File Manager for z/OS

Customization Guide

Version 11 Release 1

SC19-3133-01

Note
Before using this document, read the general information under "Notices" on page 469.

Second Edition (August 2011)

This edition applies to Version 11 Release 1 Modification Level 0 of IBM File Manager for z/OS (program number 5655-W47) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. For information on how to send comments, see "How to send your comments to IBM" on page xix.

This publication is available on the Web at:

www.ibm.com/software/awdtools/filemanager/

© Copyright IBM Corporation 2000, 2011.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this document xiii
Overview of File Manager xiii
Who might use this document xv
Experienced installers xv
Other documentation that you might need . . . xv
Document conventions xv
Syntax notation xvi
How to send your comments xviii

How to send your comments to IBM xix
If you have a technical problem xix

Summary of changes xxi

Part 1. Customizing File Manager . . 1

Chapter 1. Preparing to customize File Manager 3
Checklist for installing and customizing File Manager 3
Library names after you finish installing 3
Defining the FMN0PARM member 4
Alternatives for making File Manager available . . . 6
Concatenating libraries to the LINKLIST 6
Modifying the TSO logon procedure 7
Planning for running File Manager with or without APF-authorization 7
Deciding how to customize the File Manager audit facility 8

Chapter 2. Customizing the operating environment for File Manager 11
Modifying the ISPF environment 11
Adding File Manager to the ISPF menu 11
Defining File Manager in an ISPF command table 11
Making File Manager the default VSAM editor 12
Enabling File Manager to work with certain products 12
Enabling WebSphere MQ support 13
Customizing for processing COBOL copybooks 13
Customizing for processing PL/I include books 17
Customizing for processing HLASM copybooks 17
Customizing to use DFSORT to improve File Manager performance 17
Binding DB2 to use File Manager object access method (OAM) functions 20
Running File Manager with APF-authorization . . 21
Troubleshooting problems with APF-authorization 21
Language Environment considerations 22

Chapter 3. Customizing File Manager 23
Changing the default options 23
Setting the default national language 23
Changing the JCL skeleton for batch mode 24
Customizing File Manager batch return codes . . . 26
Changing the print and display translation tables. . 28
Changing the translation tables in English . . . 29
Changing the ASCII translation tables 29

Chapter 4. Customizing the File Manager security environment 31
Setting up the security environment by using RACF or an equivalent security product 31
Controlling access to File Manager functions with SAF 33
Controlling fullpack access to DASD volumes . . 35
Controlling Bypass Label Processing (BLP) . . . 37
Setting up the security environment by using FMNSECUR 38
Using FMNSECUR to protect DASD volumes from fullpack access 41
Syntax of the FMNS macro 41
Exit routine environment 42
Registers at entry 42
Parameter list contents 43
Registers at exit 43
Installing FMNSECUR using FMNUMODS . . . 43
Unprotected functions and profile names for protected functions 44

Chapter 5. Customizing the File Manager audit facility (FMN0POPT controlled auditing) 49
Setting the appropriate options to produce an audit trail 49
Using System Management Facilities (SMF) for audit logging 49
Recording an audit trail to a data set 50

Chapter 6. Customizing the File Manager audit facility (SAF-rule controlled auditing) 51
Controlling File Manager auditing using SAF . . . 51
When a security product other than RACF is in use 52
Implementing SAF-rule controlled auditing. . . 52
Understanding how File Manager uses SAF rules to control auditing 53
How File Manager determines whether audit log records should be written. 54
Controlling where File Manager writes audit log records 56
SAF rule examples 56
Controlling where File Manager writes audit log records 56
Controlling auditing of File Manager functions 57

© Copyright IBM Corp. 2000, 2011

File Manager auditing FACILITY and XFACILIT class resource names 59

Chapter 7. Customizing File Manager for national languages 61
Changing the print and display translation tables for languages other than English 61
Translating the message text 62
Providing a national language version of FMN0MENU 63
Translating the ISPF messages text. 63
Translating the panel text 63
Using the translated messages and panels 64
Customizing for the Japanese national language . . 65
Modifying the Japanese translation tables . . . 65
Changing the Japanese message text 65
Customizing for the Korean national language . . 66
Modifying the Korean translation tables 66
Changing the Korean message text 66

Chapter 8. Customizing File Manager to use library management system libraries 67
Accessing source code in CA-Panvalet libraries . . 67
Other library management systems 68
Accessing source code in an LMS with SUBSYS interface 68
Accessing source code in an LMS without SUBSYS interface 68
Capabilities provided by File Manager via FMNCRAEX 69
Restrictions 69
Functions which must be provided by FMNCRAEX 69
Initialize and terminate the exit. 69
Validate the library 70
Extract a member from the library. 70
Get member information 70
Get display information 70
The sample exit 70
Writing your own exit 71
Performance considerations 72

Chapter 9. Customizing File Manager to use an I/O exit 73
Exit protocol 74
Writing your exit 74
Exit control block data names 75
Using the I/O exit control block 79
Function codes and the flow of I/O exit processing 79
Information that should always be returned from the I/O Exit to File Manager 80
Initialization call discussion 80
Termination call discussion 81
Open call discussion 81
Close call discussion 81
Read and Write call discussion 81
Installing your exit 82

Chapter 10. Customizing File Manager to use a scrambling exit 85
Exit protocol 85
Writing your exit 85
Exit control block description 86
Using the scrambling exit control block 89
Function codes 89
Return codes 89
Return output field values 90
Installing your exit 90

Chapter 11. Verifying the customization of File Manager 93
Verifying the Japanese or Korean components . . . 94

Part 2. Customizing File Manager DB2 Component 97

Chapter 12. Preparing to customize FM/DB2 99
Checklist for installing and customizing FM/DB2 100
Defining the FMN2PARM member 100
Alternatives for making FM/DB2 available . . . 102
Concatenating libraries to the LINKLIST . . . 102
Modifying the TSO logon procedure. 102
Using LIBDEFs to allocate the File Manager Libraries 102
Ensuring the DB2 libraries are available to FM/DB2 103
Planning for running FM/DB2 with APF-authorization 103
Improving FM/DB2 Performance. 103
Deciding how to customize the FM/DB2 Audit facility. 104

Chapter 13. Customizing the operating environment for FM/DB2 107
Modifying the ISPF environment 107
Adding FM/DB2 to the ISPF menu 107
Defining FM/DB2 in an ISPF command table 107
Customizing the FM/DB2 Primary Option Menu 108
Granting access to the DB2 catalog (required) . . 109
Sample jobs to grant SELECT access on the DB2 catalog tables 110
Adding FM/DB2 to the DB2 Administration Launchpad 112
Adding commands to start FM/DB2 functions from the DB2 Administration Tool 113
Starting FM/DB2 functions from an external application 114

Chapter 14. Customizing FM/DB2 . . . 119
Binding DB2 (required) 119
Running multiple versions of FM/DB2 120
Creating a segmented table space (required) . . . 121
Identifying all the DB2 systems that FM/DB2 will access and defining these in the FMN2POPT module (required) 121

Examples of FMN2SSDM macros 122
Customizing the FM/DB2 options 125
Examples of FMN2POPI macros 125
Changing the default options 126
Setting the default national language 127
Changing the JCL skeletons for batch mode . . . 127
Customizing to protect update functions in FM/DB2 129

Chapter 15. Customizing the FM/DB2 audit facility (FMN2POPT controlled auditing) 131
Determining if an audit trail is to be produced . . 131
Using System Management Facilities (SMF) for audit logging 132
Recording an audit trail to a data set 133

Chapter 16. Customizing the FM/DB2 audit facility (SAF-rule controlled auditing) 135
Controlling File Manager auditing using SAF (FM/DB2 component only) 135
When a security product other than RACF is in use 136
Implementing SAF-rule controlled auditing . . 136
Understanding how FM/DB2 uses SAF rules to control auditing 137
How FM/DB2 determines whether audit log records should be written 138
Controlling where FM/DB2 writes audit log records 141
SAF rule examples 142
Controlling where FM/DB2 writes audit log records 142
Controlling auditing of update access to DB2 objects. 143
Controlling auditing of read access to DB2 objects. 144
Testing SAF-rule controlled auditing. 145
FM/DB2 auditing FACILITY and XFACILIT class resource names 146

Chapter 17. Customizing FM/DB2 for national languages 151
Changing the print and display translation tables for languages other than English 151
Translating the message text 152
Providing a national language version of FMN2MENU 152
Translating the ISPF messages text 152
Translating the panel text 153
Using the translated messages and panels 153
Providing a national language version of FMN2DENU 154
Customizing for the Japanese national language 156
Changing the Japanese message text. 156
Customizing for the Korean national language . . 156
Changing the Korean message text 156

Chapter 18. Verifying the customization of FM/DB2 157
Step 1. Define DB2 objects to be used during verification 157
Step 2. Start FM/DB2. 157
Step 3. Edit table format and information 158
Step 4. Display a view 159
Step 5. Change and run SQL 159
Step 6. Drop the new view 160

Part 3. Customizing File Manager IMS Component 161

Chapter 19. Preparing to customize FM/IMS 163
Checklist for installing and customizing FM/IMS 163
Supported databases 163
Region types 164
DLI mode 164
BMP mode 164
PSB types 164
Dynamic PSBs 165
Static PSBs 165
IMS region controller parameters 166
Security 166
IMS subsystem and FM/IMS function security 166
Protected functions 166
The FMN1SXT security exit 167
AGN security 167
RAS security 167
Securing dynamic PSBs in BMP mode 167
IMS subsystem access 167
Read only subsystems 168
BMP mode only and DLI mode only subsystems 168
Static PSB only and dynamic PSB only subsystems 168
Dynamic allocation only subsystems. 168
Releasing locks for unmodified data in Edit and Browse sessions 168
Releasing locks for modified data in Edit sessions 169
Defining the FMN1PARM member 169
Alternatives for making FM/IMS available . . . 171
Concatenating libraries to the LINKLIST . . . 171
Modifying the TSO logon procedure. 171
Deciding how to customize the FM/IMS audit facility. 171
Planning for running FM/IMS with APF-authorization 172

Chapter 20. Customizing the operating environment for FM/IMS 173
Modifying the ISPF environment 173
Adding FM/IMS to your ISPF menu 173
Defining FM/IMS in an ISPF command table 173
Customizing IMS to support the use of dynamic PSBs in BMP mode 174
Declaring the dynamic PSBs 174
Providing a DOPT ACBLIB data set 174

Chapter 21. Customizing FM/IMS . . . 177
Customizing the FM/IMS installation options module . . . 177
FMN0POPI macro . . . 179
FMN1POPD macro . . . 179
FMN1POPI macro . . . 179
FMN1AGNT macro . . . 179
FMN1END macro . . . 180
Specifying the IMS subsystems . . . 180
Specifying the parameters passed to the IMS region controller . . . 180
Controlling access to the subsystems . . . 181
Specifying selected processing options . . . 182
Specifying IMS subsystem and template data sets . . . 183
Specifying miscellaneous IMS subsystem details 183
Specifying AGNS . . . 183
Setting the default national language . . . 184
Examples of FMN1POPD, FMN1POPI and FMN1AGNT macros . . . 185
Tailoring the job control skeletons . . . 189

Chapter 22. Customizing the FM/IMS security environment . . . 191
Controlling access to IMS subsystems and FM/IMS functions . . . 191
Controlling access to the update or read-only functions . . . 192
Controlling access to individual IMS subsystems by the update or read-only functions . . . 193
Controlling access to individual functions . . . 193
Controlling access to individual IMS subsystems by individual functions . . . 194
What governs whether access is granted or denied. . . . 194
Customizing the FM/IMS security exit . . . 196
Types of security exits . . . 196
Invoking the security exit . . . 197
Common Exit Parameters . . . 197
Security Exit Parameters. . . . 198
Sample programs for a security exit . . . 200

Chapter 23. Customizing the FM/IMS audit facility (FMN1POPT controlled auditing) . . . 201
Using System Management Facilities (SMF) for audit logging . . . 201
Recording an audit trail to a data set . . . 202
Activating your changes to SYS1.PARMLIB . . . 203

Chapter 24. Customizing the FM/IMS audit facility (SAF-rule controlled auditing) . . . 205
Implementing SAF-rule controlled auditing . . . 205
Understanding how SAF controls FM/IMS audit logging . . . 206
Controlling where FM/IMS writes audit log records . . . 207
FILEM.AUDIT1.*ssid*.TOSMF . . . 208
FILEM.AUDIT1.*ssid*.TODSN . . . 208
Controlling whether or not an audit trail is created 209
FILEM.AUDIT1.*ssid.fc.db*. . . . 209
Controlling if users can request an audit trail when one is not required (Edit function only). . . . 210
FILEM.AUDIT1.*ssid*.OPTION . . . 210
Specifying that FM/IMS audit logging is to be controlled by SAF . . . 210
When a security product other than RACF is in use . . . 211

Chapter 25. Customizing FM/IMS for national languages . . . 213
Changing the print and display translation tables for languages other than English . . . 213
Translating the message text . . . 214
Providing a national language version of FMN1MENU . . . 214
Translating the ISPF messages text . . . 214
Translating the panel text . . . 215
Using the translated messages and panels . . . 215
Customizing for the Japanese national language 216
Changing the Japanese message text. . . . 216
Customizing for the Korean national language . . 217
Changing the Korean message text . . . 217

Chapter 26. Verifying the customization of FM/IMS . . . 219
Step 1. Build the sample IMS databases to be used during verification. . . . 219
Step 2. Start FM/IMS. . . . 219
Step 3. Display settings menu . . . 220
Step 4. Display DLI mode settings menu . . . 221
Step 5. Verify subsystem details . . . 222
Step 6. Verify DLI mode parameters . . . 222
Step 7. Verify DLI mode data sets . . . 223
Step 8. Verify DLI mode options . . . 224
Step 9. Verify DBD library . . . 224
Step 10. Start the browse dialog . . . 225
Step 11. Verify database data sets . . . 226
Step 12. Database positioning . . . 227
Step 13. Browse the database . . . 228

Part 4. Customizing File Manager CICS Component . . . 229

Chapter 27. Preparing to customize FM/CICS . . . 231
Checklist for installing and customizing FM/CICS 232
Defining the FMN3PARM member . . . 232
Deciding how to customize the FM/CICS audit facility. . . . 234
Publications . . . 235

Chapter 28. Customizing the operating environment for FM/CICS . . . 237
CICS resource definitions for FM/CICS . . . 237
Updating the CICS startup procedures . . . 237

Providing OMVS segments for FM/CICS users . . 237
Customizing for CICS TCP/IP. 238
IPv6 support 238
Accessing other File Manager functions from the primary option menu. 239
Providing CICS resource definitions for FM/CICS on interconnected regions 239
Limiting the regions to which FM/CICS can connect 240

Chapter 29. Customizing FM/CICS . . 241
Customizing the batch procedure. 241
Modifying and submitting FMN3INST 242
Customizing for processing COBOL copybooks . . 245
Customizing for processing PL/I include books 245
Customizing for processing HLASM copybooks 245
Customizing to use DFSORT to improve File Manager performance 245
Changing the default options 246
FMN3POPI 247
Changing the print and display tables 250
Customizing to use an I/O exit 250

Chapter 30. Customizing the FM/CICS security environment 251
Customizing to protect update functions in FM/CICS. 251
Customizing to invoke other File Manager components from FM/CICS 251

Chapter 31. CICS security and FM/CICS 253
Introduction. 253
CICS security considerations for FM/CICS . . . 253
Logon and user security. 253
Resource security 254
Command security 254
Transaction security 254
Intercommunication security 254

Chapter 32. Customizing the FM/CICS audit facility (FMN3POPT controlled auditing) 255
Audit logging under FM/CICS and CICS logging 255
Using System Management Facilities (SMF) for audit logging 255
Recording an audit trail to a data set 256

Chapter 33. Customizing the FM/CICS audit facility (SAF-rule controlled auditing) 259
Controlling FM/CICS auditing using SAF. 259
When a security product other than RACF is in use. 260
Implementing SAF-rule controlled auditing . . 260
Understanding how FM/CICS uses SAF rules to control auditing 261
How FM/CICS determines whether audit log records should be written 262
Controlling where FM/CICS writes audit log records 263
SAF rule examples 264
Controlling where FM/CICS writes audit log records 264
Controlling auditing of FM/CICS functions . . 265
FM/CICS auditing FACILITY and XFACILIT class resource names. 266

Chapter 34. Customizing FM/CICS for national languages 269
Changing the print and display translation tables for languages other than English 269
Setting the LANGUAGE option 269
Verifying that the CICS terminal is DBCS-capable 270
Translating the FM/CICS logon messages 270
Providing a national language version of FMN3MENU 270
Translating the panel text 271
Using the translated messages and panels. . . . 272
Customizing for the Japanese national language 272
Changing the Japanese logon message text . . 272
Customizing for the Korean national language . . 272
Changing the Korean logon message text . . . 272

Chapter 35. Verifying the customization of FM/CICS 275

Chapter 36. Customizing IBM Problem Determination Tools Common Components for use with FM/CICS . . 281
Library names after you finish installing 281
Translating the message text 282
Translating the panel text 283
Using the translated messages and panels. . . 283
Changing the Japanese message text. 283
Changing the Korean message text 284

Chapter 37. IBM File Manager plug-in for Eclipse 285
Installation requirements 285
Host requirements. 285
Client requirements 285
Host installation and configuration 286
Client installation 286
Eclipse P2 installation for CICS Explorer users 286
Installation Manager installation for Rational Developer for System z users 287
Problem Diagnostics 288

Appendix A. File Manager options . . 289
ABENDCC 289
ASCII 290
AUDDATAC 290
AUDITHLQ 291
AUDITLOG 292
AUDMGMTC 293
AUDPQTY 293
AUDSQTY 293

AUDSTORC 293
AUDSUNIT 294
AUDUNIT 294
AUXDATAC. 294
AUXDSN. 295
AUXHLQ 295
AUXMDSN 296
AUXMGMTC 296
AUXSTORC 297
EDITCAPS 297
COBDBCS 298
COBDPC 298
COBEXTND. 299
COBMAXRTN 299
COBMCASE. 299
COMPLANG 300
CREPLACE*n* 300
CSYSLIB*nn* 300
CYLHD 301
DATAHDR 301
DSPINC 301
DSPMAX. 302
DSPMIN 302
DSPNUM 302
DUMP 303
EDMAXVIRT 303
EOD 303
FMEDITOR 304
HEADERPG. 304
HLDBCS 304
HLMAXRTN 305
HLNOALIGN 305
HSYSLIB*nn* 305
ISPFPACK 306
LANGUAGE 306
LMS 307
LMSUBSYS 308
MQREPHLQ 308
MSGUPPER 309
PAD 310
PAGESIZE 310
PAGESKIP 310
PDATAC 311
PLI31DIGIT 311
PLI63BIT 311
PLIGRAPHIC 312
PLIMAXRTN 312
PLIUNALIGN 312
PMGMTC 313
PRINTDSN 313
PRINTLEN 313
PRINTOUT 314
PRTCLASS 314
PRTDISP 315
PRTMGMTC 315
PRTPQTY 315
PRTSQTY 316
PRTSTORC 316
PRTSUNIT 316
PRTTRANS 317
PRTUNIT. 317
PSTORC 317
PSYSLIB*nn* 318
PTRDATAC 318
PUNIT 318
RECLIMIT 318
SEC 319
SFMNMOD1 319
SMFNO 320
TAPELBL. 320
TDATAC 321
TEMPHLQ 321
TERMTYPE 322
TMGMTC 323
TRACECLS 323
TRACEOUT 323
TRCDATAC 324
TRCMGMTC 324
TRCPQTY 324
TRCSQTY 325
TRCSTORC 325
TRCSUNIT 325
TRCUNIT 326
TSTORC 326
TUNIT 326
USEIOX 326
VSAUTO 327
VSCHGAUTO 327
VSCHGFRQ 328
VSSAVE 328
WIDEPRT 329

Appendix B. FM/DB2 options. 331
FMN2SSDM 331
ATTACH 331
AUDIT 332
AUDITBROWSE 333
CCSIDWARNIGNORE 334
CPYCPYN 334
DB2CLIB 334
DB2ELIB 335
DB2LLIB 335
DB2MLIB. 336
DB2PLIB 336
DB2PROC 337
DB2RLIB 337
DB2SLIB 338
DB2TLIB 338
DESC 338
DISPLAY 339
LDFDDLN 339
LOCATION 340
LOCATION_NICKNAME 340
LODINDN 341
OPTEVT1 341
OPTEVT2 341
OPTEVT3 342
OPTEVT4 342
PLAN 342
PLAN2 342
RBXWRKN 343
ROGUNLN 343

SLDJCL1 343
SLDJCL2 344
SLDJCL3 344
SLDJCL4 344
SSID 345
STMJCL1 345
STMJCL2 345
STMJCL3 346
STMJCL4 346
TABLE_LOCKING. 346
TMPDDLN 347
TYPE 347
UNLPUNN 348
UNLUNLN 348
FMN2POPI 348
CATOWNERCDRM 348
CATOWNER 350
CCSIDWARN 351
CONNECT 352
EDITCAPS 353
OP34MOD 354
LIST 355
SHOWDATAC 357
SHOWMGMTC 357
SHOWPQTY 357
SHOWSQTY. 358
SHOWSTORC 358
SHOWSUNIT 358
SHOWUNIT. 359
SSIDCMD1 359
SSIDCMD2 359

Appendix C. FM/IMS options 361
FMN1POPD and FMN1POPI macros 361
ACBLIB 362
AUTOSAVE 362
CHGAFREQ. 363
COMPAT 363
DBDLIB*n* 364
DBRC 365
DESC 365
DFSRRC00 366
DFSVSAMP 366
DYNACB. 367
DYNALLOC 367
DYNPRFN 368
DYNPRFX 368
EDITFREQ 369
GSGNAME 370
IEBFREQ 370
IMSAUDLG 371
IMSBKO 371
IMSNBA 372
IMSOBA 372
IRLM 373
IRLMNAME. 373
LKEYDATAC 374
LKEYMGMTC 374
LKEYPQTY 374
LKEYSQTY 375
LKEYSTORC 375
LKEYSUNIT. 375
LKEYVOL*n* 376
LOADFREQ 376
LOCKMAX 377
LOGDATAC. 377
LOGMGMTC 378
LOGPQTY 378
LOGSQTY 378
LOGSTORC 378
LOGSUNIT 379
LOGUNIT 379
MACLIB 379
MAXGN 380
PARDLI 380
PROCOPTB 381
PROCOPTP 382
PROCOPTX 382
PROCOPTY 383
PSBLIB*n* 383
PSBTYPE 384
PSBTYPES 385
READONLY. 385
REGTYPES 386
RESLIB*n* 386
RKEYDATAC 387
RKEYMGMTC 387
RKEYPQTY 388
RKEYSQTY 388
RKEYSTORC 388
RKEYSUNIT 388
RKEYVOL*n* 389
SFMNSLIB 389
SSID 389
TMINAME 390
TPLLIB*n* 390
UACBLIB 391
UAGNS 391
UAUTOSAV. 392
UBUF 392
UDBDLIB 393
UDBRC 393
UDFSVSMP 394
UIEBFREQ 394
UIMSBKO 395
UIMSNBA 396
UIRLM 396
ULOADFRQ. 397
ULOCKMAX 398
UMACLIB 399
UPARDLI 399
UPROCOPB 400
UPROCOPP 400
UPROCOPX 401
UPROCOPY 402
UPSBLIB 403
UPSBTYPE 403
URECON. 404
URESLIB 404
URSR 405
UTPLLIB 405
VCURULE 406

VSMPMEM 406
XDOPTLB 407
XKEYDATAC 407
XKEYMGMTC 408
XKEYPQTY 408
XKEYSQTY 408
XKEYSTORC 408
XKEYSUNIT 409
XKEYUNIT 409
FMN1AGNT macro 409
AGN 409
DESC 410
SSID 410

Appendix D. File Manager options specified in PARMLIB members . . . 411
File Manager options specified in FMN0PARM . . 411
FMAUDIT 411
FMOPTMOD 412
Facilities for customizing the FMN0PARM definitions 413
Tags 413
Rules for specifying tags 413
Example of tag usage. 414
Included members. 415
Comments 416
Continuing specifications across multiple lines 416
FM/IMS options specified in FMN1PARM . . . 417
FMAUDIT 417
FMOPTMOD 418
Facilities for customizing the FMN1PARM definitions 419
Tags 419
Rules for specifying tags 419
Example of tag usage. 419
Included members. 421
Comments 422
Continuing specifications across multiple lines 422
FM/DB2 options specified in FMN2PARM . . . 422
FMAUDIT 423
FMOPTMOD 424
Facilities for customizing the FMN2PARM definitions 424
Tags 425
Rules for specifying tags 425
Example of tag usage. 425
Included members. 427
Comments 428
Continuing specifications across multiple lines 428
FM/CICS options specified in FMN3PARM . . . 428
FMAUDIT 429
FMOPTMOD 429
Facilities for customizing the FMN3PARM definitions 430
Tags 430
Rules for specifying tags 430
Example of tag usage. 431
Included members. 432
Comments 433
Continuing specifications across multiple lines 433

Appendix E. Customizing the FM Server 435
FM Server options. 435
Syntax of the SERVER statement 435
Modifying FMNCAS 436
Additional security considerations 437
Address space timeout 438

Appendix F. The library management system exit 439
Coding the exit: the basics 439
The main program and first argument: the operation code 439
The return code 440
Initializing the exit 441
Tracing 441
Terminating the exit 442
Coding the exit: required services 442
Common argument and the name of the library 442
Validate the library 443
Get member records 443
Get member information (metadata). 445
Get display information 446
Supporting multiple library management systems 447
Writing the exit in HLASM. 447

Appendix G. File Manager audit records 449
File Manager (base) audit records 452
File Manager (base) audit data items 452
File Manager (base) audit record types 452
Data items specific to File Manager (base) audit records 453
FM/DB2 audit records 453
FM/DB2 audit data items 453
FM/DB2 audit record types 454
Data items specific to FM/DB2 audit records 455
FM/IMS audit records 455
FM/IMS audit data items 456
FM/IMS audit record types 457
Data items specific to FM/IMS audit records 457
FM/CICS audit records 458
FM/CICS audit data items 458
FM/CICS audit record types 458
Data items specific to FM/CICS audit records 459

Appendix H. Support resources and problem solving information 461
Searching IBM support Web sites for a solution . . 461
Searching the information center 461
Searching product support documents 461
IBM Support Assistant 462
Obtaining fixes 462
Receiving support updates through e-mail notification 463
Receiving support updates through RSS feeds . . 464
If you need to contact IBM Software Support. . . 464
Determining the business impact 465
Describing problems and gathering information 465

Submitting problems 466

Notices 469
Programming interface information 470
Trademarks 470

Bibliography 471
File Manager publications 471
Related publications for DB2 471
Related publications for CICS 471
Related publications for Communications Server 471
Related publications for z/OS 471
Softcopy publications 471

Index 473

About this document

This book provides information needed to plan for, customize, maintain, and diagnose problems with IBM File Manager for z/OS (File Manager), IBM File Manager for z/OS DB2 component (FM/DB2), IBM File Manager for z/OS IMS component (FM/IMS), and IBM File Manager for z/OS CICS component (FM/CICS).

File Manager Version 11 comprises four components, File Manager base function, FM/DB2, FM/IMS, and FM/CICS. In order to install and use FM/DB2, FM/IMS, or FM/CICS, you **must** first install File Manager Version 11 base function. You cannot install FM/DB2, FM/IMS, or FM/CICS, until you have installed File Manager Version 11 base function, and you cannot use FM/DB2 Version 11, FM/IMS Version 11 or FM/CICS Version 11 with any other version of File Manager. You can install one, two or all three of FM/DB2, FM/IMS and FM/CICS in any order, at any time after you have installed File Manager Version 11 base function.

This book is divided into five sections. Part 1, "Customizing File Manager," on page 1 provides information about the base function of File Manager: Part 2, "Customizing File Manager DB2 Component," on page 97 provides information about the DB2® component of File Manager: Part 3, "Customizing File Manager IMS Component," on page 161 provides information about the IMS™ component of File Manager; Part 4, "Customizing File Manager CICS Component," on page 229 provides information about the CICS® component of File Manager. The appendixes provide information about the File Manager options, the library management system exit, and how to maintain File Manager.

In addition to information about the File Manager base function, FM/DB2, FM/IMS, and FM/CICS, this book also provides information needed to customize the Japanese and Korean language versions of File Manager and the DB2, IMS, and CICS components.

Unless otherwise stated, the information provided in this book refers to the English versions of File Manager, FM/DB2, FM/IMS, and FM/CICS.

Before you read this book, read the *File Manager Program Directory*.

Overview of File Manager

File Manager is an IBM® licensed program that helps you work with various storage media and maintain data.

File Manager provides production and development logical file manipulation for Websphere MQ queue data, HFS files, and QSAM and VSAM data sets. VSAM data sets include ESDS, KSDS, RRDS, and VRRDS. File Manager provides both applications and systems programmers with the capability to access, view, edit, find, modify, copy, maintain, repair, migrate, print and manage data and files. The Template Workbench provided in File Manager allows users (particularly applications programmers) to format data stored in any of the above files based upon the definitions in COBOL, PL/I or High Level Assembler copybooks, or to define data formats dynamically. Adding to this capability is an interactive interface that allows programmers to view data in both table and single-record

© Copyright IBM Corp. 2000, 2011

formats. In addition, selection of data can be performed using simple REXX conditions that can easily be tailored by the programmer. All of the features available on-line to manipulate data can also be applied in batch jobs.

The File Manager DB2 component extends this functionality to the DB2 environment. FM/DB2:
- Extends the File Manager editor facilities to apply to DB2 data
- Extends the ISPF object list utility to apply to DB2 data
- Extends the File Manager data create facility to DB2 data
- Extends the File Manager data copy facility to include DB2 data as both source and target
- Extends the File Manager template processing to the DB2 environment
- Provides an SQL analysis feature for both existing plans and imbedded source code
- Provides an easy interface to SQL statement generation for those who do not know SQL
- Provides an interface to generate JCL and DB2 control statements for selected DB2 utilities, without the user needing to know the syntax for these utilities

The File Manager IMS component further extends this functionality to provide support for IMS databases. FM/IMS provides database edit and browse functions such as:
- Ability to view and edit data in IMS databases with full-screen ISPF panels
- Display either multiple segments or a single segment on the one screen
- Formatted segment display (in context of COBOL or PL/I data structure)
- Formatted tabular display of data for a single segment type
- Selectable segment display (choose which segments to display based on segment keys or segment contents, or both)
- Selectable field display (choose which fields to display, and in which order)
- Unformatted segment display
- Ability to insert, delete, and repeat segments and their children
- Ability to find and change data across an entire database hierarchy
- Optional audit trail of all database updates

FM/IMS allows the user to define templates and views which associate COBOL and PL/I data structures with each segment type in a DBD; and define views which use the COBOL or PL/I data structures to specify the selection criteria for segment types.

FM/IMS provides support for either static (existing) or dynamic PSBs, the option to access databases via either DLI or BMP processing, and support for HALDB, HDAM, HIDAM, HISAM, HSAM, DEDB, MSDB, and logical databases, including databases with secondary indexes.

The File Manager CICS component provides a powerful set of utility functions for editing, browsing, printing, and altering the status of CICS resources. The CICS resources supported are files, temporary storage queues, and transient data queues. With appropriate authority the user can also modify the status of the CICS resources. FM/CICS incorporates much of the functionality of File Manager into the CICS environment.

FM/CICS uses IBM Problem Determination Tools Common Components to display panels on CICS with a similar capability to ISPF panels. FM/CICS panels allow you to select options and to specify parameters, commands and program function (PF) keys to simplify requests for common functions. FM/CICS panels provide full screen format for information display and editing.

File Manager is an ISPF application, and includes support for screen sizes larger than 24x80. You can use File Manager's functions in full-screen mode. For routine tasks, you can use File Manager in batch jobs using control statements.

File Manager uses 31-bit addressing. It is enabled for National Language Support and can be translated if required. From version 8, File Manager provides Service Oriented Architecture (SOA) support, allowing the generation of XML data from files.

Who might use this document

This book is for system programmers and system administrators who plan for, customize, and maintain File Manager.

It is also relevant to users who carry out diagnostic tasks on this product.

To use this book, you need to be familiar with the z/OS operating system, the publications that describe your system, and job control language (JCL) or exec processing.

Experienced installers

If you are experienced in installing products with SMP/E, refer to the following fast path sections in the *File Manager Program Directory*:

1. "Installation Requirements and Considerations"; see sub-section "DASD Storage Requirements"
2. "Installation Instructions"; follow each instruction as described in the *Program Directory*.

Other documentation that you might need

For the installation of File Manager, and the DB2, IMS, and CICS components, or for the installation of File Manager Japanese or Korean components, you will need to refer to the *File Manager Program Directory*.

You may also need to refer to the *File Manager User's Guide and Reference*, the *File Manager User's Guide and Reference for DB2 Data*, the *File Manager User's Guide and Reference for IMS Data*, or the *File Manager User's Guide and Reference for CICS*.

File Manager publications, related product publications, and their order numbers, are listed in the "Bibliography" on page 471.

Document conventions

This book is divided into five sections. Part 1, "Customizing File Manager," on page 1 provides information about the File Manager base function; Part 2, "Customizing File Manager DB2 Component," on page 97 provides information about File Manager DB2 component; Part 3, "Customizing File Manager IMS Component," on page 161 provides information about File Manager IMS component; and Part 4, "Customizing File Manager CICS Component," on page 229 provides information about File Manager CICS component. The appendixes provide information about the File Manager options, the library management system exit, and how to maintain File Manager.

