sort_order_typ	(pg457.1)
WQS_start_dump_func	(pg461.37)
WQS_status_spec_typ	(pg460.26)
WQS_sys_field_delaytime	(pg457.10)
WQS_sys_field_entrytime	(pg457.15)
WQS_sys_field_groupid	(pg457.14)
WQS_sys_field_name_typ	(pg457.17)
WQS_sys_field_pri	(pg457.8)
WQS_sys_field_status	(pg457.9)
WQS_sys_field_timeout	(pg457.11)
WQS_sys_field_userid	(pg457.13)
WQS_sys_field_value	(pg459.19)
WQS_sys_field_value_typ	(pg459.46)
WQS_unique_invert	(pg456.42)
WQS_update_entry_func	(pg461.36)
WQS_update_queue_func	(pg461.28)

+++ appendix +++

WQS_update_selected_func	(pg461.46)
WQS_update_workspace_func	(pg461.42)
WQS_user_field_desc	(pg458.12)
WQS_user_field_desc_typ	(pg458.22)
WQS_user_field_value	(pg459.12)
WQS_user_field_value_typ	(pg459.16)
WQS_workspace_name_typ	(pg457.25)
WQS_zother_error	(pg462.46)
andif	(pg490.34)
any_ptr	(pg491.24)
bool	(pg491.22)
byte	(pg491.23)
csm_err	(pg95.17)
csm_pro_err	(pg95.18)
efn_help	(pg499.16)
efn_unix	(pg499.17)
elif	(pg490.37)
err_AC	(pg494.21)
err_ADR	(pg496.30)
err_AE	(pg494.10)
err_AED	(pg497.12)
err_AIC	(pg497.19)
err_ANO	(pg494.11)
err_AOL	(pg496.39)
err_APD	(pg494.12)
err_ARM	(pg496.28)
err_ASF	(pg496.40)
err_ASH	(pg498.35)
err_BES	(pg495.31)
err_BJC	(pg498.25)
err_BKG	(pg494.45)
err_BR	(pg494.13)
err_BS	(pg494.14)
err_BSP	(pg494.6)
err_CAT	(pg495.29)
err_CATS	(pg498.37)
err_CB	(pg498.23)
err_CDB	(pg498.11)
err_CLT	(pg497.23)
err_CLU	(pg497.6)
err_CMT	(pg495.37)
err_CNF	(pg495.22)
err_CNT	(pg496.24)
err_CO	(pg497.47)
err_COD	(pg496.45)
err_COR	(pg497.2)
err_CPT	(pg498.21)
err_CQ	(pg494.41)

+++ appendix +++

err_CS	(pg498.31)
err_CSA	(pg496.10)
err_CSM	(pg495.20)
err_CSS	(pg497.46)
err_CT	(pg497.45)
err_CTF	(pg497.41)
err_DAM	(pg494.15)
err_DAR	(pg497.40)
err_DAS	(pg496.15)
err_DB	(pg498.18)
err_DBL	(pg497.11)
err_DBP	(pg494.42)
err_DBU	(pg498.29)
err_DBV	(pg494.40)
err_DC	(pg494.16)
err_DCH	(pg497.26)
err_DCL	(pg496.2)
err_DCQ	(pg496.12)
err_DD	(pg494.17)
err_DE	(pg494.18)
err_DEA	(pg497.7)
err_DEB	(pg496.9)
err_DEC	(pg496.1)
err_DED	(pg495.16)
err_DEE	(pg495.39)
err_DEM	(pg495.40)
err_DEO	(pg495.26)
err_DEP	(pg497.21)
err_DEQ	(pg495.34)
err_DER	(pg497.20)
err_DES	(pg496.47)
err_DEU	(pg497.5)
err_DEV	(pg497.15)
err_DEW	(pg495.28)
err_DEX	(pg496.44)
err_DEZ	(pg497.38)
err_DGP	(pg494.3)
err_DIS	(pg496.27)
err_DLI	(pg494.47)
err_DLS	(pg494.43)
err_DMS	(pg498.30)
err_DOC	(pg495.23)
err_DPD	(pg496.23)
err_DR	(pg494.22)
err_DSA	(pg496.29)
err_DSL	(pg496.19)
err_DT	(pg495.1)
err_DTI	(pg180.8)

+++ appendix +++

err_DTM	(pg494.23)
err_DTP	(pg495.38)
err_DTR	(pg494.46)
err_DVT	(pg498.27)
err_EBR	(pg498.22)
err_ED	(pg498.15)
err_EDT	(pg497.33)
err_ELA	(pg496.31)
err_EPA	(pg497.22)
err_ERM	(pg494.24)
err_ESA	(pg498.14)
err_ESC1	(pg499.1)
err_ESC2	(pg499.2)
err_ESC3	(pg499.3)
err_ESC4	(pg499.4)
err_ESC5	(pg499.5)
err_ESC6	(pg499.6)
err_ESC7	(pg499.7)
err_ESC8	(pg499.8)
err_FDT	(pg495.3)
err_FET	(pg497.1)
err_FFI	(pg498.6)
err_FIL	(pg496.43)
err_FLT	(pg495.2)
err_FMT	(pg495.46)
err_FND	(pg499.28)
err_FNS	(pg499.27)
err_FNT	(pg495.18)
err_FOR	(pg494.26)
err_FP	(pg495.32)
err_FS	(pg496.11)
err_FSM	(pg495.15)
err_FTM	(pg498.10)
err_FV	(pg495.41)
err_FW	(pg494.25)
err_GDB	(pg496.16)
err_GDD	(pg496.32)
err_GDS	(pg496.33)
err_GLO	(pg494.39)
err_GPS	(pg496.34)
err_GTI	(pg496.14)
err_GU	(pg497.37)
err_GUID	(pg498.26)
err_GV	(pg498.2)
err_HLP	(pg494.27)
err_HLT	(pg495.8)
err_HSM	(pg498.33)
err_IAI	(pg497.24)