Part 1, Part 2, Part 3, and Part 4 are further divided into chapters describing the preparation for customizing File Manager and its components, and for customizing

File Manager, its components and the operating environment. The last chapter in each section describes how to verify your installation and customization.

You will need to read Part 1, “Customizing File Manager,” on page 1.

If you intend to install FM/DB2, you will need to read Part 2, “Customizing File Manager DB2 Component,” on page 97.

If you intend to install FM/IMS, you will need to read Part 3, “Customizing File Manager IMS Component,” on page 161.

If you intend to install FM/CICS, you will need to read Part 4, “Customizing File Manager CICS Component,” on page 229.

If you intend to use FM/CICS with the File Manager Server, or intend to use the File Manager Plugin for Eclipse, you will need to read Appendix E, “Customizing the FM Server,” on page 435.

You do not need to read these sections if you are not planning to install these components.

The following section describes the syntax notation used in this book.

Syntax notation

Throughout this book, syntax descriptions use the structure defined below.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

  The >>— symbol indicates the beginning of a statement.

  The —-> symbol indicates that the statement syntax is continued on the next line.

  The >— symbol indicates that a statement is continued from the previous line.

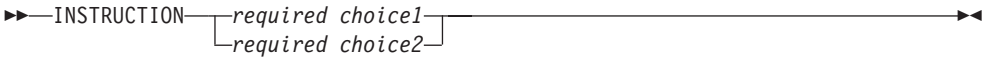
  The —>< indicates the end of a statement.

  Diagrams of syntactical units other than complete statements start with the >— symbol and end with the —-> symbol.
- **Keywords** appear in uppercase letters (for example, ASPACE) or upper and lower case (for example, PATHFile). They must be spelled exactly as shown. Lower case letters are optional (for example, you could enter the PATHFile keyword as PATHF, PATHFI, PATHFIL or PATHFILE).

  Variables appear in all lowercase letters in a special typeface (for example, *integer*). They represent user-supplied names or values.
- If punctuation marks, parentheses, or such symbols are shown, they must be entered as part of the syntax.
- Required items appear on the horizontal line (the main path).

- Optional items appear below the main path. If the item is optional and is the default, the item appears above the main path.

- When you can choose from two or more items, they appear vertically in a stack.
  If you **must** choose one of the items, one item of the stack appears on the main path.



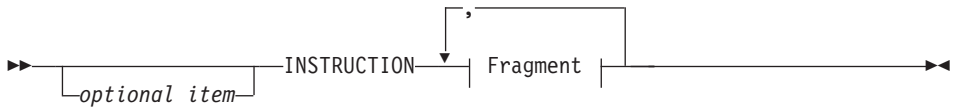
  If choosing one of the items is optional, the whole stack appears below the main path.

- An arrow returning to the left above the main line indicates an item that can be repeated. When the repeat arrow contains a separator character, such as a comma, you must separate items with the separator character.

  A repeat arrow above a stack indicates that you can make more than one choice from the stacked items, or repeat a single choice.

The following example shows how the syntax is used.

Syntax diagram



Fragment:

Notes:

1 *operand choice2* and *operand choice3* must not be specified together

A The item is optional, and can be coded or not.

B The INSTRUCTION key word must be specified and coded as shown.

C The item referred to by "Fragment" is a required operand. Allowable choices for this operand are given in the fragment of the syntax diagram shown below "Fragment" at the bottom of the diagram. The operand can

also be repeated. That is, more than one choice can be specified, with each choice separated by a comma. The note at the bottom of the syntax diagram indicates a restriction on the choice.

How to send your comments

Your feedback is important to us in helping us to provide accurate, high-quality information. If you have comments about this document or any other File Manager documentation, contact us in one of these ways:

- Use the Online Readers' Comment Form at
  `www.ibm.com/software/awdtools/rcf/`

  Be sure to include the name of the document, the publication number of the document, the version of File Manager and, if applicable, the specific location (for example, page number) of the text that you are commenting on.
- Fill out the Readers' Comment Form at the back of this document, and return it by mail or give it to an IBM representative. If the form has been removed, address your comments to:

  IBM Corporation
  Reader Comments
  DTX/E269
  555 Bailey Avenue
  San Jose, CA, 95141-9989
  U.S.A.
- E-Mail your comments to this address: comments@us.ibm.com

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

1. Send an email to comments@us.ibm.com
2. Use the form on the Web at:
   www.ibm.com/software/ad/rcf/
3. Mail the comments to the following address:

   IBM Corporation
   H150/090
   555 Bailey Avenue
   San Jose, CA
   95141-1003
   U.S.A.

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:
  FM V11R1 Customization Guide
  SC19-3133-01
- The topic and page number related to your comment
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

If you have a technical problem

Do not use the feedback methods listed above. Instead, do one of the following:

- Contact your IBM service representative
- Call IBM technical support
- Visit the IBM support portal at http://www.ibm.com/systems/z/support/

© Copyright IBM Corp. 2000, 2011

Summary of changes

Any technical changes to this document since the last edition are indicated by a vertical line to the left of the change.

© Copyright IBM Corp. 2000, 2011

Part 1. Customizing File Manager

© Copyright IBM Corp. 2000, 2011

Chapter 1. Preparing to customize File Manager

Before you can use File Manager, you will need to customize File Manager and the operating environment. Table 1 lists the customization tasks you can perform for File Manager. Read the referenced sections to see if you need to perform the customization described.

Checklist for installing and customizing File Manager

Table 1. Summary of steps for customizing File Manager and the operating environment

	Description	Page
__ 1	Add FMN0PARM to SYS1.PARMLIB.	4
__ 2	Concatenate File Manager libraries to the LINKLIST.	6
__ 3	Modify the TSO logon procedure.	7
__ 4	Add File Manager to the ISPF menu.	11
__ 5	Define File Manager in an ISPF command table.	11
__ 6	Make File Manager the default VSAM editor.	12
__ 7	Enable WebSphere MQ support.	13
__ 8	Customize to use COBOL copybooks.	13
__ 9	Customize to use PL/I include books.	17
__ 10	Customize to use HLASM copybooks.	17
__ 11	Customize for DFSORT to improve performance.	17
__ 12	Bind DB2 for OAM functions.	20
__ 13	Authorize File Manager.	7, 21
__ 14	Change the default options.	23
__ 15	Change the batch JCL skeleton.	24
__ 16	Customize the batch return codes.	26
__ 17	Provide a user I/O exit.	73
__ 18	Change the print and display translation tables.	28
__ 19	Change the ASCII translation tables.	29
__ 20	Customize the File Manager security environment.	31
__ 21	Determine how to customize the File Manager audit facility.	8
__ 21	Customize File Manager for national languages.	61
__ 22	Customize File Manager to use library management system libraries.	67

Library names after you finish installing

Throughout this book it is assumed that you have installed File Manager into the default libraries. The default high level qualifier is FMN. Therefore after you have installed File Manager the names of the target and distribution libraries will be those listed in Table 2 on page 4. If you have installed a Japanese component, the Japanese target and distribution libraries will be those listed in Table 3 on page 4.

© Copyright IBM Corp. 2000, 2011

If you have installed a Korean component, the Korean target and distribution libraries will be those listed in Table 4.

Table 2. File Manager target and distribution libraries

Target library name	Distribution library name	Usage
FMN.SFMNMOD1	FMN.AFMNMOD1	File Manager load modules
FMN.SFMNMOD2	FMN.AFMNMOD2	FM/CICS-specific load modules
FMN.SFMNMODA		FM/CICS-specific and RDz-specific load modules that must be APF-authorized
FMN.SFMNSAM1	FMN.AFMNSAM1	Sample jobs and source code
FMN.SFMNPENU	FMN.AFMNPENU	File Manager English panels
FMN.SFMNMENU	FMN.AFMNMENU	File Manager English messages
FMN.SFMNSLIB	FMN.AFMNSLIB	File Manager skeletons
FMN.SFMNTENU	FMN.AFMNTENU	File Manager English tables
FMN.SFMNMAC1	FMN.AFMNMAC1	File Manager macros
FMN.SFMNEXEC	FMN.AFMNEXEC	File Manager execs
FMN.SFMNCLIB	FMN.AFMNCLIB	File Manager clists
FMN.SFMNDBRM	FMN.AFMNDBRM	File Manager DBRMs

Table 3. File Manager Japanese target and distribution libraries

Target library name	Distribution library name	Usage
FMN.SFMNMODJ	FMN.AFMNMODJ	File Manager Japanese BMS maps
FMN.SFMNPJPN	FMN.AFMNPJPN	File Manager Japanese panels
FMN.SFMNMJPN	FMN.AFMNMJPN	File Manager Japanese messages
FMN.SFMNTJPN	FMN.AFMNTJPN	File Manager Japanese tables

Table 4. File Manager Korean target and distribution libraries

Target library name	Distribution library name	Usage
FMN.SFMNMODK	FMN.AFMNMODK	File Manager Korean BMS maps
FMN.SFMNPKOR	FMN.AFMNPKOR	File Manager Korean panels
FMN.SFMNMKOR	FMN.AFMNMKOR	File Manager Korean messages
FMN.SFMNTKOR	FMN.AFMNTKOR	File Manager Korean tables

Defining the FMN0PARM member

The File Manager Version 11 component requires an FMN0PARM member in SYS1.PARMLIB (or any other library in the logical parmlib concatenation). This is a new requirement for File Manager Version 11. Previous versions of File Manager did not require a parmlib member.

If auditing is to be controlled from parmlib (user has read access to profile FILEM.PARMLIB.BASE, see "Controlling File Manager auditing using SAF" on page 51

page 51), then member FMN0PARM must be defined in SYS1.PARMLIB (or any other library in the logical parmlib concatenation) as follows.

Default parmlib member FMN0PARM is provided in the SFMNSAM1 library. Copy this member to the appropriate system parmlib library. See below for details of methods that can be used to make this change.

Note: The sample FMN0PARM member supplied in SFMNSAM1 also includes an FMSECRTY statement. This option is not used in File Manager Version 11, and can be either omitted, or commented out. It has no effect.

File Manager will not start if the required parmlib member does not exist in the system logical parmlib concatenation for users to be audited using parmlib.

There are two methods that can be used to include the FMN0PARM member in a library in the logical parmlib concatenation. The choice of method depends on whether the installation's security software is configured to allow File Manager users READ access to the data set SYS1.PARMLIB. Method 1 can only be used when File Manager users have read access to SYS1.PARMLIB. Method 2 can be used regardless of whether File Manager users have READ access to SYS1.PARMLIB or not, and must be used when File Manager users do not have READ access to SYS1.PARMLIB.

Method 1

Place the FMN0PARM member in any library in the current logical parmlib concatenation. No IPL or other action is required to activate the new member (unless a new library was added to the logical parmlib concatenation).

Notes:

1. Method 1 cannot be used in any situation where File Manager users do not have READ access to SYS1.PARMLIB. For example, when File Manager users have READ access to another library in the logical parmlib concatenation, and the FMN0PARM member is placed in the latter library. This will not work. The key issue is whether the File Manager user has READ access to SYS1.PARMLIB.
2. Using this method results in message IEE252I being written to the system log whenever a File Manager user accesses SYS1.PARMLIB. These messages cannot be suppressed. To avoid these messages use Method 2.

Method 2

This method must be used when File Manager users do not have READ access to SYS1.PARMLIB, or when suppression of the IEE252I messages is required.

1. Create a new library with dataset attributes similar to SYS1.PARMLIB.
2. The library name for this data set must include the string "FMNPARM" in one of the qualifiers. You can choose any data set name that meets this requirement. Examples of suitable data set names are:
   SYS1.PARMLIB.FMNPARM SYS8.FMNPARM.PARMLIB
   FMNPARM.SYS8.PARMLIB SYS2.FMNPARMS.LIB
   SYS8.XFMNPARM.PARMLIB
3. Add member FMN0PARM to the new library, specifying the appropriate FMAUDIT parameter.
4. Add the new library to the logical parmlib concatenation. This can be done dynamically, or by means of a system IPL.

Note: When Method 2 is used, the FMN0PARM member must be located in the library created in Step 1 (as described above). In addition, if the FMN0PARM member specifies any include statements (see "Facilities for customizing the FMN0PARM definitions" on page 413), all the included members must also reside in the library created in Step 1.

You use the FMN0PARM member to define:

- Whether File Manager uses SAF to control File Manager audit logging.
- The SAF resource name prefix to be used by File Manager when determining access to various resources.
- Whether File Manager loads the FMN0POPT module from a specific library.

For more information, see "Deciding how to customize the File Manager audit facility" on page 8.

Alternatives for making File Manager available

You can make File Manager available to your users in one of four ways:

- By concatenating FMN.SFMNMOD1 to your LINKLIST.
- By adding FMN.SFMNMOD1 to the STEPLIB DD statement in your TSO logon procedure.
- By activating the FMN.SFMNMOD1 data set using the TSO command, TSOLIB.
- By adding FMN.SFMNMOD1 to the ISPLLIB DD concatenation.

Note: You cannot use the ISPF command LIBDEF ISPLLIB to allocate the FMN.SFMNMOD1 library, and then successfully invoke File Manager. File Manager uses LOAD instructions, whereas LIBDEF ISPLLIB is only valid for programs invoked by the ISPF SELECT PGM service. Please note that this restriction only applies to File Manager base and IMS features.

This restriction does NOT apply to the File Manager DB2 feature. LIBDEF ISPLLIB is used in the File Manager DB2 sample FMN2INIT.

To make File Manager readily available from ISPF, configure your ISPF environment as described in Chapter 2, "Customizing the operating environment for File Manager," on page 11.

Concatenating libraries to the LINKLIST

To make File Manager commonly available, add the FMN.SFMNMOD1 library to your concatenated LINKLIST.

If you plan to use functions using PL/I include books, add the Language Environment runtime library, SCEERUN, to your concatenated LINKLIST, in front of any other PL/I runtime library.

If you plan to use functions using High Level Assembler (HLASM) copybooks, ensure that the HLASM load library, ASM.SASMMOD1, is available to File Manager. You can do this by adding this library to your concatenated LINKLIST.

If you plan to use File Manager to access COBOL copybooks or PL/I include books stored in library management system libraries, add the Language Environment runtime library, SCEERUN, to your concatenated LINKLIST. For

information about using library management system libraries, see Chapter 8, “Customizing File Manager to use library management system libraries,” on page 67.

To concatenate these libraries to the LINKLIST, add them to either your LNKLST*xx* or PROG*xx* member in SYS1.PARMLIB.

Modifying the TSO logon procedure

If you did not add FMN.SFMNMOD1 to your LINKLIST, as described above, you can add this library to the STEPLIB DD statement or the ISPLLIB DD statement in your TSO logon procedure.

You also need to add the following File Manager libraries to your TSO logon procedure.

```
DDNAME ISPMLIB:  add library FMN.SFMNMENU
DDNAME ISPPLIB:  add library FMN.SFMNPENU
DDNAME ISPSLIB:  add library FMN.SFMNSLIB
DDNAME ISPTLIB:  add library FMN.SFMNTENU
DDNAME SYSEXEC:  add library FMN.SFMNEXEC
DDNAME SYSPROC:  add library FMN.SFMNCLIB
```

If you are installing the Japanese component, add the following File Manager libraries to your TSO logon procedure.

```
DDNAME ISPMLIB:  add library FMN.SFMNMJPN in front of FMN.SFMNMENU
DDNAME ISPPLIB:  add library FMN.SFMNPJPN in front of FMN.SFMNPENU
DDNAME ISPTLIB:  add library FMN.SFMNTJPN in front of FMN.SFMNTENU
```

If you are installing the Korean component, add the following File Manager libraries to your TSO logon procedure.

```
DDNAME ISPMLIB:  add library FMN.SFMNMKOR in front of FMN.SFMNMENU
DDNAME ISPPLIB:  add library FMN.SFMNPKOR in front of FMN.SFMNPENU
DDNAME ISPTLIB:  add library FMN.SFMNTKOR in front of FMN.SFMNTENU
```

If you are using the ISPF alternate libraries, you must also add the following File Manager libraries into your concatenation.

```
DDNAME: ISPSALT: add library FMN.SFMNSLIB
DDNAME: ISPMALT: add library FMN.SFMNMJPN or FMN.SFMNMKOR
DDNAME: ISPPALT: add library FMN.SFMNPJPN or FMN.SFMNPKOR
```

Planning for running File Manager with or without APF-authorization

File Manager can be run in batch mode with or without APF-authorization. File Manager cannot run APF-authorized under ISPF.

There are a number of reasons why you might want to run File Manager APF-authorized in batch mode:

- Users can be permitted to use disk fullpack processing. For more information, see “Controlling fullpack access to DASD volumes” on page 35.
- Users can be permitted to use bypass label processing (BLP), even if the system does not support BLP. For more information, see “Controlling Bypass Label Processing (BLP)” on page 37.
- Users can use compaction (IDRC) for 3480 or 3490 cartridge tapes for tape mode 08 (buffered write mode) and tape mode 28 (immediate write mode).
- Users can perform catalog actions that require APF-authorization, as described in *z/OS DFSMS Access Method Services for Catalogs*.

Planning for running File Manager with or without APF-authorization

If you plan to use SMF to record audit trail information for File Manager base function, or for the DB2 or IMS components, you must make File Manager (or at least the module FMNSMF) APF-authorized. See "Using System Management Facilities (SMF) for audit logging" on page 49 for more information.

If you plan to run File Manager APF-authorized in batch mode, and you want to run File Manager batch jobs using COBOL copybook templates, then you **MUST** also authorize the COBOL compiler library.

If you are accessing data in an LMS without a SUBSYS interface, or you are using FMNCRAEX to access data in an LMS, then you cannot run File Manager APF-authorized.

If you plan to use the FM Server for FM/CICS or for the File Manager Plugin for Eclipse access, you must make the target library, SFMNMODA, APF-authorized.

Deciding how to customize the File Manager audit facility

You have two choices with respect to auditing of File Manager audit activities:

FMN0POPT controlled auditing
: The facilities available with FMN0POPT controlled auditing are that you can specify auditing to the user's audit log data set, to the user's audit log data set with automatic (mandatory) printing of the audit log at the completion of the session, or to SMF.

SAF-rule controlled auditing
: This relies on various SAF FACILITY and XFACILIT resource rules which you define with an external security product, such as RACF (or equivalent product).

These points summarize the facilities available with SAF-rule controlled auditing:

- Auditing can be (optionally) specified for all File Manager functions
- Different auditing requirements can be specified for different TSO userids.
- Different auditing requirements can be specified for access to different resources.
- You can provide File Manager users with a "Create audit trail" option for the File Manager edit functions. This is also SAF-rule controlled. The presence of the "Create audit trail" option does not guarantee that the user can switch off auditing, since this depends on the level of access the user has to the appropriate SAF resource names. When a user has access to the "Create audit trail" option, they can always turn on auditing, even if the relevant SAF resource rules do not require auditing.
- You can specify auditing to the user's audit log data set, to the user's audit log data set with automatic (mandatory) printing of the audit log at the completion of the session, or to SMF. Dual logging (to the user's audit log data set and to SMF) can also be specified.

Some other points to consider are:

- Auditing to the user's audit log data set can result in large numbers of audit log data sets. This may have disk space implications. You may need to consider implementing automatic purging or archiving of audit log data sets.
- Auditing to SMF (only) requires additional set-up, but provides a more reliable and secure environment for capturing audit information than audit logging to the user's audit log data set.

If you determine that SAF-rule controlled auditing is not appropriate, follow the instructions in Chapter 5, “Customizing the File Manager audit facility (FMN0POPT controlled auditing),” on page 49.

If you determine that SAF-rule controlled auditing is appropriate, follow the instructions in Chapter 6, “Customizing the File Manager audit facility (SAF-rule controlled auditing),” on page 51.

The auditing method you choose will also apply to your RDz users if you are using the File Manager server.

Chapter 2. Customizing the operating environment for File Manager

This chapter describes how to customize the operating environment for File Manager. You do this after you have installed File Manager.

Modifying the ISPF environment

To make it easy to start File Manager under ISPF, configure your ISPF environment as described in the following sections.

Note: When using File Manager from ISPF, it is recommended you use at least a 4MB logon region size (in conjunction with the 32M of "extended" storage that TSO generally provides) to provide enough 24-bit storage for various functions to operate effectively. However, for relatively intensive virtual storage functions, such as the usage of templates with many fields or layouts or "memory" style editing, it may be necessary to increase the region size beyond 40M.

Adding File Manager to the ISPF menu

To add File Manager to your ISPF Primary Option Menu panel (ISR@PRIM), insert the additional lines (◄ **New**) as shown in Figure 1.

```
   ⋮
)BODY  CMD(ZCMD)
   ⋮
 9  IBM Products  IBM program development products
 10 SCLM          SW Configuration Library Manager
 11 Workplace     ISPF Object/Action Workplace
 F  File Manager  File Manager for z/OS ◄    New
   ⋮
)PROC
   ⋮
 &ZSEL = TRANS( TRUNC (&ZCMD,'.')
   ⋮
  9,'PANEL(ISRDIIS) ADDPOP'
 10,'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
 11,'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
  F,'PANEL(FMNSTASK) SCRNAME(FILEMGR) NEWAPPL(FMN)' /* File Manager */ ◄ New
   ⋮
```

Figure 1. Adding File Manager to the ISPF Primary Option Menu panel

For more information about configuring your ISPF Primary Option Menu panel, see *z/OS ISPF Planning and Customizing*.

Defining File Manager in an ISPF command table

ISPF supports four different command tables where you can define an ISPF command to invoke File Manager:

- Application command table
- User command table
- Site command table

© Copyright IBM Corp. 2000, 2011

- System command table

You can use the ISPF Command Table Utility (ISPF option 3.9) to create or change a command table that is not currently in use (the system command table, ISPCMDS, is always in use). When you add a command for File Manager to one of these command tables, you can invoke File Manager from any ISPF panel without prefixing the command with TSO.

For example, add the following entry to a command table to enable File Manager to be run from any ISPF panel by typing 'FM' on the command line.

Verb: FM

Action: SELECT PANEL(FMNSTASK) OPT(&ZPARM) SCRNAME(FILEMGR) SUSPEND NEWAPPL(FMN)

Description: File Manager for z/OS

For information about ISPF command tables, see *z/OS ISPF Planning and Customizing*. For information about the ISPF Command Table Utility, see *z/OS ISPF User's Guide Vol II*.

Making File Manager the default VSAM editor

You can configure ISPF to specify File Manager as the VSAM editor that is to be invoked automatically when a VSAM data set is specified in ISPF Options 1, 2, 3, or 11. You do this by updating the ISPF Configuration Table Keyword File to provide values for the VSAM-related keywords listed in Table 5:

Table 5. VSAM-related keywords and values

Keyword	Suggested value
VSAM_EDIT_ENABLED	YES
VSAM_EDIT_COMMAND	FMNINV DSE /
VSAM_EDIT_LIMITED	NO
VSAM_BROWSE_ENABLED	YES
VSAM_BROWSE_COMMAND	FMNINV DSB /
VSAM_BROWSE_LIMITED	NO
VSAM_VIEW_ENABLED	YES
VSAM_VIEW_COMMAND	FMNINV DSV /
VSAM_VIEW_LIMITED	NO
VSAM_RESTRICTED_BROWSE_DATASET	NONE
VSAM_RESTRICTED_EDIT_DATASET	NONE
VSAM_RESTRICTED_VIEW_DATASET	NONE

You then use this keyword file to build an ISPF configuration load module. For information about how to do this, see *z/OS ISPF Planning and Customizing*.

Enabling File Manager to work with certain products

You can enable File Manager to work with other IBM products. These products include COBOL, PL/I, DFSORT, DB2 and WebSphere MQ for z/OS (WebSphere MQ). The following sections describe how you do this.

Enabling WebSphere MQ support

If you want to use File Manager to access WebSphere MQ queue managers and queue contents on the local z/OS system where File Manager is running, you must make the WebSphere MQ libraries available to File Manager.

For File Manager base functions, the WebSphere MQ load libraries SCSQANLE, SCSQAUTH and SCSQLOAD must be made available to the TSO or batch job user, either in the linklist, or as part of the STEPLIB in the TSO procedure (or batch JCL). File Manager MQ functions make use of sending messages to the SYSTEM.COMMAND.QUEUE and creating a dynamic queue to retrieve the replies. The high level qualifier for this reply queue is customizable in the FMN0POPT table, as keyword MQREPHLQ. The sample options module contains a value of "FMNTMQL." for the high level qualifier. When the dynamic queue name is passed to the MQ APIs for construction, it will use the HLQ as specified in the options module, appended with the current userid and a ".*" string. The MQ manager being used will then return the full dynamic name to be used. This high level qualifier may be changed to suit your installations MQ security rules.

Note: ISPF LIBDEF ISPLLIB cannot be used to access these libraries.

For FM/CICS, if using batch submission, these libraries must be added to the batch procedure FMN3CICB, to the STEPLIB concatenation. For more information about FMN3CICB, see “Customizing the batch procedure” on page 241.

For FM/CICS using the FM server method, they must be added to the server's FMNLIB concatenation in FMNCAS. For more information about the FM Server and FMNCAS, see “Modifying FMNCAS” on page 436.

Customizing for processing COBOL copybooks

Overview

If you want to use File Manager view, edit, copy, print, or data create functions, with a COBOL copybook, you need to make a COBOL compiler available to File Manager. This functionality requires ADATA, therefore the COBOL compiler must be one that has the ADATA capability. COBOL compilers such as VS COBOL II, and OS/COBOL, which do not support ADATA, cannot be used. All currently supported versions of IBM Enterprise COBOL for z/OS and OS/390® provide this support, and can therefore be used.

File Manager also provides an internal version of the COBOL compiler, for use in preference to a supported COBOL compiler, or when a supported COBOL compiler is not available to File Manager, (for example, if you are using File Manager in a production environment, where the COBOL compiler is not used). This compiler is installed into FMN.SFMNMOD1, if you installed File Manager into the default libraries, with load modules named FMN9*xxxx*. (The File Manager COBOL compiler is a reduced-function compiler and cannot be used to generate executable COBOL programs.)

File Manager will use its own COBOL compiler:

- When a supported COBOL compiler cannot be accessed.
- Whenever it encounters this DD statement for FMNCOB:

```
//FMNCOB DD DUMMY
```

Normally, if a supported COBOL compiler is found, this will be used. If a supported COBOL compiler cannot be found then the File Manager COBOL compiler will be used.

However, on occasions, even if a supported COBOL compiler is available, you may want to use the File Manager COBOL compiler, for example, if you are asked to do so by IBM support.

For information about using a supported COBOL compiler, see "Using a supported COBOL compiler." For more information about using File Manager's own COBOL compiler, see "Using the File Manager COBOL compiler" on page 16.

If you are uncertain whether File Manager is using a licensed, supported COBOL compiler, or File Manager's COBOL compiler, you can check this, both in batch and under ISPF.

To check which COBOL compiler is being used in batch, submit a File Manager batch job with the control statement:

```
$$FILEM SHOWCOB
```

If a licensed, supported COBOL compiler is available to File Manager, the output of the batch job will show:

```
FMN0024I The Customer Licensed COBOL Compiler will be used.
```

If File Manager is using it's own COBOL compiler, the output of the batch job will show:

```
FMN0023I The File Manager supplied COBOL Compiler will be used
```

To check which COBOL compiler is being used under ISPF, on any File Manager command line enter the command:

```
SHOWCOB
```

If a licensed, supported COBOL compiler is available to File Manager, the message:

```
Using Licensed COBOL
```

is displayed at the top right corner. If File Manager is using it's own COBOL compiler, the message is:

```
Using FM COBOL Compiler
```

Note that for File Manager to successfully use your COBOL compiler, it must be both a licensed and a supported compiler. If you are using a licensed, non-supported compiler (for example, a licensed version of VS COBOL II), the message will still be:

```
Using Licensed COBOL
```

but any compilation of copybooks will fail.

Using a supported COBOL compiler

If a supported COBOL compiler is available and will be used by File Manager, you should review the following sections for customization tasks you might need to perform. In this case do not provide any DD statement:

```
//FMNCOB DD DUMMY
```

Adding the COBOL compiler library to the LINKLIST: If you added FMN.SFMNMOD1 to your LINKLIST, also add the COBOL compiler library to the LINKLIST.

If you did not add FMN.SFMNMOD1 to your LINKLIST, but instead added this library to the STEPLIB DD statement in your TSO logon procedure or your batch job, also add the supported COBOL compiler library to the STEPLIB DD statement.

As an alternative to adding the COBOL compiler library to the LINKLIST or the STEPLIB DD, you can provide access to the COBOL compiler library by allocating an FMNCOB DD for your COBOL compiler library:

```
//FMNCOB DD DSN=your.complib,DISP=SHR
```

You can allocate this DD statement dynamically, in your TSO logon procedure, or in a batch job.

Adding the COBOL compiler library to the batch JCL skeletons: You might also want to add your COBOL compiler library to the STEPLIB DD statements in the File Manager batch JCL skeleton, FMNFTEXC, and the FM/IMS batch JCL skeleton, FMN1FTEX. For more information about these JCL skeletons, see "Changing the JCL skeleton for batch mode" on page 24 and "Tailoring the job control skeletons" on page 189.

Authorizing the COBOL compiler library for batch processing: If you are running File Manager APF-authorized in batch mode, and you want to run File Manager batch jobs using COBOL copybook templates, then you **MUST** also authorize the COBOL compiler library. Alternatively, run your File Manager batch jobs un-authorized. (If you have not authorized your COBOL compiler library and it is in a JOBLIB or STEPLIB concatenation with your File Manager load library, then File Manager will not be APF-authorized in batch.)

Suppressing COBOL compiler warning messages: If you set TERM=YES (TERM) as a default COBOL compiler option, add an allocation for FMNTERM DD to your TSO logon procedure,

```
//FMNTERM DD DUMMY
```

This will suppress unnecessary compiler warning messages.

Using COBOL compiler options with File Manager: File Manager overrides certain COBOL compiler options when it invokes the COBOL compiler. Therefore these options cannot be marked as 'fixed' in the COBOL default options module IGYCDOPT. These options are:

```
ADATA         LIB
ADEXIT        LINECOUNT
ARITH         MSGEXIT
COMPILE       OBJECT
DBCS          OPTIMIZE
DECK          PRTEXIT
FLAG          SQL
INEXIT
```

If you have any of these options fixed at your site, you will need to build a File Manager-specific COBOL options module that does not have these options fixed, and make this module available to the COBOL compiler invoked by File Manager. For information on changing the COBOL default options and building an options module for a specific purpose, see the appropriate COBOL Customization book.

If you build a File Manager-specific COBOL options module, but do not want to make it generally available elsewhere other than to File Manager, make a copy of your COBOL compiler library containing this options module. You can then make this copy of the COBOL compiler library available to File Manager only, by means of the FMNCOB DD statement, allocated dynamically, in your TSO logon procedure, or in a batch job.

Using the File Manager COBOL compiler

If File Manager is to use it's own COBOL compiler, and there is a supported COBOL compiler available, then to activate the File Manager COBOL compiler, you must provide a DD statement:

```
//FMNCOB DD DUMMY
```

as described in the following sections.

If a supported COBOL compiler is not available to File Manager, then File Manager will use the version of the COBOL compiler shipped with File Manager regardless.

If File Manager is to use this COBOL compiler, you should review the following sections for customization tasks you might need to perform. Not all the functions available using a supported COBOL compiler are available using the File Manager compiler, and for functions that are available, the customization might differ.

Adding the COBOL compiler library to the LINKLIST: The File Manager COBOL compiler load modules are shipped in FMN.SFMNMOD1. Therefore to make this COBOL compiler available in the LINKLIST, add FMN.SFMNMOD1 to your LINKLIST.

Furthermore, to enable File Manager to use this compiler, you must allocate a DD statement

```
//FMNCOB DD DUMMY
```

You can allocate this DD statement dynamically, in your TSO logon procedure, or in a batch job. You must do this, whether or not you added FMN.SFMNMOD1 to the LINKLIST.

Adding the COBOL compiler library to the batch JCL skeletons: You might also want to use the File Manager COBOL compiler in the File Manager batch JCL skeleton, FMNFTEXC, and the FM/IMS batch JCL skeleton, FMNFTEX. To do this, uncomment the

```
//FMNCOB DD DUMMY
```

statement in the skeletons. For more information about these batch JCL skeletons, see “Changing the JCL skeleton for batch mode” on page 24 and “Tailoring the job control skeletons” on page 189.

Authorizing the COBOL compiler library for batch processing: If you are running File Manager APF-authorized in batch mode, there is no further action you need to take to use the File Manager COBOL compiler authorized. If you do not want to run your batch jobs authorized when using this compiler, then you must run all your File Manager batch jobs un-authorized.

Suppressing COBOL compiler warning messages: If you have set TERM as a COBOL compiler option, add an allocation for FMNTERM DD to your TSO logon procedure,

```
//FMNTERM DD DUMMY
```

This will suppress unnecessary compiler warning messages.

Using COBOL compiler options with File Manager: File Manager overrides certain COBOL compiler options when it invokes the COBOL compiler. These options are:

```
ADATA         INEXIT
ADEXIT        LIB
ARITH         LINECOUNT
COMPILE       OBJECT
DBCS          OPTIMIZE
DECK          PRTEXIT
FLAG          SQL
```

If you are using the File Manager COBOL compiler, these options are not marked as 'fixed' in the provided COBOL default options module, FMN9DOPT, and you cannot change this. There is no facility to manipulate the options provided in the File Manager COBOL compiler.

Customizing for processing PL/I include books

If you want to use File Manager view, edit, copy, print, or data create functions with a PL/I include book, you must use the Language Environment runtime library instead of any PL/I runtime library.

To ensure that you use the correct Language Environment runtime, either add the Language Environment runtime library, SCEERUN, to your LINKLIST, or add it to the STEPLIB DD statement in your TSO logon procedure. You must add SCEERUN in front of any other PL/I runtime library, in the LINKLIST or STEPLIB DD statement.

Furthermore you must ensure that you have the PTFs fixing APAR PK31243 applied to Language Environment. For more information, refer to the *File Manager Program Directory*.

Customizing for processing HLASM copybooks

If you want to use File Manager view, edit, copy, print, or data create functions with a HLASM copybook, ensure that the HLASM load library, ASM.SASMMOD1, is available to File Manager. You can do this either by adding it to your linklist, or to the STEPLIB DD statement in your batch job.

To use HLASM in a batch job, specify LANG=HLASM on the File Manager function statement. For example, to print a member of a data set, using an HLASM copybook, you would specify a File Manager function statement similar to :

```
$$FILEM DSP FORMAT=TABL,TCIN='FMN.SFMNSAM1(FMNACPY)',LANG=HLASM
```

Customizing to use DFSORT to improve File Manager performance

You can optionally enable File Manager to use the DFSORT COPY function to improve the performance of the File Manager Data Set Copy and Data Set Print functions, both under ISPF and in batch. In addition, if DFSORT is enabled, its' COPY function will be used to copy data to Auxiliary Storage (VSAM RRDS) when editing large data sets under ISPF.

Important Information

You do not need to have a DFSORT license to enable File Manager to use DFSORT, but you must use DFSORT R14, DFSORT V1R5, or higher.

If you use DFSORT R14 you must have PTFs UQ90054 and UQ84945 applied.

If you use DFSORT V1R5 you must have PTF UQ84946 applied.

You must apply these PTFs as appropriate, even if DFSORT is not your primary sort product.

Note: DFSORT R14, or DFSORT V1R5, or higher, are available with all releases of z/OS.

Overview

File Manager will use DFSORT to improve the performance of the Data Set Copy and Print functions, provided it can find DFSORT and verify that DFSORT is at the correct maintenance level for File Manager use. DFSORT is an optional, priced feature of z/OS; you must have a DFSORT license to use DFSORT outside File Manager. However, the DFSORT code is always shipped with z/OS; as long as you did not delete the DFSORT libraries when you installed z/OS, you can enable File Manager to use the DFSORT code. File Manager accesses DFSORT using the aliases ICEDFSRT and ICEDFSRB.

The following sections describe how you can customize File Manager and implement DFSORT to allow File Manager to use the DFSORT code. If you want File Manager to use DFSORT, follow the instructions in the appropriate sections, depending if you use DFSORT as your primary sort product, or a sort product from an independent software vendor.

Customizing when DFSORT is the primary sort product

If you use DFSORT as your primary product, you will have installed it in one of the following ways:

- **Resident:** The DFSORT library, SORTLPA, is in LPALST, and the DFSORT library, SICELINK, is in LINKLIST.
- **Non-Resident:** The DFSORT libraries SICELINK and SORTLPA are private libraries.

1. If you have installed DFSORT resident, you need take no further action to make DFSORT available to File Manager.
2. If DFSORT is installed non-resident, you will need to take different actions to enable File Manager to use DFSORT under ISPF and in batch.

   Under ISPF (interactively)

   To enable File Manager to use DFSORT under ISPF, add the DFSORT libraries, SICELINK and SORTLPA, in that order, to the STEPLIB DD statement in your TSO logon procedure. (If you do not want to add these libraries to your TSO logon procedure, you can add them via the TSOLIB command, before invoking ISPF.)

   In batch

   To enable File Manager to use DFSORT in batch, add the DFSORT libraries, SICELINK and SORTLPA, in that order, to the JOBLIB or STEPLIB DD statement for the job or step that uses File Manager. You might want to create a File Manager JCL procedure for use at your site.

   You could also add the DFSORT libraries to the STEPLIB DD statement in the batch JCL skeleton. If you do this, you must add them in the order SICELINK, SORTLPA. For more information about the batch JCL skeleton, see “Changing the JCL skeleton for batch mode” on page 24.

Customizing when DFSORT is not the primary sort product

If you use a non-IBM sort product as your primary sort product, and you want to enable File Manager to use DFSORT, you **must** place the DFSORT libraries **after** your primary sort product libraries in the system search order.

Note: DFSORT and non-IBM sort products typically have some entry points with the same names, for example SORT and ICEMAN. Therefore if the DFSORT libraries are placed before your primary sort products libraries, DFSORT will become the primary sort product. In this situation, if you do not have a license for DFSORT all sort jobs that use your primary sort product will fail.

If you use a non-IBM sort product as your primary sort product, you will have installed it in one of the following ways:

- **Resident:** Your primary sort products libraries are in LPALST or the LINKLIST or both.
- **Non-Resident:** Your primary sort products libraries are private libraries.

1. If you have installed your sort product resident, place DFSORT's SORTLPA and SICELINK libraries in the LINKLIST, (**after** your primary sort product libraries in the LINKLIST, if any), to enable File Manager to use DFSORT.
2. If you have installed your sort product non-resident, you will need to take different actions to enable File Manager to use DFSORT under ISPF and in batch.

   Under ISPF (interactively)

   To enable File Manager to use DFSORT under ISPF, add the DFSORT libraries, SICELINK and SORTLPA, in that order, **after** your primary sort product libraries, to the STEPLIB DD statement in your TSO logon procedure. (If you do not want to add these libraries to your TSO logon procedure, you can add them via the TSOLIB command, before invoking ISPF.)

   In batch

   To enable File Manager to use DFSORT in batch, add the DFSORT libraries, SICELINK and SORTLPA, in that order, to the JOBLIB or STEPLIB DD statement for the job or step that uses File Manager, **after** your primary sort products libraries. You might want to create a File Manager JCL procedure for use at your site.

   You could also add the DFSORT libraries to the STEPLIB DD statement in the batch JCL skeleton. If you do this, you must also add your primary sort products libraries. You must add them in this order, your sort product libraries, followed by SICELINK, then SORTLPA. For more information about the batch JCL skeleton, see "Changing the JCL skeleton for batch mode" on page 24.

DFSORT uses an SVC in the SORTLPA library. By default this is SVC 109. This SVC is used by DFSORT to record SMF type-16 records or to use IBM cached DASD devices (such as the 3990 model 3). If your non-IBM sort product uses its own version of SVC 109, you will need to make an alternative SVC number the default SVC for DFSORT. For information on installing DFSORT's SVC, see *DFSORT Installation and Customization* for your release of DFSORT.

File Manager does not need the DFSORT SVC for its own operations. Therefore, if DFSORT if not your primary sort product, it is not necessary to install the DFSORT SVC.

Determining if DFSORT is being used

If you expect File Manager to be using DFSORT to improve performance, and it does not appear to be doing so, you can check this both in batch and under ISPF.

To check whether DFSORT is being used in batch, submit a File Manager batch job with the control statement:

```
$$FILEM SHOWSORT
```

If DFSORT is available to File Manager, the output of the batch job will show:

```
SORT debugging is on
```

If DFSORT is not available, the output will show:

```
DFSORT not available
```

To check whether DFSORT is being used under ISPF, on any File Manager command line enter the command:

```
SHOWSORT
```

If DFSORT is available to File Manager under ISPF, the message:

```
SORT debugging is on
```

is displayed at the top right corner. If DFSORT is not available, the message is:

```
DFSORT not available
```

Summary

Once you have determined, from the above, the correct way to implement DFSORT for use by File Manager at your site, follow the instructions in *DFSORT Installation and Customization*. You do not need to customize DFSORT for use by File Manager, since File Manager automatically customizes its use of DFSORT's COPY function. File Manager does not use DFSORT's SORT or MERGE functions.

Note: When File Manager uses DFSORT it presumes that DFSORT will produce message output and will open data set FMNSRTP for DFSORT messages. If the DFSORT installation option MSGPRT is set to NONE (MSGPRT=NONE) then an OPEN abend may occur when the File Manager DFSORT interface module (FMNDFSRT) tries to open message data set FMNSRTP. To avoid this set the DFSORT installation option MSGPRT to ALL (MSGPRT=ALL).

Binding DB2 to use File Manager object access method (OAM) functions

If you intend to use File Manager OAM functions, you must bind the File Manager plan, FMNODIRS, into into the DB2 system where the OAM objects are defined, to allow the proper working of OAM functions. A sample job, FMNBDIRS, is provided in FMN.SFMNSAM1 to help you do this.

Customize FMNBDIRS according to your DB2 installation requirements. See the instructions in the sample job for more information about changes you need to make. FMNBDIRS is distributed in FMN.SFMNDBRM.

The following DB2 authorities must also be granted to each user of OAM functions:

- Directory processing requires READ authority for the following tables:
  - GROUP*nn*.OSM_OBJ_DIR (all group directories) *nn* = 00 ... 99 inclusive
  - SYSIBM.SYSTABLES
  - *owner*.VOLUME
  - OAMADMIN.CBR_COLLECTION_TBL
  - OAMADMIN.CBR_MGT_CLASS_TBL
  - OAMADMIN.CBR_STO_CLASS_TBL

- Restoring objects requires UPDATE authority for the following tables:
  - GROUP*nn*.OSM_OBJ_DIR (all group directories) *nn* = 00 ... 99 inclusive
  - OAMADMIN.CBR_COLLECTION_TBL

Running File Manager with APF-authorization

In batch mode, File Manager can run APF-authorized or non APF-authorized. File Manager cannot run APF-authorized under ISPF. You make File Manager authorized by adding FMN.SFMNMOD1 to your site-specific IEAAPF*xx* or PROG*xx* member in SYS1.PARMLIB.

Notes:

1. If you are running File Manager APF-authorized in batch mode, and you want to run File Manager batch jobs using COBOL copybook templates, then you **MUST** also authorize the COBOL compiler library. If you do not do this, you will receive an ABENDS306 from the batch job.
2. If you plan to access source code in a library management system that does not provide a SUBSYS interface, then you cannot run File Manager APF-authorized. For information about accessing source code in an LMS without a SUBSYS interface, see "Accessing source code in an LMS without SUBSYS interface" on page 68.

To activate any changes you have made to SYS1.PARMLIB members, either restart your system, or use the appropriate commands for your site to dynamically activate the changes.

Troubleshooting problems with APF-authorization

If you expect File Manager to be running APF-authorized in batch mode, and it does not appear to be doing so, you can check this by submitting a File Manager batch job with the control statement:

```
$$FILEM VER
```

If the VER function indicates that File Manager is not APF-authorized, it means that FMN.SFMNMOD1 is not APF-authorized. Check that IEAAPF*xx* is set up correctly and selected in IEASYS*xx*. You will need to re-IPL to activate IEAAPF*xx*. Alternatively, you can use PROG*xx* (if it is available on your system).

If a JOBLIB or STEPLIB statement is used to specify the File Manager load library, ensure that the File Manager load library is not concatenated with a non APF-authorized library. If a library concatenation includes a non-authorized library, (for example, you have not authorized your COBOL compiler library), then no library in that concatenation will be authorized.

If you start File Manager from an ISPF selection panel, the VER command will always show that File Manager is not APF-authorized, as File Manager cannot run APF-authorized under ISPF. File Manager functions that require APF-authorization are not supported under ISPF.

If you plan to use SMF for audit logging, for File Manager or the DB2 or IMS components, you must ensure that the module FMNSMF is APF-authorized, and provide an SMF record number in the appropriate options modules. If you find that despite having set up your options correctly and authorizing FMNSMF, no audit records are logged to SMF, check that the library containing FMNSMF is APF-authorized and that it is not in a concatenation with other libraries that are not APF-authorized.

If FMNSMF is not properly authorized no audit logging will be done to SMF even if you have provided a valid SMF record number. In this case your users may get the message `UNABLE TO LINK TO FMNSMF`.

Language Environment considerations

During the File Manager Initialization process, Language Environment (LE) preinitialization facilities (CEEPIPI) are used to create and initialize a common run-time environment.

This means that if LE is active when File Manager is invoked, there are restrictions on File Manager functionality because the CEEPIPI functions are not available.

Note that these restrictions **only** apply if LE is active when File Manager is invoked. (LE active means that File Manager has been invoked directly via an LE application (for example, from Enterprise COBOL).)

The restrictions are:

- TRAP(ON,NOSPIE) should be set to allow File Manager abend handler to function correctly.
- An LE-enabled I/O exit cannot be used.
- MOD_DATE REXX Built-in function cannot be used.
- Scramble Exit (if used) must be in Assembler (non-LE).
- Library Management exit (LMS=USERLMS) cannot be used.

Chapter 3. Customizing File Manager

This chapter describes how to customize File Manager. You do this after you have installed File Manager.

Changing the default options

Default processing options are supplied with File Manager in the module FMN0POPT. You can change these options to suit your installation requirements. You use the usermod FMNUMODP to do this. For a description of the File Manager options and the values you can specify, see Appendix A, "File Manager options," on page 289.

You can find the FMN0POPT macro statements, as supplied by IBM, in member FMN0POPT of FMN.SFMNSAM1.

You change the options as follows:

1. Copy the member FMN0POPT from FMN.SFMNSAM1 into your own source library.
2. Change the default options in the FMN0POPT member in your library as required.
3. Modify the FMNUMODP member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMNUMODP.

Note: You can also use the sample job FMN0POPH to assemble FMN0POPT if you do not want to use SMP/E.

For users connected by means of Rational Developer for z, the processing options module name and sample member is FMN4POPT. Parameters and values are as per the FMN0POPT. For a SMP/E usermod, a sample member FMN4UMDP is supplied. The sample job FMN4POPH is also supplied if you do not want to use SMP/E to customize these options.

Setting the default national language

If you installed the Japanese or the Korean component (or you provided locally translated messages and panels), you can change the default national language for your installation, for File Manager batch processing. The language used by File Manager under ISPF depends on the language setting for your ISPF session. To change the language for File Manager batch processing, set the LANGUAGE option in FMN0POPT to your language. See Table 15 on page 64 for the value to specify for LANGUAGE. See "LANGUAGE" on page 306 for more information about the LANGUAGE option.

Furthermore, FM/CICS uses the setting of LANGUAGE in FMN0POPT to determine the language to use for panels and messages. Therefore if you have installed FM/CICS and you are not using English, you should change the setting of LANGUAGE in FMN0POPT to your language. This will be Japanese or Korean

© Copyright IBM Corp. 2000, 2011

if you have installed the FM/CICS Japanese or Korean components, or another language if you have provided FM/CICS messages and panels in another language.

Changing the JCL skeleton for batch mode

Several functions in File Manager are available in batch mode, and the TSO batch environment. For these functions to run successfully, appropriate job control language statements must be provided. This is done by means of the set processing options, and a job control skeleton. The job control skeleton for File Manager is the member FMNFTEXC in FMN.SFMNSLIB.

FMNFTEXC

There are some DD statements in FMNFTEXC, that you may need to change, in the distributed version of the skeleton. These statements are commented out, and are documented below.

```
//STEPLIB  DD DSN=&FMNSMOD1,DISP=SHR
//* If using WMQ queues via batch, uncomment and update MQHLQ to suit
//* your sites WMQ high level qualifier
//*        DD DSN=MQHLQ.SCSQLOAD,DISP=SHR
//*        DD DSN=MQHLQ.SCSQANLE,DISP=SHR
//*        DD DSN=MQHLQ.SCSQAUTH,DISP=SHR
//*        DD DSN=IGY.SIGYCOMP,DISP=SHR
//*FMNCOB  DD DUMMY     Uncomment to force use of FM COBOL Compiler
//*FMNCLERR DD SYSOUT=* Uncomment to force output of Compiler listing
```

These statements are described here.

FMNFTEXC assigns a STEPLIB DD statement to the File Manager load library. FMNFTEXC assumes that you have installed File Manager into the default target libraries, and that the load library is FMN.SFMNMOD1. If you have installed File Manager into a different library than FMN.SFMNMOD1, you may either change the SFMNMOD1 parameter in the FMN0POPT module or change the //STEPLIB DD statement in the FMNFTEXC skeleton.

If necessary, change &FMNSMOD1 in the statement

```
//STEPLIB  DD DSN=&FMNSMOD1,DISP=SHR
```

to the name of your File Manager load library.

From version 10, File Manager is capable of processing Websphere Message Queuing (WMQ) queues in both online and batch scenarios for the common utilities; create, copy, compare and print. To interface with the Websphere product, the libraries SCSQLOAD, SCSQANLE and SCSQAUTH must be made available to the JOBLIB, STEPLIB or linklist concatenation. You can do this by uncommenting these lines:

```
//*        DD DSN=MQHLQ.SCSQLOAD,DISP=SHR
//*        DD DSN=MQHLQ.SCSQANLE,DISP=SHR
//*        DD DSN=MQHLQ.SCSQAUTH,DISP=SHR
```

The STEPLIB DD statement in this skeleton also concatenates another statement as follows.

```
//*        DD DSN=IGY.SIGYCOMP,DISP=SHR
```

which appears as a comment in FMNFTEXC.

In this statement, IGY.SIGYCOMP is the supported, licensed COBOL compiler library. Some batch functions which make use of COBOL require this library. If you did not add your supported COBOL compiler library to your LINKLIST, remove the * to uncomment the line, and change the DSN to the name of your supported COBOL compiler library. All currently supported versions of IBM Enterprise COBOL for z/OS and OS/390 are supported by File Manager.

If you have created a COBOL compiler library for File Manager with a special version of IGYCDOPT, you can use this DD statement to make it available to File Manager for all batch jobs using COBOL templates. See page 15 for information about a special version of IGYCDOPT for File Manager.

If you do not have a supported COBOL compiler available to File Manager (for example, if you are using File Manager in a production environment, where the COBOL compiler is not used), File Manager can use it's own internal COBOL compiler. To enable this feature, remove the * to uncomment the third statement,

```
//*FMNCOB  DD DUMMY     Uncomment to force use of FM COBOL Compiler
```

as indicated in the statement. If File Manager encounters a statement, //FMNCOB DD DUMMY, the File Manager COBOL compiler will be used. For information about the File Manager COBOL compiler, see “Using the File Manager COBOL compiler” on page 16.

If you want to produce a compile listing of copybooks processed by File Manager functions, in case of compile errors, then remove the * to uncomment the fourth statement,

```
//*FMNCLERR DD SYSOUT=* Uncomment to force output of Compiler listing
```

as indicated in the statement. You can also change the specification of SYSOUT on this statement, if necessary, to suit your site's requirements.

If you plan to enable File Manager to use the DFSORT COPY function to improve File Manager performance, you might want to add the DFSORT libraries to this STEPLIB DD statement. If you do this, you must add them in the order SICELINK, followed by SORTLPA. If you choose to do this and DFSORT is not your primary sort product, you must also add your sort products libraries in front of the DFSORT libraries. For more information about using DFSORT to improve File Manager performance, see “Customizing to use DFSORT to improve File Manager performance” on page 17.

If you plan to enable File Manager to access COBOL copybooks, PL/I include books, or HLASM copybooks in library management system (LMS) libraries, and you plan to link edit your LMS exit, FMNCRAEX, into your own load library, you might want to add this load library to the STEPLIB DD statement in the JCL skeleton. For more information about enabling File Manager to access LMS libraries, see Chapter 8, “Customizing File Manager to use library management system libraries,” on page 67.

You use the usermod FMNUMODB to modify the job control skeleton. FMNUMODB is distributed in FMN.SFMNSAM1. To do this:

1. Copy the FMNFTEXC member from FMN.SFMNSLIB to your own source library.
2. Modify the FMNFTEXC member in your own library. Change the name of the File Manager load library on the first line of the STEPLIB DD statement to the name of your load library, if necessary. Uncomment the second line of the

STEPLIB DD statement if you want to add your COBOL compiler library, and change the DSN to the name of your COBOL compiler library, if necessary. Uncomment the WMQ lines, if necessary.. Add your sort and LMS exit libraries if necessary.

3. Modify the FMNUMODB member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMNUMODB.

Note: File Manager does not provide support for the automatic generation of job routing control statements in the JCL generated by FMNFTEXC.

Customizing File Manager batch return codes

File Manager provides the facility for you to control the value of return codes issued by the File Manager batch utilities. You can set return codes to values more suitable for your site, for example, to suit your job control environment. Return codes cannot be changed on an ad hoc basis. They are established for the whole installation.

You can customize selected return codes issued by the following batch utilities:

- Data set copy (DSC)
- Data set generate (DSG)
- Data set compare (DSM)
- Data set print (DSP)
- Data set update (DSU, and including DSEB)
- Display VTOC (DVT)
- Find/change (FCH)

You customize the batch utility return codes using the macro statements sample member, FMN0RETC, and the usermod FMNUMODR, to create a load module, FMN0RETC. The samples FMN0RETC and FMNUMODR are distributed in FMN.SFMNSAM1. FMN0RETC in FMN.SFMNSAM1 shows the conditions that can be customized for each utility, with the default return codes for those conditions.

During initialization, File Manager will attempt to load FMN0RETC, and if the module is found its contents will be used to build a table of customized condition return codes. If FMN0RETC is not found, all the customizable conditions will issue their default return codes.

To customize the return codes:

1. Copy the FMN0RETC member from FMN.SFMNSAM1 to your own source library.
2. Modify the FMN0RETC member in your own library, as required. Change the number against the required conditions in FMN0RETC to the return code values you want those conditions to give. Note that you cannot change any condition to a value of 16. The value 16 is reserved for use by File Manager. Figure 2 shows the FMN0RETC macro statements as supplied by IBM.
3. Modify the FMNUMODR member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMNUMODR.

```
FMN0RETC TITLE 'File Manager for z/OS : Return Codes (Batch Utilities)'
FMN0RETC CSECT
FMN0RETC AMODE 31
FMN0RETC RMODE ANY
         FMN0RETI FUNC=DSC,      DSC conditions                        X
               NORECSSOME=1,      No records copied for some members   X
               NORECSANY=2,       No records copied for any members    X
               NONESEL=4,         No records selected to copy          X
               NOMEMBERS=4,       No members to process                X
               EMPTY=4,           Empty input data set                 X
               SKIPPED=4,         Input packed, so skipped             X
               NOREPLACE=4,       NOREPLACE prevented copy             X
               NOCPYDUPE=2,       No copy - duplicate                  X
               NOCPYREXX=2,       No copy - REXX                       X
               RECTRC=0,          Records truncated                    X
               FLDTRC=0,          Fields  truncated                    X
               RECSHORT=8,        DSC external form output truncated   X
               REXXCONFLICT=3     REXX return message conflict
         FMN0RETI FUNC=DSG,      DSG conditions                        X
               NORECSOUT=4        Zero records specified
         FMN0RETI FUNC=DSM,      DSM conditions                        X
               NOTMATCH=1,        Compare sets not empty & don't match X
               ONEEMPTY=2,        One compare set empty                X
               SKIPPED=4,         Input packed, so skipped             X
               BOTHEMPTY=4,       Both compare sets empty              X
               INVMAP=4           Invalid mapping
         FMN0RETI FUNC=DSP,      DSP conditions                        X
               MBRERR=1,          Some members not printed             X
               PRTERR=2,          Print error encountered              X
               NONESEL=4,         No records selected to print         X
               NOMEMBERS=4,       No members to process                X
               SKIPPED=4,         Input packed, so skipped             X
               EMPTY=4,           Empty input data set                 X
               NOPRTREXX=2,       No print - REXX                      X
               REXXCONFLICT=3     REXX return message conflict
         FMN0RETI FUNC=DSU,      DSU conditions                        X
               MBRNOTUPDT=1,      Some members not updated             X
               NOCHANGE=2,        Change failed                        X
               NONEUPDATED=4,     No records updated                   X
               NOMEMBERS=4,       No members to process                X
               SKIPPED=4,         Input packed, so skipped             X
               EMPTY=4            Empty input data set
         FMN0RETI FUNC=DVT,      DVT conditions                        X
               NOENTRY=4          No matching entries
         FMN0RETI FUNC=FCH,      FCH conditions                        X
               FSOME=1,           Some OK, some not OK                 X
               CFAIL=2,           Change failed                        X
               NOHIT=4,           No strings found to change           X
               NOMEMBERS=4,       No members to process                X
               SKIPPED=4,         Input packed, so skipped             X
               LOWSTOR=12,        Ran low on storage during FCH        X
               EMPTY=4            Empty input data set
         FMN0RETI END
         END   FMN0RETC
```

Figure 2. FMN0RETC macro statements as supplied by IBM

The return code from a batch utility will be the highest return code, (customized or otherwise), for all the conditions that arose during the execution of the utility. For example, suppose you have changed the return code for "No members to process" in the DSC utility from 4 to 9. If a condition arises that in isolation would result in return code 8 (normally an error), and the "No members to process" condition also occurs, the return code for the DSC utility will be 9.

Customizing File Manager batch return codes

The return code from a batch job executing more than one utility will continue to be the maximum of the codes, (customized or otherwise), returned by each of the utilities. The exception to this is that if any of the utilities encounters a terminating condition then the job step will terminate with a return code 16.

Notes:

1. Several conditions can be present in a function concurrently. For example, using DSP with a template on an empty file may result in both the "empty input data set" condition, and the "no records selected to print" condition. If you customize a return code, be aware that a higher return code for another condition may override it or be over-ridden by it. It is suggested that if you customize one return code for a given function you review that function's other return codes for compatibility.
2. In batch, you might want File Manager to abend rather than end with the original or customized non-zero return code, to prevent the execution of successive steps or jobs. You do this by means of the ABENDCC installation option, with a value less than or equal to the return code. This will then be transformed to an abend (Abend 999, Reason Code=888 (hex: 378)). For more information, see “ABENDCC” on page 289, and also the ABENDCC option of the SET command, described in “SET (Set Processing Options)” in the *File Manager User's Guide and Reference*.

Changing the print and display translation tables

By default, File Manager translates all unprintable characters in printed output to blanks (option PRTTRANS=ON), and non-displayable characters in displayed output to periods.

File Manager provides a default translation table for English (FMNTRTBS), for Japanese (FMNTRJPN), and for Korean (FMNTRKOR). For other languages you will need to provide and maintain your own Print and Display Translation table. See Chapter 7, “Customizing File Manager for national languages,” on page 61 for information on customizing File Manager for your language.

Sometimes you might want to display special characters on a terminal during a File Manager session, or print File Manager output in lowercase alphanumeric characters. To do this, you change the English translation table source member, FMNTRTBS. FMNTRTBS is distributed in FMN.SFMNSAM1.

The translate tables that are used are:

TRPRT used to translate unprintable characters to blanks in printed output. Change this table to include all the characters that you want to print.

TRBRW used to translate non-displayable characters to periods in displayed output. Change this table to include all the characters that you want to display on your terminal.

TRUPC used to translate lower case characters to uppercase characters for File Manager command verbs and command keywords only. This table generally does not need to be customized in an SBCS environment.

TRUP used to translate lower case characters to uppercase characters.

TRINV used as a second-step translate table to translate characters that are not supported by your particular display terminal. Display data is first translated using the TRBRW table, and the result of that translate is translated using this table. You should not need to change this table.

Changing the translation tables in English

If you are using English, you use the usermod FMNUMODT and FMNTRTBS, to change the print and display tables. (To change the tables in languages other than English, see Chapter 7, "Customizing File Manager for national languages," on page 61.) FMNUMODT is distributed in FMN.SFMNSAM1. To do this:

1. Check that the terminal on which you want to display File Manager panels supports the display of special characters, or that the universal character buffer (UCB) of your printer has the characters you want to use.
2. Change the File Manager options to specify PRTTRANS=ON. For information on how to do this, see "Changing the default options" on page 23.
3. Modify the File Manager translation table as follows:
   a. Copy the member FMNTRTBS from FMN.SFMNSAM1 into your own source library.
   b. Change the translation table definition statements in FMNTRTBS in your source library, according to your requirements, as described above.
   c. Modify the FMNUMODT member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
   d. Install SMP/E usermod FMNUMODT.

Changing the ASCII translation tables

You can use File Manager to translate tape data as follows:

- Translate tape input from ASCII format to EBCDIC format.
- Translate tape output from EBCDIC format to ASCII format.
- Translate tape input from ASCII format to EBCDIC format, and translate tape output from EBCDIC format to ASCII format.

If you want to use an ASCII or EBCDIC character set other than the character sets supplied by IBM, you can change the translation table definition statements.

You use the usermod FMNUMODA and the member FMNASCII to change the ASCII translation tables. FMNUMODA and FMNASCII are distributed in FMN.SFMNSAM1. To do this:

1. Copy the member FMNASCII from FMN.SFMNSAM1 to your own source library.
2. Change the translation table definition statements in FMNASCII in your source library, as required.
3. Modify the FMNUMODA member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMNUMODA.

Changing the ASCII translation tables

Chapter 4. Customizing the File Manager security environment

File Manager provides security for system oriented functions in one of two ways, either through RACF (or an equivalent security product) or through the FMNSECUR exit.

If Security Server, RACF 1.9 (or later), or an equivalent security product, is active, the System Authorization Facility (SAF) with the File Manager enhanced security facility is used for access control and authorization verification. Authorization is controlled by File Manager-specific profiles in the FACILITY class. See "Setting up the security environment by using RACF or an equivalent security product" for information on defining profiles for RACF or an equivalent security product. If you use another security product than RACF, consult the documentation for your product to determine how to define the FILEM facility to your product.

As a minimum, you should define the following individual group profiles:

```
RDEFINE FACILITY FILEM.DISK.*          UACC(READ)   or NONE
RDEFINE FACILITY FILEM.DISK.FULLPACK   UACC(NONE)
RDEFINE FACILITY FILEM.LOADMOD.UPDATE UACC(READ)   or NONE
RDEFINE FACILITY FILEM.TAPE.*          UACC(READ)   or NONE
RDEFINE FACILITY FILEM.TAPE.BLP        UACC(NONE)
RDEFINE FACILITY FILEM.VSAM.*          UACC(READ)   or NONE
RDEFINE FACILITY FILEM.OAM.*           UACC(READ)   or NONE
RDEFINE FACILITY FILEM.OTHER.ALL       UACC(READ)   or NONE
```

If SAF with RACF 1.9 (or an equivalent security product), is not active at File Manager initialization time, all File Manager special security checks during that File Manager session are passed to the FMNSECUR user exit instead of to SAF.

To use FMNSECUR, it must be installed in the LPA. If the FMNSECUR module is required and it cannot be found in the LPA, an error message is displayed, and File Manager will not initialize.

FMNSECUR is a customizable exit. It provides FMNS macros, which allow you to define a table of user names or job names, File Manager-protectable resources (called profiles), and access levels. For information on FMNSECUR, see "Setting up the security environment by using FMNSECUR" on page 38.

Note: The FMNSECUR module will not be used (even if present) if SAF with RACF 1.9 (or an equivalent security product), is active at File Manager initialization time.

Setting up the security environment by using RACF or an equivalent security product

You perform the following steps to define profiles for RACF or your equivalent security product. These steps assume that your security administrator has already controlled access to DASD volumes (DASDVOL) and facilities (FACILITY).

Notes:

1. For more information about RACF resource profiles, see *z/OS Security Server RACF Command Language Reference*.

© Copyright IBM Corp. 2000, 2011

2. The following sections contain examples of setting up facility classes for File Manager using RACF. If you are using an equivalent security product, refer to your products' documentation that describes the definition and usage of facility classes.

You can give or deny some, or all, users access to any of the following groups of File Manager functions:

FILEM.DISK.INPUT	Disk input functions
FILEM.DISK.UPDATE	Disk update functions
FILEM.TAPE.INPUT	Tape input functions
FILEM.TAPE.OUTPUT	Tape output functions
FILEM.TAPE.DUPLICATE	Tape copy functions
FILEM.TAPE.UPDATE	Tape update functions
FILEM.VSAM.UPDATE	VSAM update functions
FILEM.OAM.OUTPUT	OAM output functions
FILEM.OAM.UPDATE	OAM update functions
FILEM.LOADMOD.UPDATE	Load module update functions
FILEM.OTHER.ALL	All other functions
FILEM.TAPE.BLP	See "Controlling Bypass Label Processing (BLP)" on page 37
FILEM.DISK.FULLPACK	See "Controlling fullpack access to DASD volumes" on page 35

For more information about these groups, see Table 8 on page 46.

Three facility groups are provided to allow you to control access to File Manager base function, and to FM/IMS and FM/DB2, from the FM/CICS primary option menu. These groups are:

FILEM.CICS.BASE	Access to File Manager base function
FILEM.CICS.IMS	Access to FM/IMS
FILEM.CICS.DB2	Access to FM/DB2

If a userid running FM/CICS has read access to any of these groups, then the associated function (FM, FM/IMS or FM/DB2) will appear on the FM/CICS primary option menu and the user can invoke these functions, if they are installed.

If you have installed and customized the FM/CICS component, you should review your requirement for this access.

For more information about FM/CICS, see Part 4, "Customizing File Manager CICS Component," on page 229, and also the *File Manager User's Guide and Reference for CICS*.

Three facility groups are also provided to enable you to protect update functions in File Manager base function, FM/DB2, and FM/CICS. They are:

FILEM.BASE.UPDATE	Protect update functions in File Manager base
FILEM.DB2.UPDATE	Protect update functions in FM/DB2
FILEM.CICS.UPDATE	Protect update functions in FM/CICS

(This aspect of security is handled differently for FM/IMS, see "Controlling access to IMS subsystems and FM/IMS functions" on page 191.)