+++ appendix +++

err_IBM	(pg497.25)
err_IC	(pg497.10)
err_ICR	(pg498.7)
err_IMC	(pg497.48)
err_IMS	(pg498.34)
err_INX	(pg495.33)
err_ISM	(pg495.11)
err_IST	(pg495.9)
err_IXA	(pg496.5)
err_IXS	(pg496.6)
err_KerNetBM	(pg493.41)
err_KerNetED	(pg493.44)
err_KerNetEPP	(pg493.48)
err_KerNetERP	(pg494.1)
err_KerNetIPP	(pg493.46)
err_KerNetIRP	(pg494.2)
err_KerNetMM	(pg493.40)
err_KerNetORP	(pg493.47)
err_KerNetQM	(pg493.42)
err_KerNetRIP	(pg497.13)
err_KerNetSM	(pg493.43)
err_KerNetSSL	(pg497.14)
err_KerNetSYS	(pg493.45)
err_Kernel	(pg493.39)
err_LAST	(pg499.9)
err_LOG	(pg497.9)
err_LSI	(pg498.36)
err_MD	(pg494.28)
err_MF	(pg494.29)
err_MKF	(pg497.8)
err_MON	(pg498.24)
err_MSG	(pg497.18)
err_NBD	(pg496.37)
err_NCH	(pg497.3)
err_NL	(pg498.3)
err_NLS	(pg496.7)
err_NMI	(pg498.9)
err_OAP	(pg497.36)
err_OCR	(pg497.35)
err_OD	(pg495.47)
err_ODD	(pg495.4)
err_ODS	(pg496.4)
err_ODT	(pg495.7)
err_ODX	(pg494.20)
err_OFA	(pg497.34)
err_OLD	(pg497.16)
err_OPM	(pg496.25)
err_ORA	(pg496.21)

+++ appendix +++

err_ORK	(pg495.45)
err_OSA	(pg495.19)
err_OSI	(pg494.38)
err_OSS	(pg495.5)
err_PA	(pg497.43)
err_PDB	(pg496.36)
err_PE	(pg498.17)
err_PEP	(pg494.4)
err_PI	(pg498.4)
err_PMC	(pg495.21)
err_PPM	(pg494.7)
err_PPR	(pg494.30)
err_PRG	(pg498.20)
err_PRI	(pg495.30)
err_PRS	(pg330.7)
err_PSC	(pg497.29)
err_PSI	(pg497.32)
err_PSM	(pg497.28)
err_PUR	(pg494.32)
err_QLG	(pg494.44)
err_QMA	(pg495.17)
err_QMR	(pg495.43)
err_QS	(pg495.44)
err_RFT	(pg498.8)
err_RG	(pg494.33)
err_RMK	(pg494.31)
err_ROT	(pg496.41)
err_RPC	(pg494.8)
err_SAS	(pg495.14)
err_SCA	(pg495.12)
err_SCH	(pg495.10)
err_SCL	(pg496.13)
err_SCN	(pg496.48)
err_SCP	(pg494.34)
err_SCT	(pg495.35)
err_SDC	(pg494.19)
err_SDS	(pg498.38)
err_SEC	(pg495.36)
err_SIC	(pg496.18)
err_SIT	(pg496.8)
err_SKF	(pg496.20)
err_SLM	(pg498.32)
err_SLP	(pg497.4)
err_SLT	(pg494.48)
err_SNM	(pg498.5)
err_SNT	(pg495.6)
err_SOD	(pg498.13)
err_SPP	(pg494.5)

+++ appendix +++

err_SQI	(pg496.42)
err_SRF	(pg496.26)
err_SRT	(pg495.42)
err_STO	(pg498.28)
err_SU	(pg497.42)
err_SX	(pg497.30)
err_SYS	(pg496.17)
err_SunIAF	(pg495.48)
err_SysV	(pg498.1)
err_TAB	(pg496.3)
err_TAP	(pg497.17)
err_TEX	(pg497.31)
err_TPI	(pg497.39)
err_TT	(pg498.19)
err_UFD	(pg494.37)
err_UNIX	(pg496.22)
err_UPD	(pg497.27)
err_VP	(pg497.44)
err_VW	(pg498.12)
err_WFI	(pg495.27)
err_WFL	(pg495.24)
err_WIP	(pg496.35)
err_WP	(pg494.35)
err_WPL	(pg495.25)
err_WQM	(pg494.36)
err_WQS	(pg496.46)
err_WRT	(pg495.13)
err_WSS	(pg496.38)
err_Winter	(pg494.9)
err_category	(pg493.32)
err_encode	(pg493.29)
err_function	(pg493.33)
err_number	(pg493.34)
error_typ	(pg493.26)
ffi_err	(pg203.21)
not	(pg490.33)
orif	(pg490.35)
proj_CT	(pg499.34)
severity_typ	(pg493.27)
sockaddr_storage	(pg483.20)
success	(pg493.38)

+++ appendix +++

APPENDIX B: NOTICES

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing 2-31 Roppongi 3-chome, Minato-ku Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the

+++ appendix +++

materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: <i> the exchange of information between independently created programs and other programs (including this one) and <ii> the mutual use of the information which has been exchanged, should contact:

IBM Corporation J46A/G4 555 Bailey Avenue San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

TRADEMARKS

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (R) or (TM), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

U.S. PATENTS DISCLOSURE

This product incorporates technology covered by one or more of the following patents: U.S. Patent Numbers: 6,094,505; 5,768,416; 5,625,465; 5,369,508; 5,258,855.



Program Number: 5724-R95

Printed in USA

SC19-2696-00