These facility classes also require the option SEC=YES to be specified in FMN0POPT (for File Manager base), FMN2POPT (for FM/DB2), and FMN3POPT (for FM/CICS). For information about the SEC option, see "SEC" on page 319. For more information about the protected functions, see "Unprotected functions and

profile names for protected functions" on page 44. For a list of functions that are protected by this method, see Table 6 on page 45, "Customizing to protect update functions in FM/DB2" on page 129, and "Customizing to protect update functions in FM/CICS" on page 251.

If you do not specify SEC=YES in your options modules, then no checking of these facility classes is done.

You can also give or deny some, or all, users access to an individual File Manager function. The following examples illustrate this.

- To give universal access of NONE to a group of functions (for example, disk input functions), enter a RACF command similar to this:

  ```
  RDEFINE FACILITY FILEM.DISK.INPUT     UACC(NONE)
  ```

  This means that no users can use any functions in the group unless otherwise specified.
- To give all users access to a group of functions (for example, tape input functions), enter a RACF command similar to this:

  ```
  RDEFINE FACILITY FILEM.TAPE.INPUT     UACC(READ)
  ```
- To give a user (with userid *userid*) access to a group of functions (for example, tape output functions), enter a RACF command similar to this:

  ```
  PERMIT FILEM.TAPE.OUTPUT    CLASS(FACILITY) ID(userid) ACCESS(READ)
  ```

  Similarly, to deny a user access to tape output functions, enter a RACF command similar to this:

  ```
  PERMIT FILEM.TAPE.OUTPUT    CLASS(FACILITY) ID(userid) ACCESS(NONE)
  ```

  The PERMIT statement for FILEM.TAPE.OUTPUT overrides the universal access that you specified for FILEM.TAPE.OUTPUT.
- To give a user access to a specific function (for example, the VSAM to Tape function), enter a RACF command similar to this:

  ```
  PERMIT FILEM.FUNCTION.VT    CLASS(FACILITY) ID(userid) ACCESS(READ)
  ```

  Similarly, to deny a user access to the VT function, enter a RACF command similar to this:

  ```
  PERMIT FILEM.FUNCTION.VT    CLASS(FACILITY) ID(userid) ACCESS(NONE)
  ```

  The PERMIT statement for FILEM.FUNCTION.VT overrides any access that you specified for FILEM.TAPE.OUTPUT.
- To give a user (with userid *userid*) permission to update a load module, enter a RACF command similar to this:

  ```
  PERMIT FILEM.FUNCTION.LMU   CLASS(FACILITY) ID(userid) ACCESS(READ)
  ```

  The PERMIT statement for FILEM.FUNCTION.LMU overrides any universal access that you specified for FILEM.LOADMOD.UPDATE.
- If the FACILITY class is not already active on your system, enter the following RACF commands to activate it:

  ```
  SETROPTS CLASSACT(FACILITY)
  SETROPTS GENERIC(FACILITY)
  SETROPTS GENCMD(FACILITY)
  ```

Controlling access to File Manager functions with SAF

SAF controls access to File Manager functions as follows:

- If access to the profile FACILITY(FILEM.FUNCTION.*fc*) in the FACILITY class is defined (where *fc* is the function code), this controls access to the function.

- If access to the profile FACILITY(FILEM.FUNCTION.*fc*) in the FACILITY class is not defined, the profile name shown in Table 8 on page 46 (in the form FILEM.*group.name*) is used.
- If no profile name as shown in Table 8 is defined, then FILEM.OTHER.ALL is used. If this does not permit access then access is denied.

Some File Manager functions are protected, by default, by the FILEM.OTHER.ALL profile. These functions are listed in Table 7 on page 45.

Important information for users of non-IBM security products

File Manager issues a RACROUTE TYPE=AUTH for the required profile, and if the required profile is not defined, RACF returns RC=4. If RC=4 is returned, File Manager issues a RACROUTE for FILEM.OTHER.ALL. If a higher return code is returned, no RACROUTE is issued for FILEM.OTHER.ALL and the access request fails immediately.

You should be aware that not all non-IBM security products issue RC=4 when a RACROUTE TYPE=AUTH for a profile fails; in this case no RACROUTE is issued for FILEM.OTHER.ALL and the access request fails immediately. If this applies to your security product, you will have to provide individual profiles for all File Manager functions listed in Table 7.

ALTER, UPDATE or READ access means that the user can use the function. Access NONE means that the user cannot use the function.

This is illustrated in Figure 3.

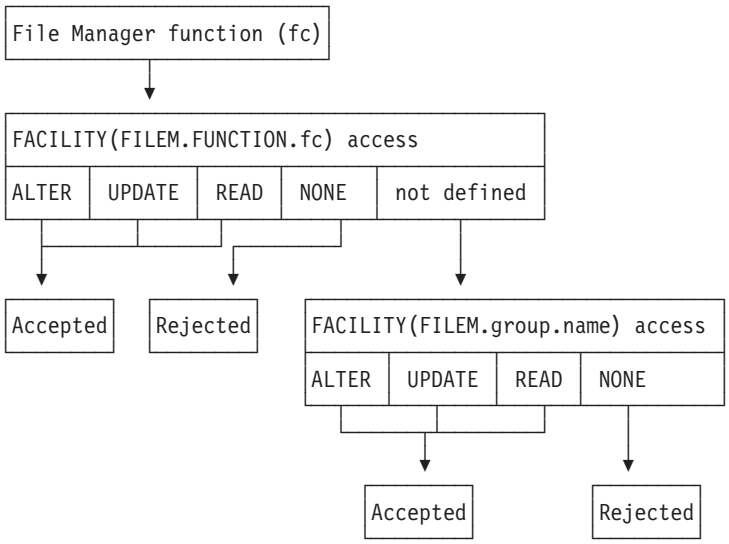


Figure 3. Access to File Manager functions

For example, the TP function is part of the FILEM.TAPE.INPUT group. You can control access to the TP function in any of the following ways:

- To give a user access to the TP function, regardless of the user's access to FILEM.TAPE.INPUT, give the user ALTER, UPDATE, or READ access to FACILITY(FILEM.FUNCTION.TP).

- To prevent a user from using the TP function, regardless of the user's access to FILEM.TAPE.INPUT, give the user NONE access to FACILITY(FILEM.FUNCTION.TP).
- To give a user access to any tape input function, unless overridden by a FILEM.FUNCTION.*fc* entry, give the user ALTER, UPDATE, or READ access to FACILITY(FILEM.TAPE.INPUT).
- To prevent a user from using any tape input function, unless overridden by a FILEM.FUNCTION.*fc* entry, give the user NONE access to FACILITY(FILEM.TAPE.INPUT).

Controlling fullpack access to DASD volumes

File Manager provides the ability to work with the entire disk volume (disk fullpack). To do this specify a File Manager disk function without specifying a data set name.

Access to disk fullpack processing is controlled as follows:

- If File Manager is not running APF-authorized, disk fullpack processing is unavailable.
- If File Manager is running APF-authorized, the user's access to the profile FILEM.DISK.FULLPACK in the FACILITY class is checked. The following access levels are possible:

  ALTER Read and update access to all volumes
  UPDATE Read access to all volumes, update access to specific volumes
  READ Read and update access to specific volumes
  NONE No fullpack access.

  Note: Since File Manager cannot run APF-authorized under ISPF, and since disk fullpack processing is not available if File Manager is not running APF-authorized, it follows that disk fullpack processing is not available under ISPF.

  Access to specific volumes is controlled with the DASDVOL class. The user needs READ access for disk read functions and ALTER access for disk update functions.

Figure 4 on page 36 and Figure 5 on page 36 show how this works for disk read and disk update functions.

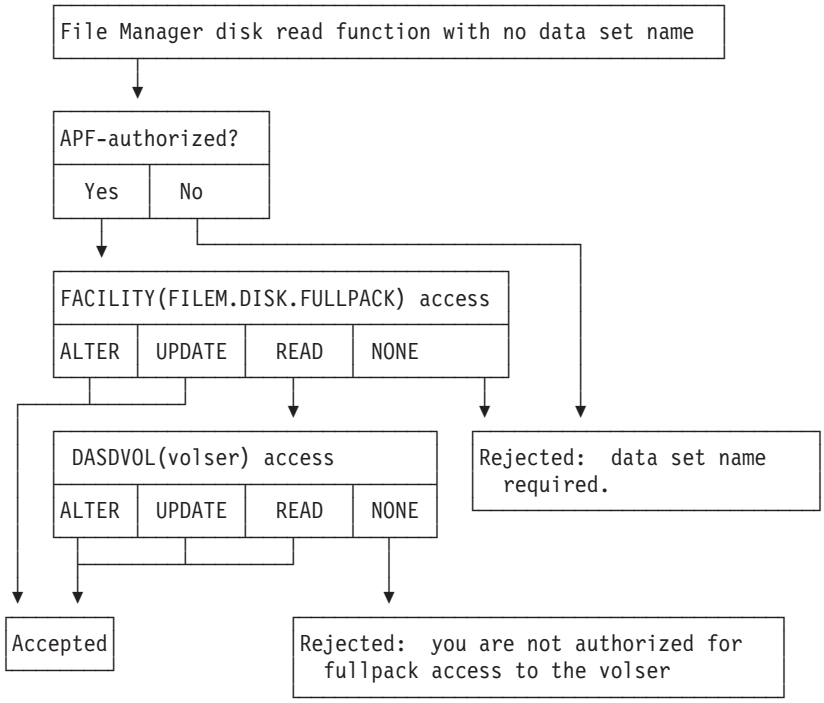


Figure 4. Fullpack processing for disk read functions

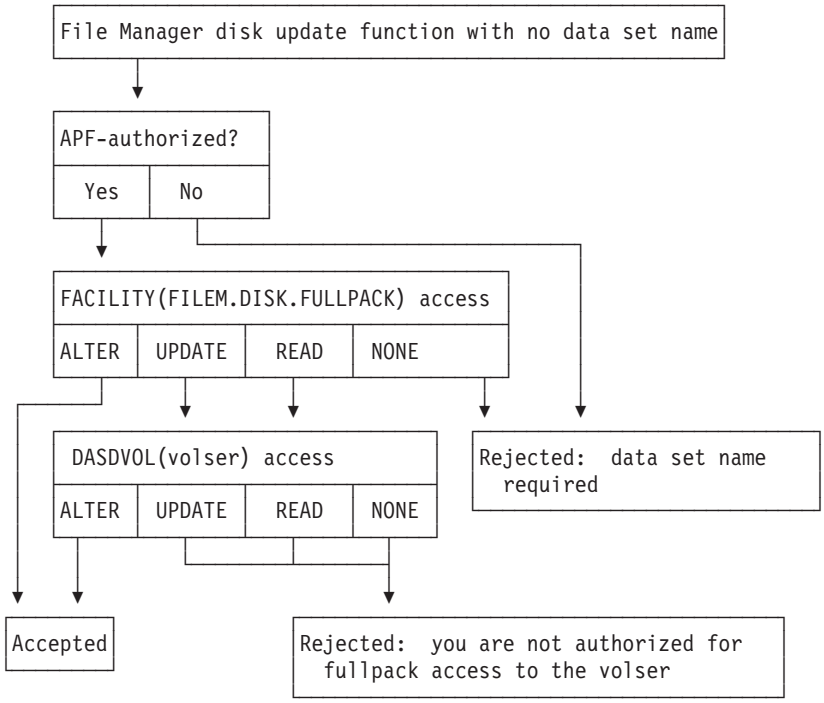


Figure 5. Fullpack processing for disk update functions

Controlling Bypass Label Processing (BLP)

You can use File Manager to control tape label processing by your users even if your system installation parameters (JES parameters) would not allow BLP.

If your installation allows BLP usage (as specified in your JES parameters), normal open processing checks if the user has access to ICHBLP. Any user with READ access or greater can use BLP. If ICHBLP is not defined to your security product, all users can use BLP.

If your installation does not allow BLP usage (as specified in your JES parameters), then if LABEL=(,BLP) is coded on the JCL control statement, BLP is converted to NL. However, File Manager users can still use BLP subject to the following conditions:

- File Manager must be running APF-authorized.
- For any function other than TLB, the user must have access to FILEM.TAPE.BLP.
- If ICHBLP is defined to your security product, the user must have READ access or greater to it.

In this case, if either of the following is coded on the JCL control statement:

- LABEL=(,BLP)
- LABEL=(,NL)

the user can specify LABEL=BLP on the first File Manager function that uses a tape. This means that the user wants to use bypass label processing (BLP) for the tape.

This is illustrated in Figure 6 on page 38.

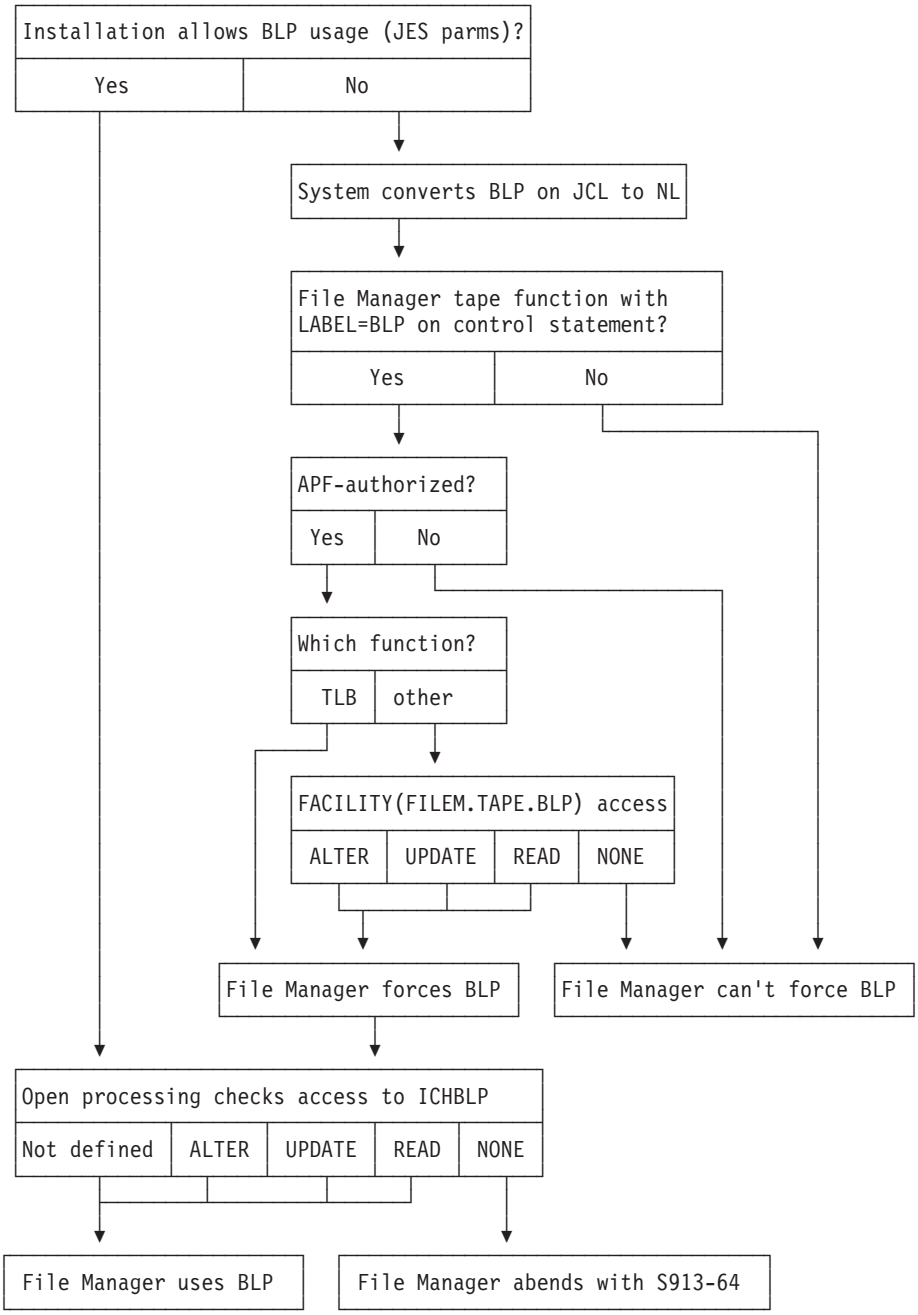


Figure 6. BLP processing by File Manager

Setting up the security environment by using FMNSECUR

A security exit is provided for security control from within File Manager if you do not use RACF or equivalent security product, or if SAF is not active at your installation. This exit is called FMNSECUR and is distributed in FMN.SFMNSAM1. You use FMNSECUR to protect selected File Manager functions from unauthorized users.

If you want to use FMNSECUR for security, customize it as described below and install it using the usermod FMNUMODS. For information on installing FMNUMODS, see "Installing FMNSECUR using FMNUMODS" on page 43.

Copy FMNSECUR from FMN.SFMNSAM1 to your own source library and edit it there. By default, in the supplied FMNSECUR, no functions are protected. This means that any user can use any File Manager function unless otherwise specified. You can override this as follows:

- Give or deny some users (or all users) access to one of the following groups of File Manager functions:

FILEM.DISK.INPUT	Disk input functions
FILEM.DISK.UPDATE	Disk update functions
FILEM.TAPE.INPUT	Tape input functions
FILEM.TAPE.OUTPUT	Tape output functions
FILEM.TAPE.DUPLICATE	Tape copy functions
FILEM.TAPE.UPDATE	Tape update functions
FILEM.VSAM.UPDATE	VSAM update functions
FILEM.OAM.OUTPUT	OAM output functions
FILEM.OAM.UPDATE	OAM update functions
FILEM.LOADMOD.UPDATE	Load module update functions
FILEM.OTHER.ALL	All other functions
FILEM.TAPE.BLP	See "Controlling Bypass Label Processing (BLP)" on page 37
FILEM.DISK.FULLPACK	See "Controlling fullpack access to DASD volumes" on page 35

  For more information about these groups, see Table 8 on page 46.
- Give or deny some users (or all users) access to an individual File Manager function.

Three facility groups are provided to allow you to control access to File Manager base function, and to FM/IMS and FM/DB2, from the FM/CICS primary option menu. These groups are:

FILEM.CICS.BASE	Access to File Manager base function
FILEM.CICS.IMS	Access to FM/IMS
FILEM.CICS.DB2	Access to FM/DB2

If a userid running FM/CICS has read access to any of these groups, then the associated function (FM, FM/IMS or FM/DB2) will appear on the FM/CICS primary option menu and the user can invoke these functions, if they are installed.

If you have installed and customized the FM/CICS component, you should review your requirement for this access.

For more information about FM/CICS, see Part 4, "Customizing File Manager CICS Component," on page 229, and also the *File Manager User's Guide and Reference for CICS*.

Three facility groups are also provided to enable you to protect update functions in File Manager base function, FM/DB2, and FM/CICS. They are:

FILEM.BASE.UPDATE	Protect update functions in File Manager base
FILEM.DB2.UPDATE	Protect update functions in FM/CICS
FILEM.CICS.UPDATE	Protect update functions in FM/CICS

(This aspect of security is handled differently for FM/IMS, see "Controlling access to IMS subsystems and FM/IMS functions" on page 191.)

These facility classes also require the option SEC=YES to be specified in FMN0POPT (for File Manager base), FMN2POPT (for FM/DB2), and FMN3POPT (for FM/CICS). For information about the SEC option, see "SEC" on page 319. For more information about the protected functions, see "Unprotected functions and profile names for protected functions" on page 44. For a list of functions that are protected by this method, see Table 6 on page 45, "Customizing to protect update functions in FM/DB2" on page 129, and "Customizing to protect update functions in FM/CICS" on page 251.

If you do not specify SEC=YES in your options modules, then no checking of these facility classes is done.

You modify FMNSECUR for your requirements by supplying FMNS macro statements to provide the control you want. Refer to the prolog in the supplied sample FMNSECUR for information about how FMNSECUR processes the FMNS macro statements, and where to insert the statements in the FMNSECUR source. The syntax of the FMNS macro is described in "Syntax of the FMNS macro" on page 41.

The following examples show how to use FMNSECUR to give or deny a user access to a group of functions or a specific function.

- To give a user access to a group of functions (for example, tape output functions), add statements similar to this to FMNSECUR:

  ```
  FMNS CLASS=FACILITY,
       ENTITY=FILEM.TAPE.INPUT,
       ACCESS=READ,
       USERID=userid
  ```

  Similarly, to deny a user access to tape output functions, add statements similar to this:

  ```
  FMNS CLASS=FACILITY,
       ENTITY=FILEM.TAPE.INPUT,
       ACCESS=NONE,
       USERID=userid
  ```

- To give a user access to a specific function (for example, the VSAM to Tape function), add statements similar to this:

  ```
  FMNS CLASS=FACILITY,
       ENTITY=FILEM.FUNCTION.VT,
       ACCESS=READ,
       USERID=userid
  ```

  Similarly, to deny a user access to the VT function, add statements similar to this:

  ```
  FMNS CLASS=FACILITY,
       ENTITY=FILEM.FUNCTION.VT,
       ACCESS=NONE,
       USERID=userid
  ```

When a user tries to use a File Manager function, FMNSECUR is called (once) with both the profile name shown in Table 8 on page 46 (in the form FILEM.*group.name*) and the function code. FMNSECUR reads through the list of FMNS macros until a match is found for the user name or job name, and either the profile name or the function code. The first match found is used.

This means that each FMNS macro effectively overrides any FMNS macro that appears after it in the file. If you want FMNSECUR to have the same behaviour as SAF (where function code specifications override profile name specifications), put all of your function code specifications before your profile name specifications.

Using FMNSECUR to protect DASD volumes from fullpack access

When you use FMNSECUR to control access to individual functions or groups of functions, you specify an access type of READ or NONE. For FILEM.DISK.FULLPACK only, you can also specify ALTER or UPDATE. The access type for FILEM.DISK.FULLPACK has the following meaning:

ALTER Read and update access to all volumes
UPDATE Read access to all volumes, update access to specific volumes
READ Read and update access to specific volumes
NONE No fullpack access.

If you give some users UPDATE or READ access to FILEM.DISK.FULLPACK, you can also specify which disk volumes the user has access to. The following examples illustrate this.

- To let a user access a disk volume with fullpack read and fullpack update functions, add statements similar to this to FMNSECUR:

  ```
  FMNS CLASS=DASDVOL,
       ENTITY=volser,
       ACCESS=ALTER,
       USERID=userid
  ```

  where *volser* is the volume serial of the disk volume.
- To permit a user to access a disk volume with fullpack read functions but not fullpack update functions, add statements similar to this to FMNSECUR:

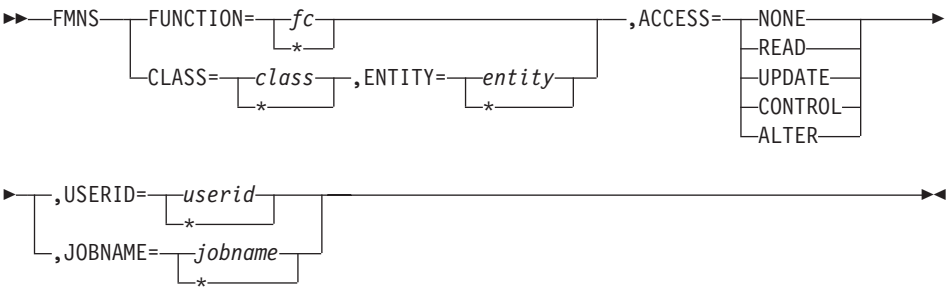
  ```
  FMNS CLASS=DASDVOL,
       ENTITY=volser,
       ACCESS=READ,
       USERID=userid
  ```

  You could also use ACCESS=UPDATE, which has the same effect as ACCESS=READ in this case.
- To prevent a user from accessing a disk volume with fullpack read or fullpack update functions, add statements similar to this to FMNSECUR:

  ```
  FMNS CLASS=DASDVOL,
       ENTITY=volser,
       ACCESS=NONE,
       USERID=userid
  ```

Syntax of the FMNS macro

Syntax



The parameters of FMNS are described below:

FUNCTION To protect a function, specify FUNCTION=*fc* (where *fc* is the function code). This is equivalent to specifying CLASS=FACILITY, ENTITY=FILEM.FUNCTION.*fc*.

CLASS To protect a group of functions, specify CLASS=FACILITY. To protect a DASD volume from fullpack access, specify CLASS=DASDVOL.

ENTITY To protect a group of functions, specify ENTITY=FILEM.*group.name* (see Table 8 on page 46 for the group names to use). To protect a DASD volume from fullpack access, specify ENTITY=*volser*.

ACCESS For all profiles except fullpack access, specify ACCESS=NONE to deny access, or any other value (READ, UPDATE, CONTROL, or ALTER) to give access. To protect fullpack access to DASD volumes, you can also use UPDATE and ALTER to grant read or update access to specific volumes. For more information, see "Using FMNSECUR to protect DASD volumes from fullpack access" on page 41.

USERID Specify either a userid or an asterisk (*) to indicate all users who have not previously been specified for this entity. If you specify USERID, do not specify JOBNAME.

JOBNAME Specify either a File Manager job name or an asterisk (*) to indicate all jobs that have not previously been specified for this entity. If you specify JOBNAME, do not specify USERID.

You can control access based on userid or job name. In batch mode, the job name is passed to FMNSECUR and the userid is also passed to FMNSECUR if it is available. FMNSECUR reads through the list of FMNS macros until a match is found for the userid or job name, and either the profile name or the function code. The first match found is used.

Exit routine environment

The following restrictions apply to this exit:

- It must be named FMNSECUR.
- It must reside in LPA (that is, within MLPA, FLPA, PLPA, EPLPA, EFLPA or EMLPA), and must therefore be reentrant. The usermod FMNUMODS installs FMNSECUR into an LPA library.
- If it is located below 16M, it is called in AMODE 24; otherwise it is called in AMODE 31.
- It will be APF-authorized only if File Manager is running APF-authorized.

Registers at entry

The contents of the registers on entry to this exit are:

Register	Contents
0	(unpredictable)
1	Address of the parameter list passed to the exit
2–12	(unpredictable)
13	Register save area
14	Return address
15	Entry point address of the exit

Parameter list contents

Register 1 points to a parameter list, which contains the following fields:

1. Pointer to an 8-character security-class string that has one of the following values:
   DASDVOL Checks a user's authority to access a DASD volume with disk fullpack processing.
   FACILITY Checks a user's authority to use a File Manager function, or a FULLPACK or BLP operation.
   TERMINAT Requests cleanup processing by the exit.
2. Pointer to a 44-character entity string:

   If parameter 1 is DASDVOL, parameter 2 is the volser.

   If parameter 1 is FACILITY, parameter 2 is the profile name shown in Table 8 on page 46.
3. Reserved.
4. Pointer to an 8-character access string. The string has the value READ, UPDATE, CONTROL, or ALTER, as described in *z/OS Security Server RACF Command Language Reference*.
5. Pointer to an 8-character function code (the name of a File Manager function).
6. Pointer to a 2-byte flags field, consisting of the following 16 bits:
   0 Batch mode
   1 Full-screen mode
   2 Line mode
   3 Command mode
   4 (reserved)
   5 XA environment
   6 ESA environment
   7–15 (reserved)
7. Pointer to an 8-character userid.
8. Pointer to an 8-character job name.
9. Reserved.
10. Pointer to a 1-fullword user field.

    This fullword is not used by File Manager. The exit can use this fullword to remember information (such as an address) from one call to another.

    File Manager initializes this fullword to binary zero at first invocation of the exit.

Registers at exit

Upon return from the exit, the register contents must be:

Register	Contents
1–14	Restored to their contents at entry.
15	A return code: 0 if the user is authorized for the resource; any nonzero value if the user is not authorized.

Installing FMNSECUR using FMNUMODS

You install FMNSECUR using the usermod FMNUMODS. FMNUMODS is distributed in FMN.SFMNSAM1. To do this:

1. Copy the FMNSECUR member from FMN.SFMNSAM1 into your own source library.

2. Modify the FMNSECUR member in your library, by coding the required FMNS macro statements.
3. Modify the FMNUMODS member in FMN.SFMNSAM1 to meet your site's requirements. This usermod will install FMNSECUR into an LPA library. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMNUMODS.

Unprotected functions and profile names for protected functions

Table 6 on page 45, Table 7 on page 45, and Table 8 on page 46, show File Manager functions with their descriptions and (where applicable) the equivalent File Manager panel you can use under ISPF.

Table 6 lists the functions that cannot be protected by RACF or an equivalent security product, or by the FMNSECUR exit.

However there are exceptions to this. Some functions listed in Table 6 on page 45 can be protected in File Manager base function by the facility class FILEM.BASE.UPDATE when the option SEC=YES is specified in FMN0POPT. These functions are indicated in Table 6 on page 45 by ✓ in the fourth column.

Similarly these functions (and others) can be protected in FM/CICS by the facility class FILEM.CICS.UPDATE, when SEC=YES is specified in FMN3POPT. For more information about protecting the CICS functions, see “Customizing to protect update functions in FM/CICS” on page 251.

The normal security checking of FILEM.FUNCTION.*function_code* applies to these functions when SEC=YES has been specified in FMN0POPT, FMN2POPT, or FMN3POPT.

Functions not listed in Table 6 are protected, and Table 8 shows the profile name used in the FACILITY class to protect a specific function.

All protected functions not listed in Table 8 are protected by the FILEM.OTHER.ALL profile.

In particular, the File Manager functions listed in Table 7 are protected by default when you receive File Manager. They are protected by the profile "FILEM.OTHER.ALL", and if you want to use these functions, you must first give the users access to these functions, by defining the profile "FILEM.OTHER.ALL" to RACF or your equivalent security product.

For information on File Manager profiles, and defining them to your security product, see “Setting up the security environment by using RACF or an equivalent security product” on page 31. In particular, if you use a non-IBM security product, see the warning in “Controlling access to File Manager functions with SAF” on page 33.

Note: In Table 8, where more than one function code is listed for the same function, and the first function code is in **bold** letters, you should use this function code when defining the relevant function profile.

Table 6. File Manager unprotected functions

Function	Equivalent panel	Description	Protect with SEC=YES
AF		Template Workbench	
APB	3.7	AFP Print Browse	
AUD		Print Audit Trail	
BTB		Batch template build	
BTU		Batch template update	
CLM	3.11.2	Load module compare	
DSC, COPY	3.3	Copy Utility	✓
DSE	2	Data Set Edit	✓
DSEB		Data Set Edit (batch)	✓
DSG	3.1	Data Create Utility	✓
DSI	3.4 I	Data Set Information	
DSM, DSCMP	3.11.1	Data Set Compare	
DSP	3.2	Print Utility	
DSU		Data Set Update (batch)	✓
DSV	1	Data Set View	
DSX	5.6	Data Set Extents	
DVT	3.5	Display VTOC	
DX		Decimal to hex conversion	
FCH	3.6	Find/Change Utility	✓
FMT	3.0	Set DBCS Format	
LVL		Show File Manager Release and PTF Level	
MB, TST	3.8	Memory Browse	
NEW, NEWS		Show File Manager Release News	
NODF, NOSORT		Do not use DFSORT	
PB	3.9	Print Browse	
PBK		Print/view copybook	
SCS	3.4	Catalog Services	
SET		Set File Manager Processing Options	
VER		Show File Manager Release and PTF Level	
VLM		View Load Module	
XD		Hex to decimal conversion	
(none)		Template Functions	

Table 7. File Manager functions protected by FILEM.OTHER.ALL

Function	Equivalent panel	Description	Profile
BSF	4.15.1	Backward space file (tape)	FILEM.OTHER.ALL
BSR	4.15.3	Backward space record (tape)	FILEM.OTHER.ALL
DVB	5.8	Data in virtual browse	FILEM.OTHER.ALL
FSF	4.15.2	Forward space file (tape)	FILEM.OTHER.ALL

Table 7. File Manager functions protected by FILEM.OTHER.ALL (continued)

Function	Equivalent panel	Description	Profile
FSR	4.15.4	Forward space record (tape)	FILEM.OTHER.ALL
OB	6.2	Object browse	FILEM.OTHER.ALL
ODL	6.1	Object directory list	FILEM.OTHER.ALL
OP	6.3	Object print	FILEM.OTHER.ALL
OQ, OS	6.6.2	Object to sequential data	FILEM.OTHER.ALL
OV	6.6.1	Object to VSAM	FILEM.OTHER.ALL
REW	4.15.5	Rewind (tape)	FILEM.OTHER.ALL
RUN	4.15.6	Rewind and unload (tape)	FILEM.OTHER.ALL
VX		VSAM to REXX stem	FILEM.OTHER.ALL
XV		REXX stem to VSAM	FILEM.OTHER.ALL

Table 8. File Manager function to profile name cross-reference

Function	Equivalent panel	Description	Profile
(none)		Using BLP for tape processing	FILEM.TAPE.BLP
(none)		Handling full disk packs	FILEM.DISK.FULLPACK
BT	4.7	Create Tape File	FILEM.TAPE.OUTPUT
DB	5.1	Disk Browse	FILEM.DISK.INPUT
DCN		Disk to console	FILEM.DISK.INPUT
DP	5.3	Disk Print	FILEM.DISK.INPUT
DRS	5.4	Disk Record Scan	FILEM.DISK.INPUT
DTE	5.2	Disk Track Edit	FILEM.DISK.UPDATE
EOF	5.5	Write EOF Record	FILEM.DISK.UPDATE
ERT	4.13	Erase Tape	FILEM.TAPE.UPDATE
EVC	4.2.8	Exported Stacked Volume Copy	FILEM.TAPE.DUPLICATE
EVL	4.14	Exported Stacked Volume List	FILEM.TAPE.INPUT
HT		HFS to tape	FILEM.TAPE.OUTPUT
INT	4.12	Initialize Tape	FILEM.TAPE.UPDATE
LMU	2	Load module edit/update	FILEM.LOADMOD.UPDATE
OE	6.5	Object Erase	FILEM.OAM.UPDATE
OO	6.6.5	Object to Object	FILEM.OAM.OUTPUT
OU	6.4	Object Update	FILEM.OAM.UPDATE
QO, SO, TLO	6.6.4	Sequential Data to Object	FILEM.OAM.OUTPUT
QT, ST, STP	4.2.7	Sequential Data to Tape	FILEM.TAPE.OUTPUT
TB	4.1	Tape Browse	FILEM.TAPE.INPUT
TC		Tape to card	FILEM.TAPE.INPUT
TCN		Tape to console	FILEM.TAPE.INPUT
TDL		Tape data set list	FILEM.TAPE.INPUT
TH		Tape to HFS	FILEM.TAPE.INPUT
TLB	4.8	Tape Label Display	FILEM.TAPE.INPUT

Table 8. File Manager function to profile name cross-reference (continued)

Function	Equivalent panel	Description	Profile
TLT	4.2.3	Tape to Labeled Tape	FILEM.TAPE.DUPLICATE
TMP	4.6	Tape Map	FILEM.TAPE.INPUT
TP	4.5	Tape Print	FILEM.TAPE.INPUT
TQ, TS, TSQ	4.2.5	Tape to Sequential Data	FILEM.TAPE.INPUT
TRL	4.4	Tape Record Load	FILEM.TAPE.UPDATE
TRS	4.10	Tape Record Scan	FILEM.TAPE.INPUT
TT	4.2.1	Tape to Tape (copy)	FILEM.TAPE.DUPLICATE
TTC	4.9	Tape to Tape Compare	FILEM.TAPE.INPUT
TTR	4.2.2	Tape to Tape Reblocked	FILEM.TAPE.DUPLICATE
TU	4.3	Tape Update	FILEM.TAPE.UPDATE
TV, TVS	4.2.4	Tape to VSAM	FILEM.TAPE.INPUT
TX		Tape to REXX Variable	FILEM.TAPE.INPUT
VO	6.6.3	VSAM to Object	FILEM.OAM.OUTPUT
VRU		Update VSAM record	FILEM.VSAM.UPDATE
VT, VTP	4.2.6	VSAM to Tape	FILEM.TAPE.OUTPUT
VU	5.7	VSAM Update	FILEM.VSAM.UPDATE
WTM	4.11	Write Tape Mark	FILEM.TAPE.UPDATE
XT		REXX Variable to Tape	FILEM.TAPE.OUTPUT
Others		Protects all functions not otherwise listed in Table 6 or this table.	FILEM.OTHER.ALL

Chapter 5. Customizing the File Manager audit facility (FMN0POPT controlled auditing)

You can use the audit logging facilities of File Manager to create an audit trail of editing activity on your data sets. You can create your audit trail either using SMF or recording the activity to an audit log data set.

File Manager determines whether or not an audit trail is produced, and whether it will be written to SMF or to an audit log data set, by the settings of the AUDITLOG and SMF options in the FMN0POPT macro.

Setting the appropriate options to produce an audit trail

If you intend to produce an audit trail there are some File Manager options you need to set in FMN0POPT. For information about changing the options in FMN0POPT see "Changing the default options" on page 23. For information about the options see Appendix A, "File Manager options," on page 289.

- If you want to produce an audit trail, specify AUDITLOG=YES in FMN0POPT.
- If you want to produce an audit trail and report on the changes made at conclusion of an edit function then specify AUDITLOG=DEMAND. The job submitted will be determined by skeleton member FMN0FTAD found in FMN.SFMNSLIB. You should customize the job card and JCL to specify the reporting options you require. For changing the audit report options in the skeleton see "AUD (Print Audit Trail Report)" in the *File Manager User's Guide and Reference*.
- If you want to write your audit trail to SMF, specify an SMF record number between 128 and 255 in FMN0POPT. If you do not specify an SMF record number in FMN0POPT then your audit trail will be written to a data set.
- However, if you do **not** want to write your audit trail to SMF, then make sure you specify SMFNO=0 in FMN0POPT.
- If you intend to write your audit trail to a data set, optionally specify a value for the AUDITHLQ option in FMN0POPT.

Using System Management Facilities (SMF) for audit logging

If you intend to use SMF for audit logging, you must also do the following:

- Ensure that the load module FMNSMF is APF-authorized. You can make FMNSMF APF-authorized either by authorizing the load library, FMN.SFMNMOD1, or by copying FMNSMF to another authorized library. For more information about authorizing FMN.SFMNMOD1, see "Running File Manager with APF-authorization" on page 21.
- Add the load module FMNSMF to the AUTHTSF list in member IKJTSO*xx* in SYS1.PARMLIB. If you do not do this, even if you have selected to record to SMF and you have specified an SMF record number, no recording will be done.
- Select an SMF record number between 128 and 255 for the audit log records, and include it in your SMF parmlib member SMFPRM*xx*.
- Specify this SMF record number in FMN0POPT.

© Copyright IBM Corp. 2000, 2011

To activate any changes you have made to SYS1.PARMLIB members, either restart your system, or use the appropriate commands for your site to dynamically activate the changes.

For more information about SMF, see *z/OS MVS System Management Facilities (SMF)*.

To report on the audit trail information collected by SMF, you must extract this information from SMF to your own data set. The information in this data set can then be printed by the File Manager Print Audit Trail utility. To do this select option 3.12 from the File Manager Primary Option Menu.

A sample job, FMNSMFX, is provided in FMN.SFMNSAM1 to help you extract the SMF data to your own data set. See the comments in the job for information about changes you need to make to the job.

Recording an audit trail to a data set

If you choose to record your audit trail to a data set, you can print this information using the File Manager Print Audit Trail utility. To do this select option 3.12 from the File Manager Primary Option Menu.

The format of the audit log data set name is determined by the setting of the AUDITHLQ parameter in the FMN0POPI definition in FMN0POPT. See "AUDITHLQ" on page 291 for more information about the AUDITHLQ option.

The following data set name formats may be generated:

`userid.FMNLOG.Dyymmdd.Thhmmss`	(when AUDITHLQ= (blank))
audithlq`.userid.FMNLOG.Dyymmdd.Thhmmss`	(when AUDITHLQ=*audithlq*)
`qual1.<qual2.><qual3.>Dyymmdd.Thhmmss`	(when AUDITHLQ=*qual1.<qual2.><qual3>*)

where:

audithlq	Is any 1-8 character constant that is valid in the context of a data set name
userid	Is the userid creating the data set
Dyymmdd	Is the date of the activity
Thhmmss	Is the time of the activity

When AUDITHLQ contains one or more periods, the AUDITHLQ value is treated as a data set prefix, with one, two or three levels. Each level of the prefix can be:

XXX	Any 1-8 character constant that is valid in the context of a data set name.
&&PREFIX	Indicates that the user's TSO prefix should be used. This will be null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix will also be null.
&&USER	Indicates that the user's logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.
&&UID	Indicates that the user's TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user's TSO logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.
&&FUNCOD	Indicates that the File Manager internal function code should be used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.

Chapter 6. Customizing the File Manager audit facility (SAF-rule controlled auditing)

Controlling File Manager auditing using SAF

You can use SAF to control the following aspects of File Manager auditing:

- Whether File Manager writes audit records to SMF, the user's audit log data set, or both.
- Whether File Manager writes audit records when accessing resources.
- Whether File Manager writes audit records only for update-related activities, for read and update-related activities, or functional information only.
- Whether File Manager automatically produces the audit log report at the end of a File Manager session ("Demand" audit logging).

For users wanting to control auditing by means of parmlib, the users being audited must have read access or more to the SAF facility class profile FILEM.PARMLIB.BASE. In this case, auditing is controlled by the FMAUDIT parameter in the FMN0PARM member. See "File Manager options specified in FMN0PARM" on page 411 for more information.

Note: File Manager will not start if a user has read access to this facility class and the FMN0PARM member is unavailable from parmlib.

Example: User PROD1 to have auditing controlled from parmlib.

Write this RACF rule:

```
RDEF FACILITY FILEM.PARMLIB.BASE AUDIT(NONE) UACC(NONE)
        OWNER(ownerid)
PE FILEM.PARMLIB.BASE ACC(READ) ID(PROD1)  CLASS(FACILITY)
```

If SAF processing is not active, or the rule is not defined, or the rule is defined and the user has no access, then no parmlib processing is performed.

If there is no parmlib processing to specify SAF-controlled auditing, then SAF controlled auditing is performed for users that have read access to the SAF facility class profile FILEM.SAFAUDIT.BASE.

Example: User PROD2 to have SAF auditing without parmlib.

Write this RACF rule:

```
RDEF FACILITY FILEM.SAFAUDIT.BASE AUDIT(NONE) UACC(NONE)
        OWNER(ownerid)
PE FILEM.SAFAUDIT.BASE ACC(READ) ID(PROD2)  CLASS(FACILITY)
```

Note: If SAF-controlled auditing is specified without parmlib processing, then the SMF record type is obtained from options module SMFNO value.

© Copyright IBM Corp. 2000, 2011

When a security product other than RACF is in use

File Manager V11 APAR PM31842 introduced a change to remove the mandatory requirement for definition of members in the logical PARMLIB concatenation. In order to achieve this, File Manager makes RACROUTE calls, with STATUS=ACCESS, to SAF FACILITY profiles:

FILEM.PARMLIB.**
FILEM.SAFAUDIT.** (only if the first call fails)

STATUS=ACCESS is specified because the RACROUTE call should not audit the call. The documentation in the *z/OS V1R11 Security Server RACROUTE Macro Reference*, in the description of RACROUTE REQUEST=AUTH,STATUS=ACCESS, states:

ACCESS

The request is simply to return the user's highest current access to the resource specified. Upon successful completion, the user's access is returned in the RACF reason code. **No auditing is done for this request.**

When RACF is used, the STATUS=ACCESS request works as documented, and no security-related logging or abends are generated, even if you do not have access to the profile.

However, when non-RACF security products (such as ACF2) are used, S047 ABEND047 may be issued in response to the above RACROUTE request.

The ACF2 global system options (GSO) definitions shown here have been used to avoid the S047 abends:

```
SET CONTROL(GSO)
INSERT SAFDEF.APFFM PROGRAM(FMN*****) RB(FMN*****) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=FACILITY,STATUS=ACCESS)
INSERT SAFDEF.APFFM2 PROGRAM(FILEM***) RB(FILEM***) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=FACILITY,STATUS=ACCESS)
F ACF2,REFRESH(SAFDEF)
INSERT SAFDEF.APFFM PROGRAM(FMN*****) RB(FMN*****) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
INSERT SAFDEF.APFFM2 PROGRAM(FILEM***) RB(FILEM***) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
F ACF2,REFRESH(SAFDEF)
```

Implementing SAF-rule controlled auditing

Use the following checklist to implement SAF-rule controlled auditing:

__ 1. Determine the SAF FACILITY and XFACILIT rules that will be required. See "Understanding how File Manager uses SAF rules to control auditing" on page 53 for more information.

__ 2. Write the relevant SAF rules. See the examples in "SAF rule examples" on page 142.

__ 3. Modify the FMN0PARM member to activate SAF-rule controlled auditing. See "File Manager options specified in FMN0PARM" on page 411 for more information.

__ 4. Test the configuration to ensure that auditing is occurring as required. You can use the facilities for customizing the FMN0PARM definitions to, for example, turn on SAF-rule controlled auditing for a specific logonid (or logonids), without changing the behaviour for other users.

__ 5. When testing is complete, activate SAF-rule controlled auditing for all File Manager users by changing the FMN0PARM member.

Understanding how File Manager uses SAF rules to control auditing

SAF (System Authorization Facility) allows applications, such as File Manager, to define "resources" that might need to be protected. The "resource" to be protected need not be something specific, such as a data set; it can be essentially any type of resource or facility that the application considers to be important. For File Manager and auditing, the "resource" is the ability to write audit log records. The resource names reflect either the type of auditing that is to occur (eg to SMF), or the File Manager function and data set.

File Manager uses two types of SAF resource names to control auditing. The SAF resource rules used by File Manager to control auditing are shown in Table 12 on page 59 and Table 13 on page 59).

Understanding SAF rule access levels

SAF provides for five levels of access to any FACILITY or XFACILIT resource. The levels of access form a hierarchy, so that a user with the highest level of access to a resource also has access to all the lower levels. The levels of access are specified in RACF rules using the following mnemonics:

NONE No access
READ Level 1 access
UPDATE Level 2 access
CONTROL Level 3 access
ALTER Level 4 access.

It is important to understand that the mnemonics used (READ, UPDATE and so on) can and do mean different things, depending on the context in which the SAF resource name is used. This can be confusing since READ and UPDATE have obvious meanings when it comes to, for example, accessing a data set. For SAF rules used to control File Manager audit, it may aid understanding to think of the mnemonics as indicating level 1 access and level 2 access.

For the SAF resource rules used by File Manager, the meanings of the various levels of access are:

NONE
The user does not have access to the resource; this typically means the user cannot write audit log records.

READ The user has level 1 access to the resource; this typically means that the user can write audit log records.

UPDATE
The user has level 2 access to the resource. This level of access only has meaning for FACILITY rule 2 (see Table 12 on page 59). A user with level 2 access can write audit log records to the user's audit log data set, and the audit log data set will be printed at the end of the user's session (on-line execution only). This is equivalent to the DEMAND audit option in the non-SAF case.

CONTROL
The user has level 3 access to the resource. This level of access is not used by File Manager.

ALTER
The user has level 4 access to the resource. This level of access is not used by File Manager.

How File Manager determines whether audit log records should be written

The determination of whether audit records are to be written for a particular File Manager function and a given TSO logonid follows this three step process:

Step 1. What is the FMAUDIT specification in the FMN0PARM member?

The FMAUDIT specification setting in the FMN0PARM member (in SYS1.PARMLIB) is the "master" switch for SAF-rule controlled auditing. Note that there are facilities available to specify different settings in the FMN0PARM member for different TSO logonids, see “File Manager options specified in FMN0PARM” on page 411 for more information. For any given TSO logonid, there are two possibilities:

SAF_CTRL=NO

SAF-rule controlled auditing is not in effect. Auditing is determined by the settings in the FMN0POPT module, see Chapter 5, “Customizing the File Manager audit facility (FMN0POPT controlled auditing),” on page 49.

SAF_CTRL=YES

SAF-rule controlled auditing is in effect. Processing continues to step 2.

Step 2. Does the user have access to write audit records?

This is determined by the user's access to rules 1 and 2 in Table 12 on page 59; the various outcomes are summarized in Table 9.

Table 9. Determination of a user's ability to write audit log records

TODSN access[1]	TOSMF access[2]	OPTION access[3]	Can write audit records?	Demand logging?	"Create audit trail" option[4]
NONE	NONE	ANY	No	No	Not visible
READ	NONE	NONE	Yes, data set only	No	Not visible
READ	NONE	READ	Yes, data set only	No	Visible
UPDATE	NONE	NONE	Yes, data set only	Yes	Not visible
UPDATE	NONE	READ	Yes, data set only	Yes	Visible
NONE	READ	NONE	Yes, SMF only	No	Not visible
NONE	READ	READ	Yes, SMF only	No	Visible
READ	READ	NONE	Yes, to data set and SMF	No	Not visible
READ	READ	READ	Yes, to data set and SMF	No	Visible
UPDATE	READ	NONE	Yes, to data set and SMF	Yes	Not visible
UPDATE	READ	READ	Yes, to data set and SMF	Yes	Visible

Table 9. Determination of a user's ability to write audit log records (continued)

TODSN access[1]	TOSMF access[2]	OPTION access[3]	Can write audit records?	Demand logging?	"Create audit trail" option[4]

Notes:

1. Refers to the level of access the user has to SAF FACILITY rule 1 in Table 28 on page 146.
2. Refers to the level of access the user has to SAF FACILITY rule 2 in Table 28 on page 146.
3. Refers to the level of access the user has to SAF FACILITY rule 3 in Table 28 on page 146.
4. The visibility of the "Create audit trail" option does not influence whether a user can write audit log records, although the user must have access to write audit log records (to either a data set or SMF), for the option to be visible.

If the user does not have the ability to write audit log records, then no check of SAF resource names in Step 3 occurs.

A user's access to write audit log records at Step 2 only indicates that auditing might occur, the final decision depends on the user's level of access to the XFACILIT resource name (or names) that apply to the particular File Manager function.

Step 3. Does the user have access to write audit records for the current function and data set? The XFACILIT resource name(s) used by File Manager to determine whether audit records should be written depend on the File Manager function being executed and the data set being accessed.

Table 10 shows the function codes which are supported.

Table 10. File Manager function codes that can be audited using SAF

Function code	Online option	Description
DSB	BROWSE prefix command	Browse
DSV	1	View
DSE	2	Edit
DSC	3.3	Copy
DSG	3.1	Create
DSP	3.2	Print
DSM	3.11	Compare
FCH	3.6	Find/Change
DSCMP		Batch compare
DSEB		Batch data set edit
DSU		Batch data set update

These restrictions apply:

- When copying data sets and SAF XFACILIT rules have been supplied for both data sets and these rules are different, the most inclusive rule is applied. For example, if the SAF XFACILIT rule for the old data set specifies UPDATE logging, and the rule for the new data set specifies FUNCTION logging, UPDATE logging is applied.
- When comparing data sets and SAF XFACILIT rules have been supplied for both data sets and these rules are different, the most inclusive rule is applied. For example, if the SAF XFACILIT rule for the old data set

specifies ALL logging, and the rule for the new data set specifies FUNCTION logging, ALL logging is applied.

- SAF XFACILIT rules do not check for data sets which are written to using the WRITE function in a REXX procedure. The SAF XFACILIT rule supplied for the function and data sets is applied to any records written using the WRITE function.
- If a data set can be copied using DFSORT, but SAF logging is required, DFSORT is not used. If PDS members can be copied using IEBCOPY, but SAF logging is required for one or more members, IEBCOPY is not used.
- If the "Use File Manager editor" option has not been selected, and a member is selected from a selection list to be browsed, edited, or viewed, and there is a SAF logging rule which affects this function and resource, the ISPF editor is not used. Instead, the File Manager function is invoked to ensure the requested logging is performed.

Controlling where File Manager writes audit log records

You can use SAF to control whether File Manager writes audit log records to SMF, the user's audit log data set, or to both.

The following table shows the SAF FACILITY class resource names used to control File Manager logging to SMF and the user's audit log data set.

Table 11. SAF FACILITY class resource names controlling disposition of File Manager audit records

FACILITY Class Name	Purpose
FILEM.AUDIT.TOSMF	Enables or disables auditing to SMF for File Manager.
FILEM.AUDIT.TODSN	Enables or disables auditing to the user's data set for File Manager.

SAF rule examples

This section shows SAF rule examples under different conditions.

Controlling where File Manager writes audit log records

You can use SAF to control whether File Manager writes audit log records to SMF, the user's audit log data set, or to both.

Table 12 on page 59 shows the SAF FACILITY class resource names used to control File Manager to logging and the user's audit log data set.

Example 1

- Disable audit logging to a user data set for all File Manager users.
- Enable File Manager audit logging to SMF for the PROD logonid.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT.TOSMF[1]
RDEL FACILITY FILEM.AUDIT.TODSN[1]
RDEF FACILITY FILEM.AUDIT.TOSMF UACC(NONE) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT.TODSN UACC(NONE) OWNER(XXXXXXX)[3]
PE FILEM.AUDIT.TOSMF ACC(READ) ID(PROD) CLASS(FACILITY)[4]
```

Explanation:
1. Delete any existing facility rule.
2. Define the facility rule for audit logging to SMF (TOSMF suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
3. Define the facility rule for audit logging to the user's audit log data set (TODSN suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
4. Allow logonid PROD to write audit log records (ACC(READ)) to SMF.

Example 2
- Enable audit logging to a user data set for all File Manager users.
- Enable demand logging for the following users, PROD1, PROD2, PROD3.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT.TOSMF[1]
RDEL FACILITY FILEM.AUDIT.TODSN[1]
RDEF FACILITY FILEM.AUDIT.TOSMF UACC(NONE) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT.TODSN UACC(READ) OWNER(XXXXXXX)[3]
PE FILEM.AUDIT.TODSN ACC(UPDATE) ID(PROD1) CLASS(FACILITY)[4]
PE FILEM.AUDIT.TODSN ACC(UPDATE) ID(PROD2) CLASS(FACILITY)[4]
PE FILEM.AUDIT.TODSN ACC(UPDATE) ID(PROD3) CLASS(FACILITY)[4]
```

Explanation:
1. Delete any existing facility rule.
2. Define the facility rule for audit logging to SMF (TOSMF suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
3. Define the facility rule for audit logging to the user's audit log data set (TODSN suffix). UACC(READ) is used so that any user, for which there is no specific rule, has read access, and can therefore write audit log records.
4. Allow logonids PROD1, PROD2, PROD3 to write audit log records with automatic printing of the audit report ("Demand logging") (ACC(UPDATE)), to SMF.

Example 3
- Disable audit logging completely for all File Manager users.
- Enable dual logging for all File Manager users.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT.TOSMF[1]
RDEL FACILITY FILEM.AUDIT.TODSN[1]
RDEF FACILITY FILEM.AUDIT.TOSMF UACC(READ) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT.TODSN UACC(READ) OWNER(XXXXXXX)[3]
```

Explanation:
1. Delete any existing facility rule.
2. Define the facility rule for audit logging to SMF (TOSMF suffix). UACC(READ) is used so that any user, for which there is no specific rule, has access (and can therefore write audit records to SMF).
3. Define the facility rule for audit logging to to the user's audit log data set (TODSN suffix). UACC(READ) is used so that any user, for which there is no specific rule, has access (and can therefore write audit records to the user's audit log data set).

Controlling auditing of File Manager functions

You can use SAF to control whether File Manager audit log records for File Manager functions which access resources. Table 10 on page 55 shows File Manager functions codes which may be logged.

Example 1

- Enable audit logging of all modifications to data set FMN.TEST.DATA using the File Manager Edit function for all users except TSO logonid MAINT1.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT.DSE.UPDATE.FMN.TEST.DATA[1]
RDEF XFACILIT FILEM.AUDIT.DSE.UPDATE.FMN.TEST.DATA OWNER(XXXXXXXX) UACC(READ)[2]
PE FILEM.AUDIT.DSE.UPDATE.FMN.TEST.DATA CLASS(XFACILIT) ID(MAINT1) ACC(NONE)[3]
```

Explanation:

1. Delete any existing XFACILIT rule.
2. Define the XFACILIT rule to log all modifications to data set FMN.TEST.DATA using the File Manager Edit function (DSE). UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).
3. A specific rule for logonid MAINT1 to prevent audit log records being written.

Example 2

- Enable audit logging of all records which are read or modified for data set FMN.TEST.DATA using the File Manager Edit function for user SERVIC1.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT.DSE.ALL.FMN.TEST.DATA[1]
RDEF XFACILIT FILEM.AUDIT.DSE.ALL.FMN.TEST.DATA OWNER(XXXXXXXX) UACC(NONE)[2]
PE FILEM.AUDIT.DSE.ALL.FMN.TEST.DATA CLASS(XFACILIT) ID(SERVIC1) ACC(READ)[3]
```

Explanation:

1. Delete any existing XFACILIT rule.
2. Define the XFACILIT rule to log all records which are read or modified for data set FMN.TEST.DATA using the File Manager Edit function (DSE). UACC(NONE) specifies that no TSO userids write audit log records (in the absence of any over-riding more specific rule).
3. A specific rule for logonid SERVICE1 to write audit log records.

Example 3

- Enable audit logging of functional information for member MEM1 in library FMN.TEST.DATA.PDS using the File Manager Print utility for all users.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT.DSP.FUNCTION.FMN.TEST.DATA.PDS.MEM1[1]
RDEF XFACILIT FILEM.AUDIT.DSP.FUNCTION.FMN.TEST.DATA.PDS.MEM1 OWNER(XXXXXXXX) UACC(READ)[2]
```

Explanation:

1. Delete any existing XFACILIT rule.
2. Define the XFACILIT rule to log function information when member MEM1 in library FMN.TEST.DATA.PDS is printed using the File Manager Print utility (DSP). UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).

Example 4

- Enable audit logging of functional information for all access to Websphere MQ Queue FMN.TEST.QUEUE which is managed by Websphere MQ Queue Manager FMN1 for all users.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT.*.FUNCTION.FMN1:FMN.TEST.QUEUE[1]
RDEF XFACILIT FILEM.AUDIT.*.FUNCTION.FMN1:FMN.TEST.QUEUE OWNER(XXXXXXXX) UACC(READ)[2]
```

Explanation:
1. Delete any existing XFACILIT rule.
2. Define the XFACILIT rule to log function information when Websphere MQ Queue FMN1:FMN.TEST.QUEUE is accessed using any File Manager function (*). UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).

File Manager auditing FACILITY and XFACILIT class resource names

These two tables (list FACITITY and XFACILIT class resource names and details.

Table 12. File Manager auditing FACILITY class resource names

Rule Number	Resource Name[1]	Purpose
1	FILEM.AUDIT.TODSN	Allows a user to write audit log records to the user's audit log data set.
2	FILEM.AUDIT.TOSMF	Allows a user to write audit log records to SMF.
3	FILEM.AUDIT.OPTION	Allows the user access to the "Create audit trail" option on the File Manager Edit panels.

Table 13. File Manager auditing XFACILIT class resource names

Resource Name	Purpose
FILEM.AUDIT.functioncode.ALL[1].resource[2]	Allows users to write audit log records for all records which are read and modified for the specified data set (resource) using the File Manager function (function code).
FILEM.AUDIT.functioncode.UPDATE.resource	Allows users to write audit log records for all modifications to the specified data set (resource) using the File Manager function (function code).
FILEM.AUDIT.functioncode.FUNCTION.resource	Allows users to write audit log records containing information for the specified data set (resource) using the File Manager function (function code).

Notes:
1. It is recommended that this option be used with caution. The size of the data set being accessed and the editing technique used will have an impact on the number of read records logged. And this may have an impact on the performance of File Manager.
2. If a member name is to be included it must be period qualified and not enclosed in brackets.

   A Websphere Queue name must be prefixed with the Websphere MQ Queue Manager name followed by a colon.

File Manager auditing FACILITY and XFACILIT class resource names

Chapter 7. Customizing File Manager for national languages

You can customize File Manager for national languages other than English. If you are using Japanese or Korean, and you have installed the File Manager Japanese or Korean components, you might not need to perform any other customization for the Japanese or Korean national languages.

If you are using a language other than English, Japanese or Korean (for example, French), you will need to perform the customization tasks listed in Table 14. Perform **ALL** these steps if you plan to use a different language than English, Japanese or Korean.

Table 14. Summary of steps for customizing File Manager for a national language

Step	Description	Page
__ 1	Set the LANGUAGE option for batch processing, if required.	23
__ 2	Change the TERMTYPE option in FMN0POPT, if you are using a DBCS language.	322
__ 3	Confirm that you have the correct terminal type set in ISPF (ISPF option 0).	
__ 4	Create a print and display translation table for your language.	61
__ 5	Translate the File Manager message text to your language.	62
__ 6	Provide a version of FMN0MENU for your language.	63
__ 7	Translate the File Manager ISPF messages to your language.	63
__ 8	Translate the File Manager panels to your language.	63

Changing the print and display translation tables for languages other than English

If you are using a language other than English, Japanese or Korean (for example, French), you create print and display tables specific to your language. You use the usermod FMNUMODX, and the member FMNTRTBS to do this. FMNUMODX and FMNTRTBS are distributed in FMN.SFMNSAM1.

A translation table is also provided for German. This is FMNTRDEU. You can modify this, if necessary, using the usermod FMNUMODG, distributed in FMN.SFMNSAM1.

DBCS languages
This step is essential if you are using a DBCS language other than Japanese or Korean.

You do this as follows:

1. Check that the terminal on which you want to display File Manager panels supports the display of special characters, or that the universal character buffer (UCB) of your printer has the characters you want to use.
2. Change the File Manager options to specify PRTTRANS=ON. See "Changing the default options" on page 23 for information on how to do this.

© Copyright IBM Corp. 2000, 2011

3. If you are using any DBCS language, change the File Manager options to specify TERMTYPE=3270KN.
4. Create a multicultural support version of the translation table as follows:
   a. Copy the FMNTRTBS member from FMN.SFMNSAM1 into your own source library, with the name FMNTR*yyy*, where *yyy* is one of the following language codes:

FRA	French
DEU	German
ITA	Italian
JPN	Japanese
PTG	Portuguese
ESP	Spanish
DAN	Danish
ENP	Upper case English
KOR	Korean
DES	Swiss German
CHT	Traditional Chinese
CHS	Simplified Chinese
XXX	Other

   b. Change the translation table definition statements in your source member, FMNTR*yyy*, in your source library, as required.

      File Manager provides support under ISPF for command verbs and command keywords entered in lower or uppercase as long as they are valid SBCS-only strings, and the a - z codepoints in the host codepage used correspond to the standard EBCDIC a-z codepoints. If this is not the case, and you want to be able to enter lower or mixed case command verbs and command keywords, then you will need to modify the TRUPC translation table in FMNTRTBS, to specify the uppercase A - Z EBCDIC codepoints at locations corresponding to the a - z codepoints in the codepage used.
   c. Modify the FMNUMODX member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
   d. Install SMP/E usermod FMNUMODX.

Translating the message text

All File Manager base function messages are stored in the FMN0MENU source member. This CSECT is part of the root module so that an English version of the messages is always available. In addition, all messages used by File Manager under ISPF are provided in the library, FMN.SFMNMENU. Using both FMN0MENU and members in FMN.SFMNMENU, you can provide your own set of translated messages in another language. To use the messages you have translated, see “Using the translated messages and panels” on page 64.

To provide translated versions of the messages, you must provide a version of FMN0MENU for your language, as described in “Providing a national language version of FMN0MENU” on page 63, **and** provide translated versions of the appropriate members in FMN.SFMNMENU, as described in “Translating the ISPF messages text” on page 63. (You should not need to provide a Japanese or a Korean version of the messages if you have installed the File Manager Japanese or Korean components.)

Providing a national language version of FMN0MENU

FMN0MENU contains the assembler source for the messages. To provide translated versions of the messages:

1. Copy the member FMN0MENU from FMN.SFMNSAM1 to your own source library with the name FMN0M*yyy*, where *yyy* is the same language code that you specified to change your print and display translation tables. (See 4a on page 62 above.)
2. Change the message text in FMN0M*yyy* in your library.
3. Modify the FMNUMODM member in FMN.SFMNSAM1 to meet your site's requirements, using the same language code as above. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMNUMODM.

Translating the ISPF messages text

All File Manager ISPF messages are provided in English. They are also provided in Japanese if you have installed the Japanese component of File Manager, and in Korean if you have installed the Korean component of File Manager. You can translate some or all of these messages into other languages.

All File Manager ISPF messages are stored in FMN.SFMNMENU. You translate a message as follows:

1. Find the members in FMN.SFMNMENU that contain the messages you want to translate. The message members specific to the File Manager base function are all named FMNB*zzzz* or FMNM*nn*.
2. Create a library with the same characteristics as FMN.SFMNMENU, with the name FMN.SFMNM*yyy*, where *yyy* is the same language code that you specified to change your print and display translation tables. (See 4a on page 62.) Copy the required message members from FMN.SFMNMENU to this library.
3. Change the required message texts in these members in your library.

To use the messages you have translated, see "Using the translated messages and panels" on page 64.

Note: Be sure to include **ALL** the messages in the message members you copy to your own library. When File Manager needs to display an ISPF message, it uses ISPF services to do this. Therefore the search for a message is made according to ISPF rules. Thus, if ISPF finds the message member it requires in your library, but the message number required is not in that member, ISPF will not look in any other library for the message, but will give an error. However, if you omit a complete message member from your library, then ISPF will use the English message member from the next library in the ISPMLIB concatenation. For more information about defining and using ISPF messages, see *z/OS ISPF Dialog Developer's Guide*.

Translating the panel text

All File Manager panels are provided in English. They are also provided in Japanese if you have installed the File Manager Japanese component, and in Korean if you have installed the File Manager Korean component. You can translate some or all of these panels into another language. (If no translated version of a particular panel is available, File Manager uses the English version.)

Translating the panel text

All File Manager panels are stored in FMN.SFMNPENU. You translate a panel as follows:

1. Find the members in FMN.SFMNPENU that you want to translate. The panel members specific to the File Manager base function are all named FMN*xzzzz* or FMN0*zzzz* where *x* is an alphabetic character (A—Z). (FM/IMS panels are named FMN1*zzzz* and FM/DB2 panels are named FMN2*zzzz*.)
2. Create a library with the same characteristics as FMN.SFMNPENU, with the name FMN.SFMNP*yyy*, where *yyy* is the same language code that you specified to change your print and display translation tables. (See 4a on page 62.) Copy the required panel members from FMN.SFMNPENU to this library.
3. Change the required panel text in the members in your library. A panel may reference a **help** panel by means of a `.HELP` statement. If any panel you are changing contains any of these `.HELP` statements, also copy and change these referenced members in your library.

To use the panels you have translated, see "Using the translated messages and panels."

Using the translated messages and panels

To use your translated messages in a batch job, specify the appropriate language with the LANGUAGE processing option, using the keywords shown in Table 15. See "Changing the default options" on page 23 for information on how to do this.

Table 15. Keyword values for the LANGUAGE option

Language	Code	Specify on LANGUAGE option...
French	FRA	FRENCH
German	DEU	GERMAN
Italian	ITA	ITALIAN
Japanese	JPN	JAPANESE
Portuguese	PTG	PORTUGUESE
Spanish	ESP	SPANISH
Danish	DAN	DANISH
Upper case English	ENP	UPPERENG
Korean	KOR	KOREAN
Swiss German	DES	SGERMAN
Traditional Chinese	CHT	CHINESET
Simplified Chinese	CHS	CHINESES
Other	XXX	OTHER

For example, to use French messages, specify LANGUAGE=FRENCH.

Under ISPF, the language used in messages and panels is determined by the setting of the national language for ISPF, for the current ISPF session. For information on changing the national language setting for your ISPF session, see *z/OS ISPF Dialog Developer's Guide*.

When your ISPF session is set up to use your language, you need to add your libraries to the appropriate ISPF concatenation, in front of any File Manager English libraries. For example, to use your translated messages, add

FMN.SFMNNM*yyy* to ISPMLIB, in front of FMN.SFMNMENU. To use your translated panels, add FMN.SFMNNP*yyy* to ISPPLIB, in front of FMN.SFMNPENU.

Customizing for the Japanese national language

Modifying the Japanese translation tables

The File Manager Japanese component provides translation tables for display and print that are used by File Manager when the Japanese language is selected. You can change these Japanese translation tables if necessary. You use the usermod FMNUMODJ and the member FMNTRJPN to change the Japanese translation tables. FMNUMODJ and FMNTRJPN are distributed in FMN.SFMNSAM1.

To do this:

1. Copy the member FMNTRJPN from FMN.SFMNSAM1 to your own source library.
2. Change the translation table definition statements in the FMNTRJPN source member in your library, according to your requirements.

   File Manager provides support under ISPF for command verbs and command keywords entered in lower or uppercase as long as they are valid SBCS-only strings, and the a - z codepoints in the host codepage used correspond to the standard EBCDIC a-z codepoints. If this is not the case, and you want to be able to enter lower or mixed case command verbs and command keywords, then you will need to modify the TRUPC translation table in FMNTRJPN, to specify the uppercase A - Z EBCDIC codepoints at locations corresponding to the a - z codepoints in the codepage used.
3. Modify the FMNUMODJ member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMNUMODJ.

Note: Under ISPF, to enable user DBCS data to be displayed or edited, the terminal type must be set to 3277KN or 3278KN, using ISPF option 0.

Changing the Japanese message text

If you have installed the File Manager Japanese component, all File Manager base function Japanese messages are stored in the FMN0MJPN source member. Normally you should not need to modify this module. However, if you do want to modify it, you can do so by means of the usermod, FMNUMODN.

To do this:

1. Copy the member FMN0MJPN from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN0MJPN in your library.
3. Modify the FMNUMODN member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMNUMODN.

Customizing for the Korean national language

Modifying the Korean translation tables

The File Manager Korean component provides translation tables for display and print that are used by File Manager when the Korean language is selected. You can change these Korean translation tables if necessary. You use the usermod FMNUMODK and the member FMNTRKOR to change the Korean translation tables. FMNUMODK and FMNTRKOR are distributed in FMN.SFMNSAM1.

To do this:

1. Copy the member FMNTRKOR from FMN.SFMNSAM1 to your own source library.
2. Change the translation table definition statements in the FMNTRKOR source member in your library, according to your requirements.
3. Modify the FMNUMODK member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMNUMODK.

Note: Under ISPF, to enable user DBCS data to be displayed or edited, the terminal type must be set to 3277KN or 3278KN, using ISPF option 0.

Changing the Korean message text

If you have installed the File Manager Korean component, all File Manager base function Korean messages are stored in the FMN0MKOR source member. Normally you should not need to modify this module. However, if you do want to modify it, you can do so by means of the usermod, FMNUMODQ.

To do this:

1. Copy the member FMN0MKOR from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN0MKOR in your library.
3. Modify the FMNUMODQ member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMNUMODQ.

Chapter 8. Customizing File Manager to use library management system libraries

File Manager provides the facility to allow you to access source code stored in either library management (LM) or source code management (SCM) systems.

The difference between library management systems and source code management systems is not well defined, but in general, SCM systems tend to provide software packaging, version management of packages, and build management in addition to the typical LM functions. LM systems tend to be simpler and focus on efficiently storing data (files) in a library and managing versions or levels of the individual files. An example of a library management system is CA-Panvalet. An example of a source code management system is IBM's SCLM.

In this chapter both LMs and SCMs are referred to as library management systems or LMSs.

There are three possible scenarios to accessing source code in LMSs:
- The LMS is CA-Panvalet.
- The LMS (not CA-Panvalet) provides a SUBSYS interface.
- The LMS (not CA-Panvalet) does not provide a SUBSYS interface.

If you want to use File Manager to access COBOL copybooks, PL/I include books, or HLASM copybooks stored in LMS libraries, the customization tasks you must perform are different, depending on which of the above scenarios applies to your LMS. The possible customization tasks are listed in Table 16.

Table 16. Summary of steps for customizing File Manager to use LMS libraries

Step	Description	Page
__ 1	Determine if your LMS provides a SUBSYS interface.	
__ 2	Set the LMS and LMSUBSYS options, as necessary.	307
__ 3	Ensure that the correct release of Language Environment is available to File Manager.	68, 68
__ 4	Determine if your LMS requires your exit to run below the 16MB line.	72
__ 5	Write your own File Manager exit and link edit it into a load library available to File Manager, as load module FMNCRAEX.	71, 439

Accessing source code in CA-Panvalet libraries

File Manager provides an interface to CA-Panvalet, enabling you to use File Manager to work with COBOL copybooks, PL/I include books, and HLASM copybooks stored in CA-Panvalet libraries. Before you can do this, you must change the default options to specify one of:
- **LMS=PANVALET**
- **LMS=(USERLMS,PANVALET)**
- **LMS=(PANVALET,USERLMS)**

If you want to access only CA-Panvalet libraries, specify **LMS=PANVALET**. For the use of LMS=(...,USERLMS) or LMS=(USERLMS,...), see "Other library management systems" on page 68.

© Copyright IBM Corp. 2000, 2011

For information about the LMS option, see “LMS” on page 307. For more information on how to change the default options, see “Changing the default options” on page 23.

If you do not change the LMS option to one of the above values, you will receive an error message if you attempt to access copybooks in a CA-Panvalet library.

Notes:

1. This support is provided for CA-Panvalet release 14 and above.
2. Ensure that your CA-Panvalet library is defined either with DSORG=DA, or with DSORG=PS and LRECL=0.

Other library management systems

Accessing source code in an LMS with SUBSYS interface

If your LMS provides a SUBSYS interface, you need to perform tasks #2 and #3 from Table 16 on page 67, as described here.

- Set the LMSUBSYS option to the value for the SUBSYS interface provided by your LMS vendor, (for example, LMSUBSYS=LAM).
- Ensure that the LMS option is set to NO.
- Ensure Language Environment is available to File Manager, if you have not already done so.

  You can do this by one of the following methods:
  – Add the Language Environment runtime library, SCEERUN, to your concatenated LINKLIST.
  – Place SCEERUN in the STEPLIB DD statement of the logon proc.
  – Your users can use the TSO command, TSOLIB, to add SCEERUN to the search list.

Note: You cannot access source code from an LMS with a SUBSYS interface and also from CA-Panvalet. The two are mutually exclusive.

Accessing source code in an LMS without SUBSYS interface

If your LMS does not provide a SUBSYS interface, you need to perform tasks #2, #3, #4, and #5, from Table 16 on page 67, as described here.

- Set the LMSUBSYS option to a blank (the IBM-supplied default in FMN0POPT.)
- Set the LMS option to USERLMS.
- Ensure Language Environment is available to File Manager, if you have not already done so.

  You can do this by one of the following methods:
  – Add the Language Environment runtime library, SCEERUN, to your concatenated LINKLIST.
  – Place SCEERUN in the STEPLIB DD statement of the logon proc.
  – Your users can use the TSO command, TSOLIB, to add SCEERUN to the search list.
- Code and link your own user exit, a load module named FMNCRAEX, into the File Manager load library. If you have installed File Manager in the default libraries this will be FMN.SFMNMOD1. You can access source code in multiple LMSs through a single FMNCRAEX exit.

If you are using an LMS without a SUBSYS interface, you cannot run File Manager APF-authorized.

Sample user code for FMNCRAEX, written in COBOL, is distributed in the File Manager sample library, FMN.SFMNSAM1. The following sections discuss the functions and requirements for FMNCRAEX. Appendix F, "The library management system exit," on page 439 discusses the sample exit itself.

Capabilities provided by File Manager via FMNCRAEX

Your user exit, FMNCRAEX, will enable File Manager to provide the following capabilities:

1. The ability to extract LMS files containing COBOL copybooks, PL/I include books and High Level Assembler copybooks. These files hold data definitions for File Manager template and view processing. File Manager template and view processing can be used by File Manager in the following situations:
   a. When browsing, viewing or editing a file or IMS database.
   b. When using certain File Manager utilities.
2. The ability to select from a list of members contained in the LMS. In the process of selecting from a member selection list (MSL), you can also view the contents of members with the View or Browse command.
3. The ability to support multiple LMSs through a single exit. For more information on writing your exit to support multiple LMSs, see "Supporting multiple library management systems" on page 447.

Restrictions

To be accessed from the File Manager panels, a named library must exist. To expand on this: suppose a library data set name is entered in an LMS-related File Manager panel field (for example, the Copybook or Template **Data set name** field on the edit entry panel). A PDS or PDSE data set of the given name must exist. File Manager checks this prior to calling the user exit. Therefore the exit cannot support user defined, logical names which would otherwise be translated by the exit.

Functions which must be provided by FMNCRAEX

File Manager uses FMNCRAEX to access a supported library. The exit must provide the following high level functions:

1. Initialize and terminate the exit.
2. Validate the library.
3. Extract a member from the library.
4. Get member information, or metadata. That is, get the name and other information about members, given a member name or name pattern containing wildcard characters.
5. Get display information. This provides to File Manager the heading to use on a member selection list ISPF panel for members in the library.

These functions are discussed in more detail in the following sections.

Initialize and terminate the exit

If you write your exit in a high level language (HLL) such as COBOL or PL/I, very little work is needed to initialize or terminate the exit. Language Environment initializes and preserves working storage for HLLs between calls.

If you write your exit in High Level Assembler (HLASM), you can use the Initialize and Terminate calls to get and release storage needed by the exit. The

second argument in every call to the exit, the RAM-WORK-AREA-PTR, can be set by the exit at any time. File Manager preserves and passes this value with every call to the exit. This provides a way for the exit to gain addressability to its own storage on every call.

Validate the library

File Manager allows users to enter a copybook library name and member name on many panels. By means of FMNCRAEX File Manager will allow multiple types of copybook libraries without having to specify the type.

The exit must be able to take a library name passed to it and determine whether it is a library which the exit supports. The possible results are:

1. **Yes**: the exit supports this library.
2. **No**: the exit does not support this library. File Manager should try other access methods (for example, CA-Panvalet or PDS I/O) but should not use the exit to access this data set.
3. **Unknown**: the exit was unable to access the library for various reasons (for example, data set in use or security violation).

Extract a member from the library

File Manager uses the exit to extract a member from the LMS library. Since File Manager uses this mechanism from panels which only allow specification of a library name and member name, the exit must be able to determine a unique member solely from the combination of the library and member names. This requires some default decisions by the exit. For example, an LMS can maintain multiple versions of a member. The exit may need to decide to always choose the most recent version when asked for the member by name.

Get member information

File Manager uses the exit to extract information about members, such as the name, date created, date last modified, size, and modification level.

The exit must format the data for each member to display the information on a File Manager ISPF Member Selection List (MSL) panel. This panel has a prefix command area or **SEL** field on the left of each line, then the member name, then the "Prompt" area where messages such as "*Browsed" can be placed, and finally space for member attributes such as date created. The attributes placed here may depend on the functions supported by the LMS, and are completely controlled by the exit. File Manager simply places the attribute text provided by the exit into the MSL panel.

Get display information

File Manager uses the "Get member information" function described above to present metadata on a File Manager ISPF MSL panel. The FMNCRAEX exit uses the "Get display information" function to provide the column headings for the MSL panel.

The sample exit

Sample code for the exit, FMNCRAEX, written in COBOL, is provided in FMN.SFMNSAM1. There are two members in FMN.SFMNSAM1 to do this, FMNCRAEX and FMNCRACJ.

FMNCRAEX is the COBOL code for the sample exit. It contains working COBOL code. Although FMNCRAEX performs a useless function - File Manager can use it to access members of a PDS as if they were in a library management system - it serves as a working model of an exit. To demonstrate the sample exit using FMNCRAEX, the PDS must be defined as FB 80 and must contain a member named $$FMN$$ in order to be validated as a supported library.

FMNCRACJ contains JCL which compiles two COBOL programs (FMNCRAEX and TEST) and then runs the main (TEST) program. The TEST program calls the sample COBOL exit FMNCRAEX. You can use the COBOL program, TEST, to demonstrate the exit and to verify your own exit prior to making it available to File Manager. You do not have to have any LMS library available to run the demonstration. Refer to the prolog to FMNCRACJ for information about the changes you need to make.

Writing your own exit

To provide your own exit in COBOL, you can use the members FMNCRAEX and FMNCRACJ. FMNCRAEX and FMNCRACJ are distributed in FMN.SFMNSAM1. To do this:

1. Code your own version of FMNCRAEX in your source library, using FMNCRAEX from FMN.SFMNSAM1 as a base.
2. Modify the sample job FMNCRACJ in FMN.SFMNSAM1 to meet your site's requirements. Refer to the sample job for information about any changes you might need to make.
3. Modify the TEST program in FMNCRACJ as required.
4. Run FMNCRACJ to verify your exit.
5. When you are satisfied with the testing of your exit using FMNCRACJ, link edit your version of FMNCRAEX into a load library available to File Manager. If you installed File Manager into the default libraries this will be FMN.SFMNMOD1.

The COBOL sample exit is described in detail in Appendix F, “The library management system exit,” on page 439.

To provide your own exit in another high level language or High Level Assembler, code your own version of FMNCRAEX in your own source library, compile it and link edit it into a load library available to File Manager. If you installed File Manager into the default libraries this will be FMN.SFMNMOD1.

If you link edited your version of FMNCRAEX into your own load library, you might want to add this library to the STEPLIB DD statement in the batch JCL skeleton, for use in batch jobs. For more information about the batch JCL skeleton, see “Changing the JCL skeleton for batch mode” on page 24.

If you want to test your exit before making it generally available in production, link edit your version of FMNCRAEX into your own load library and use the TSOLIB command to activate your own load library. Refer to the *z/OS TSO/E Command Reference*, for information about the TSOLIB command.

If the CONCATD command is available on your system, you can add your load library to the ISPLLIB concatenation using the TSO command:

```
CONCATD F(ISPLLIB) DA(your.loadlib) SHR BEFORE
```

Notes:

1. Some LMSs require that code calling them, or the data structures passed to them, or both, be below the 16 MB line. If that is the case, you must provide the appropriate compiler and link edit options.
2. If you are using FMNCRAEX, you cannot run File Manager APF-authorized.
3. Refer to "Writing the exit in HLASM" on page 447 for other considerations when writing your exit in HLASM.

Performance considerations

The major performance considerations for the exit are related to GetMemberInfo and wildcard processing:

If there are no wildcards

When a member name is provided to GetMemberInfo without any wildcard characters (* or %), attention should be given to ensuring the fastest possible response time. This function is used as an existence check on the member. The member attributes (RHS field) are not displayed on an ISPF panel, so, if you want to, you can leave this field blank.

If a wildcard is present

Some user libraries contain large numbers of members. Ten to thirty thousand members are not unusual. Attention should be given to finding ways to give good performance when GetMemberInfo is called with wildcards. How to do this depends on the functions offered by the LMS. Be aware that File Manager gets all members matching the given pattern from the exit before presenting any information to the user via a member selection list.

Chapter 9. Customizing File Manager to use an I/O exit

File Manager allows you to provide a user I/O exit for use when processing data. This enables you to process records in data sets which are subject to any kind of pre- or post-processing not offered directly by File Manager. Such processing could include compression, encryption, or other site-specific activities performed on records in a data set.

There are a number of ways that data sets can be processed, and how the processing is made visible to users. Therefore File Manager allows you to provide an I/O exit, to be associated with a particular data set, and to specify the name of the exit. The I/O exit is a program that interfaces with File Manager using a predefined communication area.

This I/O exit is not intended to fully support compressed or encrypted data sets. Specifically, it does not provide member name list access to data sets which are in other than PDS or PDSE format. However, you can access members of a proprietary compressed library (PDS or other format) through the I/O exit, as long as the member name is known and is provided to File Manager.

The File Manager I/O exit operates only on the data, not the data set, and supports the following functions:

- Interactive view, edit and browse in the File Manager base component
- Batch and interactive utilities:
  - Data set copy (DSC)
  - Data set compare (DSM)
  - Data set edit in batch (DSEB)
  - Data set generate (DSG)
  - Data set print (DSP)
  - Data set update (DSU)
  - Find/change (FCH)

You cannot use the File Manager I/O exit to operate directly on COBOL copybook or PL/I include data sets, or on template data sets. The I/O exit, when specified, can apply only to the data being processed, not to the template or copybook data sets used to format or select the processed data. However, the I/O exit can be used when the template or copybook data set is itself the subject of a function.

You use the library management system (LMS) exit to process COBOL copybook or PL/I include data sets, or template data sets, when associated with an input or output data set. For information on providing an LMS exit, see Chapter 8, “Customizing File Manager to use library management system libraries,” on page 67 and Appendix F, “The library management system exit,” on page 439.

To specify that an I/O exit can be used, set the USEIOX option, in FMN0POPT. If required, you can also provide the name of a site-specific I/O exit. For more information about USEIOX, see “USEIOX” on page 326. For information about changing the options in FMN0POPT, see “Changing the default options” on page 23. This exit name can be overridden by the user interactively or in batch. *File Manager User's Guide and Reference* describes how to use the I/O exit.

© Copyright IBM Corp. 2000, 2011

Exit protocol

The I/O exit provides record-level support. File Manager performs all the I/O functions and the exit can process the record after it is read or before it is written. The exit supports the following protocol:

Initialization
The exit indicates that the specified data set is to be processed by the exit and returns information about supported functions (once per data set).

Open File Manager opens the data set and provides vital information about the data (multiple possible per data set).

A series of record access calls implementing the following functions:

- Record read
- Record write

The record to be processed is provided in an input buffer. The exit should process the data and return it in the output buffer while setting the output (record) length. A length of zero for the output record indicates that the processing did not change the record and the input record can be used "as is".

At any time during read/write processing the exit can indicate that it should no longer process the current data set. File Manager will then stop invoking the exit for read/write processing and continue processing without the involvement of the exit. The Close and Terminate calls will still occur.

Note: The read and write requests may occur in random record order. No assumption about data processing should be made based on such order. This is important with some encryption technologies which require a sequentially ordered data stream.

Error codes can be set and messages returned by the exit to be displayed or printed for the user. Some codes immediately terminate the current function while some allow for continued processing. The messages are displayed regardless of the return code every time there is a non-blank message returned by the exit. See also page 80 for information about messages returned by your exit.

Close File Manager closes the data set after returning from the exit.

Termination
The exit performs any required 'cleanup' processing.

Writing your exit

File Manager does not supply a default I/O exit. If you plan to use an I/O exit to process your data, you must provide one or more for your installation. You must also set the USEIOX option in FMN0POPT. For information about the USEIOX option, see “USEIOX” on page 326.

Any exit you provide must be in the form of a load module, in any load library available to File Manager, either by a STEPLIB DD statement, or in LINKLIST, or LPALIST. If File Manager attempts to load an exit and is unable to find it, an error message is displayed.

You can write your own exit in any high level language, for example, COBOL, PL/I, or High Level Assembler. Sample exits for COBOL, PL/I, and HLASM are provided in FMN.SFMNSAM1. They are:

FMNIOXEA Sample HLASM exit

FMNIOXEC Sample COBOL exit

FMNIOXEP Sample PL/I exit

FMNIOXHF Sample HLASM exit using HFS

Copybooks providing the I/O exit control blocks for COBOL, PL/I, and HLASM are distributed in the File Manager macro library, FMN.SFMNMAC1. They are:

FMNIOXCB Control block for HLASM programs

FMNIOXCC Control block for COBOL programs

FMNIOXCP Control block for PL/I programs

These copybooks are described in "Exit control block data names." See "Using the I/O exit control block" on page 79 for information on how to write your exit using the copybooks, and the processing your exit should perform.

Notes:

1. The File Manager I/O exit does not support 24-bit addressing mode.
2. File Manager supports an exit written in any supported release of COBOL and PL/I.
3. File Manager returns general information about the data set, which can be interpreted by your exit routine. Some data in the exit control blocks, (for example, DSORG, RECFM), mimic data used in z/OS system control blocks. Therefore, for convenience, in the COBOL copybook, FMNIOXCC, level-88 values are defined.
4. File Manager also processes HFS files as simulated QSAM files. For more information about File Manager and HFS files, refer to "Using UNIX System Services and the Hierarchical File System" in the *File Manager User's Guide and Reference*. There are no major differences on the exit routine level. An exit routine will get an address to a path name rather than the data set name in this case, but the way a (simulated) record is processed remains the same.

Exit control block data names

Table 17. Exit data names

Description	Data type	Set by:	Value
Eye catcher	CL8	FM initialization	"UIOEXCB"
Interface version	F	FM initialization	**1** Initial value
Level support	F	I/O exit initialization	**0** Process without exit **1** I/O performed by File Manager **2** Reserved for future use

Table 17. Exit data names (continued)

Description	Data type	Set by:	Value
Function code	F	FM, for every call to the exit	**1** Initialize once per data set **2** Terminate once per data set **3** Open, multiple possible per data set **4** Close, multiple possible per data set **5** Read, required **6** Write, required **7** Reserved for future use **8** Reserved for future use **9** Reserved for future use
Exit return code	F	I/O exit	**0** OK, default. See note [1] **4** Warning **8** Error, terminate function **12** Severe error, terminate function **16** Fatal error, terminate function
Warning code	F	I/O exit	**0** Normal warning, default. See note [1] **1** Reserved for future use **2** Complete, continue processing without exit
Functions supported by exit	X	I/O exit initialization	**1** Read, required **2** Write, required **3** Read + write, default **32** Reserved for future use **64** Reserved for future use **128** Reserved for future use
Operating mode	X	FM initialization	**1** TSO in batch **8** Keyword mode **16** Command mode **32** Query/answer mode **64** Fullscreen ISPF **128** Batch, without TSO
DDNAME	CL8	FM initialization	DDname allocated by File Manager
DSN	CL44	FM initialization	fully qualified DSN
Member name	CL8	FM initialization	Member name, only if the data set is opened as sequential, not PO
VOLSER	CL6	FM initialization	First VOLSER of the data set, set to blank if not used

Table 17. Exit data names (continued)

Description	Data type	Set by:	Value
DSORG	2X	FM open	Data set organization: 1st byte: **1** Unmovable **2** PDS(E) **32** DA **64** PS **128** IS 2nd byte: **1** HFS **8** VSAM **16** IAM involved
VSAM catalog entry	X	FM open	**8** Path **16** Alternate index **32** Index **64** Data **128** Cluster
VSAM type information	X	FM open	**4** IAM **8** LDS **16** VRRDS **32** RRDS **64** ESDS **128** KSDS
RECFM	X	FM open	**2 - M** Machine control character **4 - A** ASA control character **8 - S** Spanned **16 - B** Blocked **32 - T** Track overflow **64 - V** Variable **128 - F** Fixed **128+64 - U** Undefined
Allocation mode	X	FM initialization	**1** Old **2** Mod **4** New **8** (new,catlg) **16** Shr **128** JCL/DD allocated
Open mode	X	FM open	**1** Input **2** Output **4** Update/inout **8** Initial load (VSAM)
Processing mode	X	FM initialization	**128** Temporary file used in support edit of data (dsn - original name preserved)
LRECL	F	FM open	record length (non-VSAM)
BLKSIZE	F	FM open	block size (non-VSAM)
MLRECL	F	FM open	maximum record length (VSAM)
CISZ	F	FM open	CI size (VSAM)
KEYLEN	F	FM open	key length

Table 17. Exit data names (continued)

Description	Data type	Set by:	Value
RKP	F	FM open	relative key position (relative to zero offset into the IO area)
RBA	XL8	FM or I/O exit read	XRBA (VSAM)
SLOT	F	FM or I/O exit read	slot number (VSAM, RRDS/VRRDS)
MAXL	F	I/O exit open	maximum record length ever to be returned by the exit after read and decompression/decoding. The initial value is the same as that for the file on DASD.
Estimated raw (that is, uncompressed) data set size, as a percentage of the original size. See note [2]	F	I/O exit open	**0** Unknown **nnn** Percentage of the original size **100** Initial value
Current input record pointer	A	FM, all IO	I/O area address passed to the exit (size at least as big as the maximum record length) holds the record to be processed by read/write
Current input record length	F	FM, all IO	length of record passed to the exit
Current output record pointer	A	FM, all IO	I/O area address passed to the exit (size at least as big as the maximum record length) for the output record, holds the record being the result of processing by read/write
Current output record length	F	I/O exit, all IO	Length of record passed from the exit, if zero then the input record will be assumed to be unchanged and the output record will be ignored
HFS path name pointer	A	FM initialization, open	Address of an HFS path name
HFS path name length	F	FM initialization, open	Length of an HFS path name
HFS object type	X	FM initialization, open	HFS object type specified by the HFS path: **4** Socket **8** Symbolic link **16** FIFO **32** Regular file **64** Special character file **128** Directory **Note:** Only regular files are supported. Directory can be set at initialization only. Other types are added for completeness.

Table 17. Exit data names (continued)

Description	Data type	Set by:	Value
HFS processing mode	X	FM initialization, open	HFS processing mode used in an HFS file processing: **64** Binary mode (records of fixed, arbitrary defined, length) **128** Text mode (record boundaries determined by delimiters)
Message	CL80	I/O exit, all	(Error) message - can be provided for output after each operation, non-error messages will be suppressed after 100 messages per session. See note [3]
User scratchpad area	CL1024	UIOEX, all	Maintained by the exit from INIT to TERM

Notes:

1. Zero is the default. This code is initialized to zero on every call from File Manager to the exit, before the control block is passed to the exit.
2. This is a compression ratio, where 100 means the same size as the original data. 100 is the initial value. See "Open call discussion" on page 81 for more information, and examples of a compression ratio.
3. See page 80 for more information about message suppression.

Using the I/O exit control block

The I/O exit control block is the single parameter passed with each call from File Manager to the I/O exit. It is also used to return data from the I/O exit to File Manager.

Function codes and the flow of I/O exit processing

The I/O exit control block contains a function code field. This is set by File Manager before the I/O exit is called. It indicates the type of I/O function being performed by File Manager. The I/O exit is given the opportunity to perform actions at the points named by the function codes during File Manager processing. The function codes are also discussed in "Exit protocol" on page 74. They are: Initialization, Termination, Open, Close, Read and Write.

Note that File Manager makes the "real" I/O calls to the operating system. As a result, the I/O exit does not get control at exactly the time of the real I/O operation. Instead, the I/O exit gets control, as follows:

- **Open:** After the "real" data set open
- **Close:** Before the "real" close
- **Read:** Immediately after a read from DASD (before File Manager truncation or padding or other changes)
- **Write:** Immediately before a write to DASD (after File Manager truncation or padding or other changes)

The Function Code is the key input that your I/O exit should use to determine what actions to take. The following sections discuss the flow of control and data during I/O exit processing by focussing on the call type, as determined by the function code.

Information that should always be returned from the I/O Exit to File Manager

The I/O exit should set any non-zero exit return code as appropriate. The exit return code field is initialized to zero on every call from File Manager to the exit.

The I/O exit can also set an informational, warning, or error message into the message field of the I/O exit control block. You can code your exit to return any number of messages, however File Manager will suppress all informational (RC=0) and warning (RC=4) messages after the first 100. Error messages (RC=8 or higher) are never suppressed. The File Manager message:

```
Number of messages issued by exit reached the limit. Remaining non-error (RC < 8)
messages will be suppressed
```

is issued after 100 non-error messages have been produced. The limit of 100 messages cannot be changed.

The I/O exit can set the warning code to 2 at any time, to tell File Manager to continue processing and not call the I/O exit again. The I/O exit will still be called for termination and, if an open was made, for close. In order for File Manager to check the warning code, the I/O exit must also set the exit return code to 4 (warning).

Initialization call discussion

The I/O exit is always called, for any particular data set, first with one (and only one) initialization call. This is an opportunity for the exit to indicate the type of processing it will provide for the data set, for example, to allocate storage.

At initialization, the exit should decide whether to process the data set, and, if necessary, obtain dynamic storage.

- **Deciding whether to process the data set:** the data set name and other information is available in the I/O exit control block. If the exit "knows" from the data set name and other information that it should not process the data set, the exit should indicate that File Manager should not call the exit again. There are two ways to do this:
  - Only at initialization, you can set the level support field to 0. This indicates that File Manager should not call the exit.
  - In response to any I/O exit call, you can set the exit return code to 4 (warning), and set the warning code to 2, to avoid further calls, and let File Manager handle the remaining I/O without the exit.

  Otherwise set the level support field to 1 to indicate that the exit will process the data set.
- **Obtaining dynamic storage:**
  - Typically only an assembler exit needs to obtain storage. COBOL and PL/I storage and addressability is preserved between calls to the exit. However, File Manager provides both a current input and a current output record buffer, so even an assembler exit may not need to allocate storage.

– Pointers to allocated storage and other information can be saved in the user scratchpad area at the end of the I/O control block, and will be maintained between initialization and termination calls.

There is no reason to change the values in the **Functions supported by exit** field. Both read and write support are required, and this is the default.

Termination call discussion

The last call for a given data set is always one (and only one) termination call. The termination call is intended primarily to allow de-allocation of resources allocated at initialization time. File Manager makes the termination call even if exit processing was terminated by an error or a request from the exit to continue without calling the exit again.

Open call discussion

The I/O exit may receive multiple consecutive open calls, for various reasons internal to File Manager. If the exit receives at least one open call, then the exit will receive at least one close call.

The open function code means the exit is being called immediately after a "real" file open. Additional information is now available (compared to what was available at initialization). The exit should:

- Decide whether to continue processing the data set. (See “Initialization call discussion” on page 80.)
- Set the MAXL (maximum data length) field in the I/O exit control block. There is no need to change this value if the record length is not to be changed by the exit.
- Set the estimated raw data set size, as a percentage of the original. For example, setting the estimated raw data set size to:
  50 Indicates that an unencrypted data set is 50% of the encrypted data.
  100 Indicates that an unencrypted data set will be the same size as the encrypted data set.
  200 Indicates that the unencrypted data set will be twice the size of the encrypted data set (for example, as a result of decompression).

  This allows File Manager to estimate the amount of in-memory storage that will be required. The sample exits in FMN.SFMNSAM1 use a setting of 200.

Close call discussion

The I/O exit may receive multiple consecutive close calls, for various reasons internal to File Manager. If the exit receives at least one open call, then the exit will receive at least one close call.

The close function code means the exit is being called immediately prior to a "real" file close.

Read and Write call discussion

In this section:

- The I/O exit control block field "open mode" refers to the data set open operation, therefore "input" means open for input, and "output" means open for output.

- References in I/O exit control block fields to "current input record" refer to the data in the input buffers provided by File Manager to the I/O exit. Similarly, references to "current output record" refer to the data in the output buffers.

Processing flow for read and write function codes:

- **During a read operation**, File Manager reads a record from a data set and places it in the I/O exit current input record. File Manager then calls the exit with a read function code. The exit should then copy the current input record to the current output record buffer, performing any required processing. On return from the exit, File Manager uses the processed current output record from the exit.
- **During a write operation**, File Manager takes an in-memory record and places it into the I/O exit current input record. The exit should then copy the current input record to the current output record buffer, performing any required processing. On return from the exit, File Manager writes the processed current output record from the exit to DASD.

Important additional information to note about Read and Write

- In addition to setting the current output record data, the I/O exit should also set the current output record length during a read or write operation. If the current output record length is not set, or is set explicitly to zero, any changes in the current output record data buffer will be ignored, and the current input record will be used as the current output record.
- When processing a variable length sequential file, the data portion **does not** include the record descriptor word (RDW) containing the record length. The length is stored in a separate field in the control block.
- The I/O exit code should not assume that it is operating on records in sequential order. For example, the File Manager editor might read or write individual records out of order.
- The I/O exit could receive read and write calls in any sequence. For example, the File Manager Find/Change utility might read a number of records and then write a number of records, and then go back to reading.

Installing your exit

You can provide more than one user I/O exit. For example, you might want to provide a different exit for each user, or different exits for use with different data sets. You can only specify one exit name for the option USEIOX, so you could make this your site-specific or default exit. Individual users can specify different exits interactively or in batch.

Two usermods are provided, FMNUMODU and FMNUMODH, to enable you to install High Level Assembler exits under the control of SMP/E. FMNUMODU will install a basic exit written in HLASM. FMNUMODH will install an exit written in HLASM, which enables File Manager to process data in an HFS. Other exits written in HLASM, and exits written in other high level languages must be provided outside of SMP/E.

To provide a site-specific exit in HLASM, using FMNUMODU or FMNUMODH:

1. Change the USEIOX option in FMN0POPT. For more information on how to change the default options, see “Changing the default options” on page 23. For information about the USEIOX option, see “USEIOX” on page 326.
2. Code your own version of FMNIOXEA or FMNIOXHF, in your own source library. Refer to the information in “Exit control block data names” on page 75

and "Using the I/O exit control block" on page 79. You can use FMNIOXEA or FMNIOXHF in FMN.SFMNSAM1 as examples.

3. Modify the FMNUMODU or FMNUMODH members in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermods for information about changes you may need to make.
4. Install SMP/E usermods FMNUMODU or FMNUMODH. Your exit will be installed in FMN.SFMNMOD1 if you installed File Manager into the default libraries.

To provide further exits in HLASM:

1. Code your own exit in your own source library. You can use FMNIOXEA or FMNIOXHF in FMN.SFMNSAM1 as examples.
2. Assemble and link your exit into a load library available to File Manager, for example, FMN.SFMNMOD1.

JCL is provided in the job FMNIOXHA to help you to assemble and link your exit. Refer to the instructions in the job for changes you need to make. FMNIOXHA is distributed in FMN.SFMNSAM1.

To provide your own exit in COBOL or PL/I:

1. If you plan to use this exit as your site-specific or default exit, change the USEIOX option in FMN0POPT.
2. Code your own exit in your own source library. Refer to the information in "Exit control block data names" on page 75 and "Using the I/O exit control block" on page 79. You can use FMNIOXEC or FMNIOXEP in FMN.SFMNSAM1 as examples.
3. Compile and link your exit into a load library available to File Manager, for example, FMN.SFMNMOD1.

JCL is provided in the job FMNIOXCO to help you to compile and link your exit in COBOL. JCL is provided in the job FMNIOXPL to help you to compile and link your exit in PL/I. Refer to the instructions in the jobs for changes you need to make. FMNIOXCO and FMNIOXPL are distributed in FMN.SFMNSAM1.

Chapter 10. Customizing File Manager to use a scrambling exit

File Manager provides the facility to enable you to create test data based upon production data but allowing the user to change the values of certain fields that may contain sensitive information. This facility is known as field scrambling.

In addition to the facilities provided for scrambling in File Manager, you can also write your own scrambling exit, to perform scrambling operations that cannot be done by the supplied scrambling function.

To specify a scrambling exit, select **Scramble Type** 4 ("Exit") from the Field Attributes panel of the template editing process, where you have specified the fields to be scrambled. File Manager displays the Scramble Exit Specification panel where you can provide the exit name, a constant to be passed to the exit, and specify whether numeric fields are to be formatted before the value is passed to the exit. The exit can be an assembler or LE-enabled program. The *File Manager User's Guide and Reference* describes how to use the scrambling exit.

Exit protocol

The scrambling exit is called during a copy operation with an input and output template, and where the output field has been defined with **Scramble Type** 4 ("Exit") and mapped to an input field. The exit is called in these situations:

Initialization
The first call to the exit for each field is an initialization call that causes the exit to perform set-up procedures. A 1024-byte user area is provided, in addition to the user constant for the exit to use.

Scramble
This call is repeated for each instance of the field being copied. On this call, the exit is expected to provide File Manager with the scrambled output value for the field, or a directive for File Manager to scramble the field.

Termination
The last call for each field is a termination call that causes the exit to perform clean-up procedures.

Writing your exit

File Manager does not supply a default scrambling exit. If you plan to use a scrambling exit you must provide one or more for your installation.

Any exit you provide must be in the form of a load module, in any load library available to File Manager, either by a STEPLIB DD statement, or in LINKLIST, or LPALIST. If File Manager attempts to load an exit and is unable to find it, an error message is displayed.

You can write your own exit in any high level language, for example, COBOL, PL/I, or High Level Assembler. Sample exits for COBOL, PL/I, and HLASM are provided in FMN.SFMNSAM1. They are:

© Copyright IBM Corp. 2000, 2011

FMNSCXEA Sample HLASM exit

FMNSCXEC Sample COBOL exit

FMNSCXEP Sample PL/I exit

These samples perform a simple scrambling algorithm of changing all uppercase characters to 'A', all lowercase characters to 'a', and all numbers to '1'. They are designed purely to demonstrate the capabilities of an exit. Refer to the prolog in each sample for more information about coding your exit.

Copybooks providing the scrambling exit control blocks for COBOL, PL/I, and HLASM are distributed in the File Manager macro library, FMN.SFMNMAC1. They are:

FMNSCXCA Control block for HLASM programs

FMNSCXCC Control block for COBOL programs

FMNSCXCP Control block for PL/I programs

These copybooks are described in "Exit control block description." See "Using the scrambling exit control block" on page 89 for information on how to write your exit using the copybooks.

Notes:

1. The File Manager I/O exit does not support 24-bit addressing mode.
2. File Manager supports an exit written in any supported release of COBOL and PL/I.
3. File Manager provides information concerning the input and output field formats as described by the exit control block. See Table 18

Exit control block description

Table 18. Exit control block

Description	Data type	Set by:	Value
Eye catcher	CL8	FM initialization	'UFLDXCB'
Version	F	FM initialization	**1** Initial value
Function Code	F	FM for every call to the exit	**1** **Initialize** - once per field **2** **Scramble** - each time the field is copied **3** **Terminate** - once per field
Input Field Address	A	FM, for every scramble call	Field area address of the input field value
Input Field Length	F	FM, for every scramble call	Field length of the input field value area
Input Field Subscript Address	A	FM, for every scramble call	For dimensioned fields this contains the address of an array of fullword subscripts values
Input Field Dimensions	F	FM, for every scramble call	Number of subscripts for dimensioned fields
Input Field Name Address	A	FM, for every scramble call	Address of template or copybook field name

Table 18. Exit control block (continued)

Description	Data type	Set by:	Value
Input Field Name Length	H	FM, for every scramble call	Template or copybook field name length
Input field type	H	FM for every scramble call	**1** Alphanumeric (includes alphabetic, and group items). **2** Binary. **3** Packed decimal. **4** Zoned decimal. **5** Numeric edit. **6** External floating point. **7** Floating point. **8** Decimal floating point. **9** Date. **10** Time. **11** Timestamp. **12** DBCS.
Input nullable	CL1	FM for every scramble call	(Y/N). Y is for DB2 nullable columns, otherwise N.
Input varying	CL1	FM for every scramble call	(Y/N). Y is for varying field or column, otherwise N.
Input varying	CL1	FM for every scramble call	(Y/N). Y is for varying null terminated strings, otherwise N.
Input signed	CL1	FM for every scramble call	For COBOL: **0** No sign clause. **1** Leading sign. **2** Leading sign separate. **3** Trailing. **4** Trailing sign separate. For PL/I: **Y** Signed. **N** Not signed.
Input DB2 null indicator	H	FM for every scramble call	DB2 null indicator. Only valid for DB2 nullable columns.
Input varying field length	H	FM for every scramble call	The length value for a varying field or column.
Input field precision	H	FM for every scramble call	This is the precision. That is, the number of significant digits of a binary, zoned or packed decimal field.
Input field scale	H	FM for every scramble call	The scale. That is, the number of digits to the right of the decimal point (it indicates where the decimal point would be positioned).
Output field address	A	FM for every scramble call	Field area address of the output field value.
Output field subscript address	A	FM for every scramble call	For dimensioned fields, this contains the address of an array of fullword subscript values.
Output field dimensions	F	FM for every scramble call	Number of subscripts for dimensioned fields.
Output field name address	A	FM for every scramble call	Address of Template or copybook field name.

Table 18. Exit control block (continued)

Description	Data type	Set by:	Value
Output field name length	H	FM for every scramble call	Template or copybook field name length.
Output field type	H	FM for every scramble call	**1** Alphanumeric (includes alphabetic, and group items). **2** Binary. **3** Packed decimal. **4** Zoned decimal. **5** Numeric edit. **6** External floating point. **7** Floating point. **8** Decimal floating point. **9** Date. **10** Time. **11** Timestamp. **12** DBCS.
Output nullable	CL1	FM for every scramble call	(Y or N). Y is for DB2 nullable columns, otherwise N.
Output varying	CL1	FM for every scramble call	(Y or N). Y is for varying field or column, otherwise N.
Output varying	CL1	FM for every scramble call	(Y or N). Y is for varying null terminated strings, otherwise N.
Output signed	CL1	FM for every scramble call	For COBOL:: **0** No sign clause. **1** Leading sign. **2** Leading sign separate. **3** Trailing. **4** Trailing sign separate. For PL/I: **Y** Signed. **N** Not signed.
Output field precision	H	FM for every scramble call	The precision. That is, the number of significant digits of a binary, zoned, or packed decimal field.
Output field scale	H	FM for every scramble call	The scale. That is, the number of digits to the right of the decimal point (it indicates where the decimal point would be positioned).

Table 18. Exit control block (continued)

Description	Data type	Set by:	Value
Return Code	F	FM sets to zero, exit sets on return	**0** Successful. For Scramble option , the output value is used for the output field. **1** Random. Request for FM to perform random scrambling. Only valid for Scramble option . **2** Repeat. Request for FM to perform repeatable scrambling. Only valid for Scramble option . **3** Translate. Request for FM to perform translatable scrambling. This requires that a data set name and column in and out values have been provided on the field attributes panel for this field. **8** Unsupported input field type. **10** Invalid input value. **16** Severe Error terminate.
Output field length	F	FM for every scramble call. Exit to set on return for formatted fields.	This value should be provided when returning a formatted numeric. The output area length is 80 and the returned length must be less than or equal to this.
Output null indicator	H	FM for every scramble call. Exit to set when returning for nullable field.	The exit can change the null indicator for nullable fields by setting this value.
Output varying length	H	FM for every scramble call. Exit can set when returning varying field.	The exit can set this value for a varying field on return. For any other type of field, the value is ignored.
User constant	CL80	FM for every scramble call	This contains the constant defined during scramble exit specification in template edit.
User Area	CL1024	Exit to set the values. Same area is referenced for each call.	This is a user area for the exit to use and is available for every call to the exit.

Using the scrambling exit control block

Function codes

The function code determines the action of the scramble exit. The initialization and termination calls are once for each field. The scramble calls are once for each instance. If a field is an array, then the exit is called for each element of the array.

Return codes

For initialization and termination, the return code is either zero for success, or non-zero indicating a severe error has occurred and File Manager is to be terminated.

For scramble, the exit can return codes as described in the control block.

0 Instructs File Manager to update the output field with the value return. If the value passed to the exit was formatted numeric, then File Manager treats the value returned from the exit as a formatted numeric which is converted to the internal format before updating the field.

1 Instructs File Manager to perform random scrambling on the field. Any scrambling value or range specifications are used. The output field value is ignored.

2 Instructs File Manager to perform repeatable scrambling on the field. Any scrambling value or range specifications are used. The output field value is ignored.

3 Instructs File Manager to perform translate scrambling on the field. For this return code to be valid, you must have specified a value data set (**Dsn**), column, and column out values **In** and **Out**), as well as selecting the **Value** option. If you have not provided these options for the field, then the copy process is terminated with an error.

8 The scramble exit being called does not support the input field type. The process is terminated with an error message.

10 The scramble exit being called has found an invalid input field value. The process is terminated with an error message.

16 Severe error - terminate.

Note: Any other return code is interpreted as a severe error that terminates processing.

Return output field values

When a return code of zero is returned with the function code for scramble, the output field is updated with the output value provided by the scramble exit. The output value should be valid for the field being updated.

Installing your exit

You can provide more than one user scrambling exit. For example, you might want to provide a different exit for each user, or different exits for use with different data sets. Also, you could define a different exit for each field or column you have selected for scrambling. Thus one copy, export or import operation could load multiple exit programs to satisfy the scrambling requirements of different fields or columns. Individual users can also specify different exits.

To provide an exit in HLASM, COBOL or PL/I:

1. Code your own exit in your own source library. Refer to the information in “Exit control block description” on page 86 and “Using the scrambling exit control block” on page 89. You can use FMNSCXEA as a HLASM example, FMNSCXEC as a COBOL example, or FMNSCXEP as a PL/I example. These samples are distributed in FMN.SFMNSAM1.
2. Assemble or compile and link your exit into a load library available to File Manager, for example, FMN.SFMNMOD1.

Sample jobs are provided to help you do this. These are:

FMNSCXHA Assemble and link a HLASM exit

FMNSCXCO Compile and link a COBOL exit

FMNSCXPL Compile and link a PL/I exit

Refer to the instructions in the jobs for changes you need to make. These jobs are distributed in FMN.SFMNSAM1.

Chapter 11. Verifying the customization of File Manager

After you have completed all the necessary steps involved in the initial installation and customization of File Manager base function, you need to perform the following steps to verify your customization, before completing the installation process. You might need to refer to the *File Manager User's Guide and Reference*.

1. Edit and submit the batch installation verification program (IVP) job, FMNVERIF, as described in the *File Manager Program Directory*.

   Check the results to ensure the job ran successfully.
2. Log on to a TSO userid that is enabled to access and use File Manager.
3. Start File Manager:
   - If you have added an option for File Manager to your ISPF Primary Options menu (see "Adding File Manager to the ISPF menu" on page 11), enter the option value you have assigned to File Manager and press enter. For example, if you have assigned 'F' to File Manager, enter 'F' and press Enter.
   - If you defined File Manager to the ISPF command table (see "Defining File Manager in an ISPF command table" on page 11), verify that File Manager can be started by entering the command 'FM' on any ISPF command line.

   The File Manager Primary Option Menu is displayed.

   Note: The first time you use File Manager, a Copyright panel appears. After reading the panel text, press the Cancel key (PF12). In subsequent File Manager sessions, this panel will not automatically appear.
4. Enter `ver` on the command line to display the release level and PTF level of File Manager. A panel is displayed similar to this:

```
IBM File Manager for z/OS Version 9 Release 1

(not APF authorized)

Service Levels of installed components

             Base       IMS        DB2        CICS
English      -NONE-     -NONE-     -NONE-     -NONE-
```

   Notes:

   a. This will always show File Manager as `not APF authorized`, even if you have made File Manager APF-authorized, as File Manager cannot run APF-authorized under ISPF.

   b. When you first install File Manager, `-NONE-` will be shown under each component that you have installed. Subsequently, when you have applied service to File Manager, a PTF number will be shown, indicating the PTF level of each component you have installed. If you have not installed a component, that component will not be shown at all. If you have installed the Japanese or Korean language components, another line will be displayed indicating the service level of that component.
5. Press the Exit key (PF3) to end the File Manager session, and return to your ISPF menu.

Several other sample batch jobs are distributed in FMN.SFMNSAM1 to help you verify the installation of File Manager and demonstrate it's functions. They are:

© Copyright IBM Corp. 2000, 2011

Sample job	Function
FMNSAMC1	Copy all records from one file to another, modifying a record in the process.
FMNSAMF1	Find JCL exec statements that run a particular program.
FMNSAMF2	Find JCL EXEC statements that run a particular program, and rename the program in each member in place.
FMNSAMP1	Print the JCL members that execute a particular program.
FMNSAMP2	Print selected records and drop all other records, formatting the output from a template.
FMNSAMRS	Perform a simple record selection, and copy the selected records to a file.
FMNSAMSP	Split selected records to two output files.
FMNSAMU1	Make selective updates in a member of a data set, in place, using function DSEB and REXX.
FMNSAMU2	Make selective updates in a member of a data set, in place, using function DSU and FASTREXX.

You will have to make changes to these jobs before you can run them. Refer to the sample jobs for information about the changes you need to make.

Complete the installation

You can now complete the installation of File Manager by performing the ACCEPT processing. The steps involved are described in the *File Manager Program Directory*.

Verifying the Japanese or Korean components

After you have completed all the necessary steps involved in the initial installation and customization of the File Manager Japanese or Korean components, you need to perform the following steps to verify your customization, before completing the installation process. You might need to refer to the *File Manager User's Guide and Reference*.

Note: The File Manager Japanese and Korean components both require the File Manager base function product to have already been installed.

1. Log on to a TSO userid that is enabled to access and use File Manager. Ensure the ISPF terminal type is set to 3277KN or 3278KN in ISPF option 0.
2. Start File Manager:
   - If you have added an option for File Manager to your ISPF Primary Options menu (see "Adding File Manager to the ISPF menu" on page 11), enter the option value you have assigned to File Manager and press Enter. For example, if you have assigned 'F' to File Manager, enter 'F' and press Enter.
   - If you defined File Manager to the ISPF command table (see "Defining File Manager in an ISPF command table" on page 11), verify that File Manager can be started by entering the command 'FM' on any ISPF command line.

   The English or Japanese (or Korean) File Manager Primary Option Menu is displayed. depending on the default defined.

   Note: The first time you use File Manager, a Copyright panel appears. After reading the panel text, press the Cancel key (PF12). In subsequent File Manager sessions, this panel will not automatically appear.
3. If the English menu appears:

- Check the ISPPLIB concatenation and ensure that the Japanese or Korean panel library (FMN.SFMNPJPN or FMN.SFMNPKOR) is allocated in front of the English panel library (FMN.SFMNPENU).
- Check the ISPMLIB concatenation and ensure that the Japanese or Korean message library (FMN.SFMNMJPN or FMN.SFMNMKOR) is allocated in front of the English message library (FMN.SFMNMENU).

4. Enter `ver` on the command line to display the release level and PTF level of File Manager. If you are verifying the Japanese component, a panel is displayed similar to this:

```
IBM File Manager for z/OS Version 9 Release 1

(not APF authorized)

Service Levels of installed components

             Base       IMS        DB2        CICS
English      -NONE-     -NONE-     -NONE-     -NONE
Japanese     -NONE-     -NONE-     -NONE-     -NONE-
```

   If you are verifying the Korean component, a panel is displayed similar to this:

```
IBM File Manager for z/OS Version 9 Release 1

(not APF authorized)

Service Levels of installed components

             Base       IMS        DB2        CICS
English      -NONE-     -NONE-     -NONE-     -NONE
Korean       -NONE-     -NONE-     -NONE-     -NONE-
```

   Check that the information displayed conforms with the level of File Manager that you are verifying. If you have not installed a component in Japanese or Korean (for example, you have not installed the FM/IMS Japanese component), that component will not be shown at all.

5. Press the Exit key (PF3) to end the File Manager session and return to your ISPF menu.

Complete the installation

You can now complete the installation of File Manager by performing the ACCEPT processing. The steps involved are described in the *File Manager Program Directory*.

Verifying the Japanese or Korean components

Part 2. Customizing File Manager DB2 Component

© Copyright IBM Corp. 2000, 2011

Chapter 12. Preparing to customize FM/DB2

Before you can install and customize FM/DB2 you must have installed the File Manager base function. This and the following chapters assume you have installed FM/DB2 into the same target and distribution libraries as File Manager base function. You could encounter problems using File Manager if you do not install FM/DB2 in the same libraries as the base function.

Furthermore, before you can use FM/DB2 you must customize FM/DB2 and the operating environment. You may also need to customize File Manager base function. Part 1, "Customizing File Manager," on page 1 describes the customization of File Manager base function.

FM/DB2 requires access to various DB2 catalog tables in order to function correctly. Granting access to DB2 resources is a DB2 task, not a File Manager task. The configuration of DB2 security/authorization is described in the *DB2 Administration Guide*. If your site has a DB2 Administrator and/or Security Administrator, you should consult with them before proceeding. Only general information is provided on how you might allow FM/DB2 users SELECT access to the DB2 catalog tables. You should carefully consider your site's security requirements and how DB2 security is implemented and administered when selecting the most appropriate method for your installation. For example, many sites have a standard that SELECT access on some, or all, of the DB2 catalog tables must not be granted to PUBLIC. If, after careful consideration of your site's security requirements and FM/DB2's requirement for access to the DB2 catalog, you are unable to provide FM/DB2 user's with the appropriate level of access, **you should not attempt to install and use FM/DB2**.

There are five customization tasks you **must** perform for FM/DB2. These are:

- Adding the FMN2PARM member to SYS1.PARMLIB, described in "Defining the FMN2PARM member" on page 100.
- Granting access to the DB2 catalogs, described in "Granting access to the DB2 catalog (required)" on page 109.
- Binding the FM/DB2 packages and plans for DB2, described in "Binding DB2 (required)" on page 119.
- Creating a segmented table space in a TEMP database, described in "Creating a segmented table space (required)" on page 121.
- Customizing the FM/DB2 subsystem selection panel for DB2, described in "Identifying all the DB2 systems that FM/DB2 will access and defining these in the FMN2POPT module (required)" on page 121.

You will not be able to use FM/DB2 if you have not performed these tasks.

Table 19 on page 100 lists these and all the other customization tasks you can perform for FM/DB2. Read the referenced sections to see if you need to perform the customization described.

© Copyright IBM Corp. 2000, 2011

Checklist for installing and customizing FM/DB2

Table 19. Summary of steps for customizing FM/DB2 and the operating environment

	Description	Page
__ 1	Add FMN2PARM to SYS1.PARMLIB - **required**.	100
__ 2	Concatenate FM/DB2 libraries to the LINKLIST.	102
__ 3	Modify the TSO logon procedure.	102
__ 4	Authorize FM/DB2.	103
__ 5	Customize to improve FM/DB2 performance.	103
__ 6	Add FM/DB2 to the ISPF menu.	107
__ 7	Define FM/DB2 in an ISPF command table.	107
__ 8	Customize the FM/DB2 Primary Option Menu.	108
__ 9	Grant access to the DB2 catalogs - **required.**	109
__ 10	Add FM/DB2 to the DB2 Admin Launchpad	112
__ 11	Add commands to start FM/DB2 functions from DB2 Admin Tool	113
__ 12	Bind DB2 - **required**.	119
__ 13	Set up to run multiple versions of FM/DB2.	120
__ 14	Create a segmented table space in a TEMP database - **required**.	121
__ 15	Identify all the DB2 systems that FM/DB2 will access - define these in the FMN2POPT module - **required**.	121
__ 16	Customize FM/DB2 options.	125
__ 17	Change the default options.	126
__ 18	Change the batch JCL skeletons.	127
__ 19	Customize to protect update functions.	129
__ 20	Decide how to customize the FM/DB2 audit facility.	104
__ 21	Customize FM/DB2 for national languages.	151

Defining the FMN2PARM member

If auditing is to be controlled from parmlib (user has read access to FILEM.PARMLIB.DB2, see "Controlling File Manager auditing using SAF (FM/DB2 component only)" on page 135) then member FMN2PARM must be defined in SYS1.PARMLIB (or any other library in the logical parmlib concatenation) as follows.

Default parmlib member FMN2PARM is provided in the SFMNSAM1 library. Copy this member to the appropriate system parmlib library. See below for details of methods that can be used to make this change.

Note: The sample FMN2PARM member supplied in SFMNSAM1 also includes an FMSECRTY statement. This option is not used in FM/DB2 and can be either omitted or commented out. It has no effect.

File Manager, DB2 component, will not start if the required parmlib member does not exist in the system logical parmlib concatenation for users to be audited using parmlib.

There are two methods that can be used to include the FMN2PARM member in a library in the logical parmlib concatenation. The choice of method depends on whether the installation's security software is configured to allow FM/DB2 users READ access to data set SYS1.PARMLIB.

Method 1 can only be used when FM/DB2 users have read access to all libraries in the logical parmlib concatenation.

Method 2 can be used regardless of whether FM/DB2 users have READ access to the libraries in the logical parmlib concatenation.

Method 2 must be used when FM/DB2 users do not have READ access to one or more libraries in the logical parmlib concatenation.

Method 1

Place the FMN2PARM member in any library in the current logical parmlib concatenation. No IPL or other action is required to active the new member (unless a new library was added to the logical parmlib concatenation).

Note: FMN0POPT controlled auditing cannot be used in any situation where FM/DB2 users do not have READ access to all of the libraries in the logical SYS1.PARMLIB concatenation.

For example, when:

- There are six libraries in the logical parmlib concatenation, for simplicity: libraries A, B, C, D, E and F.
- FM/DB2 users have read access to five of these libraries: A, B, D, E, F.
- Library C may be SYS1.PARMLIB, or any other library in the logical SYS1.PARMLIB concatenation.

This will not work, the attempt by an FM/DB2 user to access the logical SYS1.PARMLIB concatenation will fail with a security-related (913) abend.

Method 2

This method must be used when FM/DB2 users do not have READ access to all of the libraries in the logical parmlib concatenation.

1. Create a new library with dataset attributes similar to SYS1.PARMLIB.
2. The library name for this data set must include the string "FMNPARM" in one of the qualifiers. You are free to choose any data set name that meets this requirement. Examples of suitable data set names are:

   SYS1.PARMLIB.FMNPARM
   SYS8.FMNPARM.PARMLIB
   FMNPARM.SYS8.PARMLIB
   SYS2.FMNPARMS.LIB
   SYS8.XFMNPARM.PARMLIB
3. Add member FMN2PARM to the new library, specifying the appropriate FMAUDIT parameter.
4. Add the new library to the logical parmlib concatenation. This can be done dynamically, or via a system IPL.

Note: When Method 2 is used, the FMN2PARM member must be located in the library created in point 1 (as described above). In addition,

should the FMN2PARM member specify any include statements (see "Facilities for customizing the FMN2PARM definitions" on page 424), all the included members must also reside in the library created in point 1 (as described above).

You use the FMN2PARM member to define the following:
- Whether File Manager will use SAF to control File Manager audit logging.
- The SAF resource name prefix to be used by File Manager when determining access to various resources.
- Whether File Manager should load the FMN2POPT module from a specific library.

See "Deciding how to customize the FM/DB2 Audit facility" on page 104 for more information.

Alternatives for making FM/DB2 available

You can make FM/DB2 available to your users either by concatenating FMN.SFMNMOD1 to your LINKLIST or adding it to the STEPLIB DD statement in your TSO logon procedure.

To make FM/DB2 readily available from ISPF, configure your ISPF environment as described in Chapter 2, "Customizing the operating environment for File Manager," on page 11.

Concatenating libraries to the LINKLIST

To make FM/DB2 commonly available, add the FMN.SFMNMOD1 library to your concatenated LINKLIST. If you did not do this for File Manager base function, add this library to either your LNKLST*xx* or PROG*xx* member in SYS1.PARMLIB.

You can also choose to add your DB2 load and run libraries to your LINKLIST. If you do this, you might not need to specify these libraries in your FMN2SSDM macros. See "DB2LLIB" on page 335 and "DB2RLIB" on page 337.

Modifying the TSO logon procedure

If you made File Manager base function available to TSO, then you do not need to make any further changes to TSO for FM/DB2. Otherwise, add the File Manager libraries to your TSO logon procedure, as described in "Modifying the TSO logon procedure" on page 7.

Using LIBDEFs to allocate the File Manager Libraries

If you do not want to add the File Manager libraries to your default logon procedure, you can use an initialization CLIST or REXX exec to allocate the required libraries, using LIBDEFs. A sample initialization REXX exec, FMN2INIT, to allocate libraries for FM/DB2 is distributed in FMN.SFMNSAM1.

FMN2INIT allocates the File Manager libraries to the required concatenations. It assumes that you have installed File Manager into the default target libraries. If you have not done this, and you want to use FMN2INIT to allocate your libraries, you will need to modify it to use your libraries. Change **ONLY** the statements that reference the File Manager libraries, for example:

```
sfmnllib = "'FMN.SFMNMOD1'"
```

Do **NOT** change or remove any other statement. In particular do not change or remove the statement:

```
address TSO "ALTLIB ACT APPLICATION(EXEC)  DATASET("sfmnexec")"
```

If you choose to use a CLIST or a REXX exec to dynamically define the required FM/DB2 libraries using LIBDEFs, also copy the REXX exec FMN2RESS from FMN.SFMNEXEC to a library that is defined in the SYSPROC or SYSEXEC concatenation at the time ISPF is started. If possible, this library should be the same library that contains the EXEC or CLIST that you initially use to start FM/DB2.

Note: Do not place FMN2RESS in a library defined by the TSO ALTLIB command. FMN2RESS is used during the invocation of a second (or subsequent) FM/DB2 edit session (related edit). The invocation of the related edit session will fail if this exec cannot be accessed at related edit invocation time. Libraries defined by the TSO ALTLIB command are not propagated to the second ISPF session started for the related edit, so it is not sufficient to place this exec in a library defined by the ALTLIB command.

Ensuring the DB2 libraries are available to FM/DB2

FM/DB2 requires access to the same DB2 libraries as the DB2I online utility.

You can allocate the DB2 libraries required by FM/DB2 using one of the following methods:

- You can include the DB2 libraries in your TSO logon procedure; or in a CLIST or EXEC run as part of the TSO logon procedure; or in a CLIST or EXEC executed prior to the initialization of FM/DB2. This method is not appropriate if FM/DB2 will be connecting to DB2 systems at different version/maintenance levels. If you use this method you might not need to specify the DB2 libraries in the FMN2SSDM macros, but you will still need to review the contents of the FMN2POPT options module. See “Changing the default options” on page 126.
- Specify the DB2 libraries in the FMN2SSDM macro entry for the DB2 system. This is the preferred method and allows FM/DB2 to connect to DB2 systems at different version/maintenance levels. The DB2 libraries are allocated as part of the FM/DB2 connect process, and de-allocated at the end of the FM/DB2 session, or when FM/DB2 connects to a different DB2 system. See “FMN2SSDM” on page 331.

Planning for running FM/DB2 with APF-authorization

If you intend to use SMF for recording audit trail information for FM/DB2, then you must ensure the File Manager load module, FMNSMF, is APF-authorized. If you have not made File Manager base function APF-authorized, then either add FMN.SFMNMOD1 to your list of authorized libraries, or copy FMNSMF into an authorized library.

Improving FM/DB2 Performance

DB2 does not have an index defined on the SYSIBM.SYSFOREIGNKEYS catalog table. FM/DB2 will access the SYSFOREIGNKEYS catalog table whenever it generates a template for a DB2 object, unless you specify that it should not. To avoid the possibility of poor performance when generating FM/DB2 templates, particularly if you have a large number of foreign keys, turn off access to the SYSFOREIGNKEYS catalog table.

You do this by means of one of the FM/DB2 global options. For more information about setting the FM/DB2 global options, see *File Manager User's Guide and Reference for DB2 Data*. Alternatively, you can create a user-defined index on the SYSIBM.SYSFOREIGNKEYS catalog table. The non-unique index should specify the following columns (in order):

1. CREATOR
2. TBNAME

The following sample DDL can be used to define the index:

```
CREATE INDEX nnnnnn.SYSFKIX                    1
         ON SYSIBM.SYSFOREIGNKEYS
        (
         CREATOR
        ,TBNAME
        )
        USING STOGROUP mmmmmm                  1
              PRIQTY pp                        2
              SECQTY ss                        2
```

1 change the name of the index, and of the storage group, *nnnnnn* and *mmmmmm*, to suit your requirements.

2 change the primary and secondary quantities, *pp* and *ss*, to suit your requirements.

A sample job, FMN2CIX, is distributed in FMN.SFMNSAM1 to help you do this.

Once the index has been successfully created, execute the RUNSTATS utility against DSNDB06.SYSDBASE. See your site's DB2 Systems Administrator for additional information.

Deciding how to customize the FM/DB2 Audit facility

You have two choices with respect to auditing of FM/DB2 audit activities. These are:

1. Use FMN2POPT controlled auditing. This relies on the AUDIT parameter specified in the FMN2SSDM macro for each DB2 system that is accessed by FM/DB2. See Appendix B, "FM/DB2 options," on page 331 for a description of the FMN2SSDM macro parameters.

   The following points summarize the facilities available with FMN2POPT controlled auditing:
   - Different audit settings can be specified for each DB2 system that FM/DB2 might access
   - Auditing occurs for key FM/DB2 functions only (edit, copy, import). Other FM/DB2 functions are not subject to audit (for example print, export and issuing DB2 commands).
   - The audit settings for any DB2 system apply equally to all FM/DB2 users connected to that DB2 system.
   - The audit settings for any DB2 system apply equally to all DB2 objects within that DB2 system.
   - The "Create audit trail" option is visible to the user for those FM/DB2 functions where auditing might occur.
   - You can specify auditing to the user's audit log data set, to the user's audit log data set with automatic (mandatory) printing of the audit log at the completion of the session, or to SMF.
2. Use SAF-rule controlled auditing. This relies on various SAF FACILITY and XFACILIT resource rules, which you define with an external security product, such as RACF (or equivalent product).

The following points summarize the facilities available with SAF-rule controlled auditing:

- Different audit settings can be specified for each DB2 system that FM/DB2 might access.
- Auditing can be (optionally) specified for all FM/DB2 functions, however audit records are not written for the SQL statements used by FM/DB2 to build templates, or for other internal purposes.
- Different auditing requirements can be specified for different TSO userids.
- Different auditing requirements can be specified for access to different DB2 objects, or to different types of SQL statements, or when issuing DB2 commands.
- You can provide FM/DB2 users with a "Create audit trail" option for some FM/DB2 functions. This is also SAF-rule controlled. The presence of the "Create audit trail" option does not guarantee that the user can switch off auditing, since this depends on the level of access the user has to the appropriate SAF resource names. When a user has access to the "Create audit trail" option, they can always turn on auditing, even if the relevant SAF resource rules do not require auditing.
- You can specify auditing to the user's audit log data set, to the user's audit log data set with automatic (mandatory) printing of the audit log at the completion of the session, or to SMF. Dual logging (to the user's audit log data set and to SMF) can also be specified.

Some other points to consider are:

- Auditing to the user's audit log data set can result in large numbers of audit log data sets, and this may have disk space implications. You may need to consider implementing automatic purging or archiving of audit log data sets.
- Auditing to SMF (only) requires additional set-up, but provides a more reliable and secure environment for capturing audit information than audit logging to the user's audit log data set.
- With SAF-rule controlled auditing in effect to SMF only, an FM/DB2 user can determine that SAF-rule controlled auditing is in effect, but there is no indication to the user which activities are actually being audited.
- With SAF-rule controlled auditing in effect, failure to write records to the SMF audit log - when this is required - will result in the termination of the current FM/DB2 function.

If you determine that SAF-rule controlled auditing is not appropriate, follow the instructions in Chapter 15, “Customizing the FM/DB2 audit facility (FMN2POPT controlled auditing),” on page 131.

If you determine that SAF-rule controlled auditing is appropriate, follow the instructions in Chapter 16, “Customizing the FM/DB2 audit facility (SAF-rule controlled auditing),” on page 135.

Chapter 13. Customizing the operating environment for FM/DB2

This chapter describes how to customize the operating environment for FM/DB2. You do this after you have installed FM/DB2.

Modifying the ISPF environment

To make it easy to start FM/DB2 under ISPF, configure your ISPF environment as described in the following sections.

Adding FM/DB2 to the ISPF menu

To add FM/DB2 to your ISPF Primary Option Menu panel (ISR@PRIM), insert the additional lines (◄ **New**) as shown in Figure 7. You could add FM/DB2 in your Primary Option Menu after the File Manager base function.

```
   ⋮
)BODY  CMD(ZCMD)
   ⋮
 9  IBM Products  IBM program development products
 10 SCLM          SW Configuration Library Manager
 11 Workplace     ISPF Object/Action Workplace
 F  File Manager  File Manager for z/OS
 FD FM/DB2        File Manager/DB2            ◄ New
   ⋮
)PROC
   ⋮
 &ZSEL = TRANS( TRUNC (&ZCMD,'.')
   ⋮
  9,'PANEL(ISRDIIS) ADDPOP'
 10,'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
 11,'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
  F,'PANEL(FMNSTASK) SCRNAME(FILEMGR) NEWAPPL(FMN)' /* File Manager */
 FD,'PANEL(FMN2ST00) SCRNAME(FMDB2) NEWAPPL(FMN2)' /* FM/DB2       */ ◄ New

   ⋮
```

Figure 7.

For more information about configuring your ISPF Primary Option Menu panel, see *z/OS ISPF Planning and Customizing*.

Defining FM/DB2 in an ISPF command table

ISPF supports four different command tables where you can define an ISPF command to invoke FM/DB2:

- Application command table
- User command table
- Site command table
- System command table

You can use the ISPF Command Table Utility (ISPF option 3.9) to create or change a command table that is not currently in use (the system command table,

© Copyright IBM Corp. 2000, 2011

ISPCMDS, is always in use). When you add a command for FM/DB2 to one of these command tables, you can invoke FM/DB2 from any ISPF panel without prefixing the command with TSO.

Add the following entry to a command table to enable FM/DB2 to be run from any ISPF panel by typing 'FD' on the command line.

Verb: FD

Action: SELECT PANEL(FMN2ST00) OPT(&ZPARM) SCRNAME(FMNDB2) SUSPEND NEWAPPL(FMN2)

Description: File Manager DB2 Component

For information about ISPF command tables, see *z/OS ISPF Planning and Customizing*. For information about the ISPF Command Table Utility, see *z/OS ISPF User's Guide Vol II*.

Customizing the FM/DB2 Primary Option Menu

This customization step is optional. Review this information to determine if you need to perform this step.

The FM/DB2 Primary Option Menu panel (FMN2ST00) provides an option to select DB2 Interactive (DB2I). If DB2I is selected from the FM/DB2 Primary Option Menu, the exec FMN2DB2I is called.

If you have customized the method used to start DB2I, for example, if your site displays a DB2 authorization panel before starting DB2I, then you may need to modify the FMN2ST00 panel, or the FMN2DB2I exec.

The information provided here is not specific to any particular method of customization that may be required, owing to the wide scope of possible customizations.

Panel FMN2ST00, as supplied, includes the following:

```
&ZSEL = TRANS(TRUNC(&ZCMD,'.')
  0,'PANEL(&FMN2STSC)'
  1,'CMD(FMNDB2 D2B) MODE(FSCR)'
  2,'CMD(FMNDB2 D2E) MODE(FSCR)'
  3,'PANEL(FMN2STUT)'
  4,'PANEL(FMN2STSS)'
  5,'CMD(FMN2DB2I SSID(&FMN2SSIS)) MODE(FSCR)'    ◄ DB2I
  X,'EXIT'
  ' ',' '
```

DB2I indicates where FMN2DB2I is called when Option 5, DB2I, is selected.

The exec FMN2DB2I, as supplied, allocates a data set to DDNAME DSNETBLS, before invoking DB2I using the following ISPF SELECT statement:

```
"SELECT CMD(DSNECPRI SSID("ssid")) NEWAPPL(DSNE) PASSLIB MODE(FSCR)"
```

You may need to customize either the panel FMN2ST00, or the exec FMN2DB2I, if your site has a customized exec or panels called prior to starting DB2I.

Notes:

1. FM/DB2 provides two methods for allocating the File Manager libraries. See "Modifying the TSO logon procedure" on page 102 for further information. You must ensure that any processing in your site's DB2 invocation procedure does not de-allocate the FM/DB2 libraries.
2. If your site's DB2 customization uses LIBDEF statements to allocate the required DB2 libraries, FM/DB2 allocates the required DB2 libraries (as defined in the FMN2SSDM macro for the connected DB2 system) at connect time: that is, when FM/DB2 is first started, or whenever the DB2 SSID is changed. FM/DB2 is connected to the DB2 system shown against "DB2 SSID" on the Primary Option Menu panel, FMN2ST00. This means that it is unnecessary for the DB2I invocation procedure to allocate the DB2 libraries prior to starting DB2I. You can determine the current LIBDEF allocation by starting FM/DB2 and issuing the ISPLIBD command on the command line.
3. FMN2ST00 is distributed in FMN.SFMNPENU. FMN2DB2I is distributed in FMN.SFMNEXEC. If you decide to modify these or any other FM/DB2 parts to complete your customization for invoking DB2I, create a usermod to do this, to keep track of the changes to SMP/E controlled parts. For information about modifying panels, refer to the *z/OS ISPF Dialog Developer's Guide*. For information about modifying execs, refer to the *z/OS ISPF Services Guide*.

Granting access to the DB2 catalog (required)

FM/DB2 uses dynamic SQL, issued against the DB2 catalog, as part of its processing. To make FM/DB2 available and to ensure correct and optimum operation, you must ensure that each FM/DB2 user has SELECT access against the DB2 catalog tables. **This is a non-negotiable requirement.**

DB2 authorization configuration can only be achieved using DB2, or an external security server (or both). Since these are external products, only general guidance is provided here on the authorization and security issues associated with the use of FM/DB2. You must determine the best approach to providing the required level of access for FM/DB2 users, based on your installation's unique requirements. You must ensure SELECT access against the DB2 catalog tables is given for all DB2 systems that are accessible to FM/DB2 users.

FM/DB2 requires the use of DYNAMICRULES(RUN) when the FM/DB2 plan is bound. See "Binding DB2 (required)" on page 119 for further information. The effect of DYNAMICRULES(RUN) behaviour is that FM/DB2 uses the *run behaviour* for dynamic SQL statements, as summarized here:

- DB2 uses the authorization ID of the application process and the SQL authorization ID (the value of the CURRENT SQLID special register) for authorization checking of dynamic SQL statements.
- DB2 uses the authorization ID of the application process and the SQL authorization ID (the value of the CURRENT SQLID special register) as the implicit qualifier of table, view, index, and alias names.
- Dynamic SQL statements use the values of application programming options that were specified during installation. The installation option USE FOR DYNAMICRULES has no effect.
- GRANT, REVOKE, CREATE, ALTER, DROP, and RENAME statements can be executed dynamically.

The important point here is that the authorization ID of the application process is used for authorization checking of dynamic SQL statements, including access to the DB2 catalog tables.

In a simple DB2 installation, the user's TSO logonid is used as the DB2 authorization ID, so SELECT access on the appropriate DB2 catalog tables would need to be granted to a list of DB2 authids, which would be the same as the list of the TSO logonids used by FM/DB2 users. You can achieve the same result by granting SELECT access on the DB2 catalog tables to PUBLIC.

In a more complicated DB2 installation, an external security server may be used to control access to DB2 resources, and a DB2 authorization exit may be used to convert a user's TSO logonid into a primary and one or more secondary DB2 authids. See the *DB2 Administration Guide* for the appropriate version of DB2 for more information.

When an external security server is used to manage DB2 authorization, and a DB2 authorization exit is available, one suggested approach to granting FM/DB2 users SELECT access to DB2 catalog tables is as shown here:

- Determine an otherwise unused, generic DB2 authid for use by FM/DB2 users. For example, FMUSER. You can select any DB2 authid as required.
- Use the FM/DB2 sample jobs (see "Sample jobs to grant SELECT access on the DB2 catalog tables") to grant SELECT access on the DB2 catalog tables to the selected DB2 authid. You must modify the sample jobs to achieve this.
- Configure the security software and DB2 authorization exit to:
  1. Validate that the user has authority to use the FMUSER DB2 authid.
  2. Convert the user's TSO logonid into DB2 authid FMUSER during the connection to DB2. This means that authorization checking for dynamic SQL statements issued by FM/DB2 are made using DB2 authid FMUSER, rather than the user's TSO logonid.
- As an additional security measure, grant EXECUTE on the FM/DB2 plans to the generic DB2 authid, rather than PUBLIC.

When implemented, the above approach can be used to ensure:

1. SELECT access on the DB2 catalog tables is not granted to PUBLIC.
2. Only the generic DB2 authid need have SELECT access on the DB2 catalog tables.
3. The external security system can be used to validate and control which users are allowed access to the generic DB2 authid, and hence to FM/DB2 functionality.
4. Individual FM/DB2 users may have no direct access - using a DB2 authid that is the same as their TSO logonid - to the DB2 catalog tables.
5. Only those users with access to the generic DB2 authid can execute the FM/DB2 plans.

Sample jobs to grant SELECT access on the DB2 catalog tables

These jobs are provided in FMN.SFMNSAM1 to grant SELECT access on the DB2 catalog tables to PUBLIC:

1. FMN2GSC8 (DB2 V8)
2. FMN2GSC9 (DB2 V9)
3. FMN2GVW8 (DB2 V8)

4. FMN2GVW9 (DB2 V9)
5. FMN2GV28 (DB2 V8)
6. FMN2GV29 (DB2 V9)

Sample jobs FMN2GSC8 and FMN2GSC9 grant SELECT access on the DB2 catalog tables to PUBLIC. This is the simplest method of ensuring FM/DB2 users have access to the DB2 catalog tables. It is also the least secure.

Sample jobs FMN2GVW8 and FMN2GVW9:

- Create views on the DB2 catalog tables with a different owner, but the same name as the DB2 catalog tables
- Grant access on the newly created views to PUBLIC.

If you intend to use these views, you must set the CATOWNER parameter in the FMN2POPI to match the value of the owner specified in these sample jobs.

For information about FMN2POPI, see "FMN2POPI" on page 348. For information about changing the options in FMN2POPT, see "Changing the default options" on page 126.

Sample jobs FMN2GV28 and FMN2GV29:

- Create views on the DB2 catalog tables with a different owner, but the same name as the DB2 catalog tables.
- These views include all the DB2 catalog tables normally accessed by FM/DB2. Some DB2 catalog tables are not referenced and therefore no views for these DB2 catalog tables are defined.
- Not every column of the referenced DB2 catalog tables is included in the views. The columns that are included are required for the correct functioning of FM/DB2. Omitting additional columns may impact the functioning of FM/DB2, in the worst case rendering the product unusable.
- Grant access on the newly created views to PUBLIC.

If you intend to use these views, you must set the CATOWNER parameter in the FMN2POPI to match the value of the owner specified in these sample jobs.

For information about FMN2POPI, see "FMN2POPI" on page 348. For information about changing the options in FMN2POPT, see "Changing the default options" on page 126.

Notes:

1. If you create views of the DB2 catalog tables and change the CATOWNER parameter in the FMN2POPI macro, the following example outlines how FM/DB2 attempts to access the DB2 catalog. In the example, it is assumed that the owner of the views (CATOWNER value) is SYSIBMV.
   - FM/DB2 generates an SQL statement like SELECT * FROM **SYSIBMV**.SYSTABLES.
   - FM/DB2 attempts to execute the SQL statement.
   - If the execution fails (for whatever reason), FM/DB2 does NOT attempt to access the DB2 catalog using, for example, SELECT * FROM SYSIBM.SYSTABLES.
   - In summary, FM/DB2 attempts to access the DB2 catalog once only, using the specified CATOWNER value as the owner for the relevant catalog table(s).

2. If you do not want to grant SELECT access against the DB2 catalog tables or views to all users (that is, to PUBLIC), you can customize the GRANT statements in the sample jobs to list individual userids.
3. In situations where you do not want to expose all the information in the DB2 catalog tables, you can define views (a "minimal subset") that refer only to the DB2 catalog tables, and columns within those tables, that are needed by FM/DB2. Sample jobs are provided in FMN.SFMNSAM1 to do this, for DB2 versions 8 and 9. They are:
   - **FMN2GV28** for DB2 version 8 subsystems.
   - **FMN2GV29** for DB2 version 9 subsystems.

   The view definitions in these sample jobs refer only to the columns needed by FM/DB2, and not all catalog tables are included. There are consequences to using views that include only some columns of the actual DB2 catalog table. For example, option 3.4 of FM/DB2 provides an option to show all columns of certain catalog tables. The columns shown reflect the columns of the view and not necessarily all columns of the catalog table, reducing the usefulness of the product.
4. If your installation's security standards prevent access to the SYSIBM.*AUTH tables, you may omit these views, or create the views and then not grant access. This has the effect of disabling the Object Privileges utility (Option 3.5).
5. You can define different views for different DB2 subsystems. For example, you could define the views in FMN2GVW8 against a development system, but the views in FMN2GV28 against a production system. You must ensure, however, that the owner (CATOWNER value) used is consistent across all DB2 subsystems accessed by FM/DB2.
6. When you define views on the DB2 catalog table for use by FM/DB2, you cannot change the object name of the catalog table; you can only change the owner. For example, defining a view **SYSIBMV.SYSTABLES** on the table **SYSIBM.SYSTABLES** is valid. However, defining a view **SYSIBM.TABLES** on the table **SYSIBM.SYSTABLES** is invalid. You cannot rename the columns of the catalog tables in the views you define for use by FM/DB2. The owner (CATOWNER value) of the views you create must be the same for every view.

Adding FM/DB2 to the DB2 Administration Launchpad

You can add FM/DB2 to the list of applications on the DB2 Admin Launchpad - see *DB2 Administration Tool for z/OS User's Guide and Reference*, Chapter 3, for a description of the DB2 Admin Launchpad and how it is customized.

You need to complete the following tasks:

1. Define a Launchpad entry for FM/DB2 using the instructions in the DB2 Administration Tool manual. Use the values shown in Table 20 on page 113.
2. Locate the sample exec, FMN2ADIN, (distributed in FMN.SFMNSAM1.)
3. Copy FMN2ADIN to a library that is allocated to SYSEXEC DD NAME when the DB2 Admin Launchpad is active. This could be the SADBEXEC library.
4. Customize the sample member, FMN2ADIN, in your library by changing the names for the File Manager libraries in the following statements:

```
sfmnexec = "'FMN.SFMNEXEC'"
sfmnllib = "'FMN.SFMNMOD1'"
sfmnmlib = "'FMN.SFMNMENU'"
sfmnplib = "'FMN.SFMNPENU'"
sfmnslib = "'FMN.SFMNSLIB'"
sfmntlib = "'FMN.SFMNTENU'"
```

   to the appropriate values, if you are not using the default values.

Notes:

1. These instructions assume you will run FM/DB2 using LIBDEF/ALTLIB statements to allocate the File Manager application libraries.
2. If all the File Manager application libraries are available when the DB2 Admin Launchpad is active, for example, when the libraries have been allocated in the user's TSO logon procedure, you can remove all the statements to assign the File Manager libraries and to initiate and terminate the LIBDEFs and ALTLIB libraries. In this situation you should copy the sample exec to the File Manager exec (SFMNEXEC) library.

Table 20. Values for DB2 Admin Launchpad definition

Admin Tool Field	Value
PID	5655-U29
REL	912
NAME	File Manager DB2
CDE	FMN
GRP	Choose a value (1..4)
STAT	Y
CMD	SELECT MODE(FSCR) CMD(%FMN2ADIN)

Adding commands to start FM/DB2 functions from the DB2 Administration Tool

You can add one or more line commands to start the FM/DB2 editor and other FM/DB2 functions to the list of line commands that may be issued against a list of DB2 objects in DB2 Admin Tool. See the *DB2 Administration Tool for z/OS User's Guide and Reference*, Chapter 2 for a detailed description of how to do this. One reason to do this is to allow the FM/DB2 editor to be invoked for a DB2 object (table, view or alias) that appears in an Admin Tool object list.

You need to complete the following tasks:

1. Locate the sample exec FMN2ADLC (distributed in FMN.SFMNSAM1).
2. Review the codes and descriptions in the sample exec. You can change the line command code or description if required. Be careful to avoid conflicts with any existing line command codes in the Admin Tool table.
3. Follow the directions in the *DB2 Administration Tool for z/OS User's Guide and Reference* to add the commands to the list of available commands for the Admin Tool "Tables, Views and Aliases" display. Be sure to check that the appropriate Admin Tool tables are updated for the various types of tables that may be displayed by DB2 Admin Tool. See the *DB2 Administration Tool for z/OS User's Guide and Reference*.
4. Copy FMN2ADIE to a library that is allocated to SYSEXEC DD NAME when the DB2 Admin Launchpad is active. This could be the SADBEXEC library.
5. Customize sample member, FMN2ADIE, in your library by changing the names for the File Manager libraries in the following statements:

```
sfmnexec = "'FMN.SFMNEXEC'"
sfmnllib = "'FMN.SFMNMOD1'"
sfmnmlib = "'FMN.SFMNMENU'"
sfmnplib = "'FMN.SFMNPENU'"
sfmnslib = "'FMN.SFMNSLIB'"
sfmntlib = "'FMN.SFMNTENU'"
```

to the appropriate values, if you are not using the default values.

Notes:

1. These instructions assume you will run FM/DB2 using LIBDEF/ALTLIB statements to allocate the File Manager application libraries.
2. If all the File Manager application libraries are available when the DB2 Admin Tool is active, for example, when the libraries have been allocated in the user's TSO logon procedure, you can remove all the statements to assign the File Manager libraries and to initiate and terminate the LIBDEFs and ALTLIB libraries. In this situation you should copy the sample exec to the File Manager exec (SFMNEXEC) library.

When an FM/DB2 function is started via a DB2 Admin Tool line command, the following restrictions apply:

- You cannot issue a command to change the currently connected DB2 system.
- You cannot issue DB2 commands.
- Exiting the first panel displayed after the DB2 Admin Tool line command is issued returns to the DB2 Admin Tool Object List display. It is not possible to navigate to other parts of File Manager. You can, however, use the pull-down options to perform FM/DB2 functions.

Starting FM/DB2 functions from an external application

You can use the FM/DB2 exec FMN2INEX to start an FM/DB2 function from an external application such as a REXX exec or CLIST. FMN2INEX is distributed in FMN.SFMNEXEC. When using this interface you need to ensure the following:

- The File Manager libraries must be allocated prior to calling FMN2INEX.
- If you use LIBDEFs and ALTLIB statements to allocate the File Manager libraries, invoke FMN2INEX using the following command:

  ```
  "SELECT MODE(FSCR) CMD(%FMN2INEX "parms") NEWAPPL(FMN2) PASSLIB SUSPEND"
  ```

  This will ensure that the File Manager libraries are available to FMN2INEX.

The parameters that can be passed to FMN2INEX are shown in Table 21.

The function codes that can be specified are shown in Table 22 on page 115.

For example, referring to Table 21 and Table 22 on page 115, you might build the following parameter string,

```
SSID(DFE2) OW(DSN8910) NM(EMP) F(4.5)
```

and then your select statement would be:

```
"SELECT MODE(FSCR) CMD(%FMN2INEX SSID(DFE2) OW(DSN8910) NM(EMP) F(4.5)),
NEWAPPL(FMN2) PASSLIB SUSPEND"
```

Table 21. Parameters for FMN2INEX

Parameter	Description
SSID	The DB2 subsystem (or data sharing group) ID that FM/DB2 should connect to.
OW	The owner (creator) of a DB2 object that FM/DB2 should access. The owner is only passed if the function will accept an owner. Owner values are not passed to menu panels, or any option panel. See Table 22 on page 115 for additional exceptions.

Table 21. Parameters for FMN2INEX (continued)

Parameter	Description
NM	The name of a DB2 object that FM/DB2 should access. The name is only passed if the function will accept a name. Name values are not passed to menu panels, or any option panel. See Table 22 for additional exceptions.
F	• A code that indicates which FM/DB2 function should be executed. The code corresponds to the FM/DB2 main menu option used to select the desired function (eg 3.6 for import). • For editor functions only, if the function code has 'O' as a suffix, the editor function entry panel will be displayed. • For editor functions without the 'O' suffix, the first data displayed will be the data for the specified DB2 object.

Table 22. Function codes for FMN2INEX

Function code	Description
B	FM/DB2 Browse function. First panel displayed shows data for the specified DB2 object.
BO	As for B, except the editor function entry panel is the first panel displayed.
1	FM/DB2 View function. First panel displayed shows data for the specified DB2 object.
1O	As for 1, except the editor function entry panel is the first panel displayed.
2	FM/DB2 Edit function. First panel displayed shows data for the specified DB2 object.
2O	As for 1, except the editor function entry panel is the first panel displayed.
3	Displays the FM/DB2 Utilities menu.
3.1	FM/DB2 Print function. First panel displayed is the print function entry panel.
3.2	File Manager Object functions. First panel displayed is the Object Functions function entry panel. Any owner or name values are NOT passed.
3.3	FM/DB2 Copy function. First panel displayed is the copy (from) function entry panel.
3.4	File Manager Object List function. First panel displayed in the Object List function entry panel.
3.5	File Manager Object Privileges function. First panel displayed in the Object Privileges function entry panel.
3.6	FM/DB2 Import function. First panel displayed is the import (from) function entry panel.
3.7	FM/DB2 Export function. First panel displayed is the export function entry panel.
3.8	FM/DB2 Data Create function. First panel displayed is the data create function entry panel.
3.9	FM/DB2 Utilities function. First panel displayed is the DB2 utilities function entry panel. Any owner or name values are NOT passed.
3.10	FM/DB2 Audit Log function. First panel displayed is the audit log function entry panel. Any owner or name values are NOT passed.
4	Displays the FM/DB2 SQL Prototyping, Execution and Analysis menu.
4.1	File Manager Basic SQL Prototyping function. First panel displayed is the basic SQL prototyping function entry panel.

Table 22. Function codes for FMN2INEX (continued)

Function code	Description
4.2	File Manager Advanced SQL Prototyping function. First panel displayed is the advanced SQL prototyping function entry panel. Any owner or name values are NOT passed.
4.3	File Manager "Enter, Execute and Explain SQL Statements" function. First panel displayed is the "Enter, Execute and Explain SQL Statements" function entry panel. Any owner or name values are NOT passed.
4.4	File Manager "Edit and Execute SQL Statements from a Data Set" function. First panel displayed is the "Edit and Execute SQL Statements from a Data Set" function entry panel. Any owner or name values are NOT passed.
4.5	File Manager Explain Utilities function. First panel displayed is the explain utilities function entry panel. Any owner or name values are NOT passed.
6	File Manager Enter and Execute DB2 Commands function. First panel displayed in the enter and execute DB2 commands function entry panel. Any owner or name values are NOT passed.
0	Displays the FM/DB2 Options menu.
0.0	Displays the FM/DB2 System Options menu.
0.0.1	Displays the FM/DB2 Print Settings panel.
0.0.2	Displays the first FM/DB2 Systems Options panel.
0.0.3	Displays the FM/DB2 Batch Job Card information panel.
0.0.4	Displays the FM/DB2 Compiler Language selection panel.
0.0.5	Displays the FM/DB2 COBOL Processing Options panel.
0.0.6	Displays the FM/DB2 HLASM Processing Options panel.
0.0.7	Displays the FM/DB2 PL/I Processing Options panel.
0.0.8	Displays the FM/DB2 Temporary Data Set Allocations panel.
0.0.9	Displays the FM/DB2 Output Data Set Allocation Options panel.
0.0.10	Displays the FM/DB2 Trace Options panel.
0.1 (0.2)	Displays the first FM/DB2 Editor options panel.
0.3	Displays the FM/DB2 Utility Processing Options panel.
0.3.1	Displays the FM/DB2 Print Utility Options panel.
0.3.3	Displays the FM/DB2 Copy Utility Options panel.
0.3.4	Displays the FM/DB2 Object list utility options panel.
0.3.7	Displays the FM/DB2 Export utility options panel.
0.3.L	Displays the FM/DB2 DB2 LOAD utility options panel.
0.3.UL	Displays the FM/DB2 DB2 Utility LISTDEF options panel.
0.3.UO	Displays the FM/DB2 DB2 Utility OPTIONS options panel.
0.3.UT	Displays the FM/DB2 DB2 Utility TEMPLATE options panel.
0.3.UU	Displays the FM/DB2 DB2 Unload utility options panel.

When an FM/DB2 function is started using FMN2INEX, the following restrictions apply:

- You cannot issue a command to change the currently connected DB2 system.
- Exiting the first panel displayed by the FMN2INEX invocation returns to the caller. It is not possible to navigate to other parts of File Manager. You can however use the pull-down options to perform FM/DB2 functions.

These restrictions do not apply when no option code is specified for the FMN2INEX invocation.

Only the function codes listed in Table 22 on page 115 are supported. Any attempt to use any other code will result in the primary FM/DB2 menu being displayed; the erroneous function code will be discarded.

Chapter 14. Customizing FM/DB2

This chapter describes how to customize FM/DB2. You do this after you have installed FM/DB2. In particular, there are two tasks you **must** perform, "Binding DB2" and "Customizing the DB2 subsystem selection panel".

Binding DB2 (required)

Before you can use FM/DB2, you must bind the FM/DB2 packages and application plans, and grant EXECUTE privilege on these plans to FM/DB2 users. A sample job, FMN2BIND, is provided in FMN.SFMNSAM1 to help you do this. It is a requirement that FM/DB2 is bound using the DYNAMICRULES(RUN) bind option (the default). The use of DYNAMICRULES(BIND) disables FM/DB2's ability to dynamically execute SQL statements such as GRANT/REVOKE/ALTER and RENAME, and is therefore not recommended. The DEFINERUN and DEFINEBIND options of the DYNAMICRULES bind option are not supported. See the *DB2 Command Reference* for the appropriate version of DB2, section "Bind and Rebind options" for a detailed explanation of the DYNAMICRULES bind options.

In FMN2BIND, you will see that the members bound into the packages are called ADBxxxx. These members are aliases of DBRMs and are provided in FMN.SFMNDBRM.

Run the sample job, FMN2BIND, to bind the packages and plans, and to grant the DB2 EXECUTE privilege on the FM/DB2 plans to PUBLIC. If you do not grant EXECUTE privilege on the FM/DB2 plans to public, you will encounter DB2 authority errors, including SQLCODE-551 errors, when you attempt to run FM/DB2. As distributed, this sample job contains separate job steps for DB2 version 8 and 9 systems, and all the job steps are commented out.

You **must** run a version of this job for each DB2 subsystem you want FM/DB2 to use, either by direct or remote connection. Select and un-comment the job steps appropriate for these required DB2 systems. Refer to the comments in the job for information on modifications you need to make to the job to do this. A return code of 4 is acceptable from this job.

When there are multiple DB2 systems at your site, on possibly different LPARs or different physical machines, and the DB2 systems are configured so that remote access is possible from one DB2 system to another using the DB2 DDF facility, you must ensure that:

- You run a version of the FMN2BIND job against **every DB2 system that FM/DB2 will access**, either directly or remotely. The FMN2BIND job requires access to the DBRMs shipped with the product (in SFMNDBRM). You may need to copy this DBRM library so that the DBRMs are available to the FMN2BIND job when it is run. You need to do this for each LPAR or physical machine where a DB2 system, that FM/DB2 is to access, is located.
- The FMN2BIND job completes successfully; that is, the FM/DB2 plans and packages are bound correctly. See the comments in the FMN2BIND job, for the appropriate version of DB2, for acceptable return codes.

© Copyright IBM Corp. 2000, 2011

- EXECUTE privilege on the DB2 plans is successfully granted to PUBLIC. If this step is not successful it is possible that FM/DB2 will work correctly when connected to a local DB2 system, but fail when an attempt is made to access a remote DB2 system.

In FMN2BIND you can change the plan names for the FM/DB2 plan and the FM/DB2 reverse engineering plan. The default names for these plans are FMN2PLAN and FMN2GEN respectively. If you choose to change these plan names, you **must** change the values for `PLAN` and `PLAN2` to your chosen names, in your FMN2SSDM macros, when you define your DB2 subsystems to FM/DB2. For information about FM/DB2 and DB2 subsystems, see "Identifying all the DB2 systems that FM/DB2 will access and defining these in the FMN2POPT module (required)" on page 121. For information about the FMN2SSDM macro, see "FMN2SSDM" on page 331.

Running multiple versions of FM/DB2

If you intend to install multiple (different) versions of FM/DB2, you need to ensure that each FM/DB2 installation is **COMPLETELY INDEPENDENT** of the other(s).

In particular, you must ensure that:
- Different (and unique) plan and package names are used for each FM/DB2 installation.
- The FM/DB2 plan and packages are bound using the appropriate DBRM modules for each version of FM/DB2.
- Each version of FM/DB2 uses a unique FMN2POPT module, and that module must specify the correct plan name for that version of FM/DB2.
- When FM/DB2 is used to access remote DB2 systems, the plan and packages for each FM/DB2 version must be bound on every DB2 system that might be accessed using FM/DB2.

The notes below cover the situation where an existing version (say version n) of FM/DB2 is installed. A new version (say n+1) of FM/DB2 is available and will be installed. The two versions of FM/DB2 are to be run concurrently while acceptance testing is completed for the new version of FM/DB2.

The default plan names are used for the existing version of FM/DB2 that is already installed:
- FMN2PLAN
- FMN2GEN

Similarly, the default package names are used for the existing version of FM/DB2 that is already installed:
- FMN2PLPK
- FMN2GNPK

For the new version of FM/DB2, select new plan and package names, for example:
- FMN2PLNA (instead of FMN2PLAN)
- FMN2GENA (instead of FMN2GEN)
- FMN2PLPA (instead of FMN2PLPK)
- FMN2GNPA (instead of FMN2GNPK)

Note that any new plan or package names can be selected, as long as the new names are different to all other FM/DB2 plan and package names already in use.

For the new version of FM/DB2, modify the sample FMN2BIND job to specify the amended plan and package names, as shown above. Run the FMN2BIND job on every DB2 system that the new version of FM/DB2 will have access to, either via direct or remote connection. Ensure that the DBRM library specified in the new bind job is for the new version of FM/DB2.

Modify the sample FMN2POPT job for the new version of FM/DB2. Specify the new plan names in the FMN2SSDM macro definition for each DB2 system that FM/DB2 will access, either directly or remotely, as shown below:

```
FMN2SSDM  SSID=XXXX,PLAN=FMN2PLNA,PLAN2=FMN2GENA
```

Assemble the new FMN2POPT module, and ensure that this is available when a user executes the new version of FM/DB2.

For information about modifying FMN2POPT, see "Changing the default options" on page 126. For information about FMN2SSDM, see "FMN2SSDM" on page 331.

Creating a segmented table space (required)

FM/DB2 uses DECLARE GLOBAL TEMPORARY TABLE statements to implement some features of reverse engineering. Each DB2 subsystem that runs FM/DB2 must have a database that is defined as TEMP that contains a segmented table space to support the use of declared temporary tables. The page size of the segmented table space must be at least 8 KB.

After FM/DB2 is installed, you must create a database that is defined as TEMP and a segmented table space in this database, on each DB2 subsystem that runs FM/DB2, and that is not already set up to support the use of declared temporary tables.

A sample job, FMN2CGEN, is provided in FMN.SFMNSAM1, to help you do this. Sample statements are supplied to create the objects necessary to support declared temporary tables. You must run a version of this job for each DB2 subsystem you want FM/DB2 to use. Refer to the comments in the job for information about changes you may need to make to the job.

Identifying all the DB2 systems that FM/DB2 will access and defining these in the FMN2POPT module (required)

Before you can use FM/DB2, you **must** define to FM/DB2 the DB2 subsystems, or DB2 data sharing groups, which FM/DB2 will use.

To do this you must provide an FMN2SSDM macro for each required subsystem, to be included in the FM/DB2 options macro, FMN2POPT. If you are using DB2 data sharing groups, you must also provide an FMN2SSDM macro for each data sharing group id.

You must then install FMN2POPT on your FM/DB2 system, using the usermod FMN2UMDP. "FMN2SSDM" on page 331 describes the syntax of the options in FMN2SSDM. See "Changing the default options" on page 126 for information about other changes to make to FMN2POPT and how to install your version of FMN2POPT.

Each FMN2SSDM macro describes one subsystem or data sharing group. You can provide as many FMN2SSDM macros as you require in one FMN2POPT macro. You can specify an FMN2SSDM macro for any supported version of DB2 in the same FMN2POPT macro.

If your environment includes many DB2 subsystems, you can optionally specify whether FM/DB2 displays all DB2 subsystems on the subsystem selection menu, or only those systems that are currently active. You can also restrict which DB2 subsystems FM/DB2 will attempt to connect to. To do this, you specify an FMN2POPI macro and include it in FMN2POPT. "FMN2POPI" on page 348 describes the syntax of the options in FMN2POPI.

You can also optionally provide site-specific initial values for LISTDEF and TEMPLATE libraries, and an initial OPTIONS EVENT statement, which will be used in every batch utility job generated by FM/DB2. You can also specify default template names to be used in various clauses in several DB2 utility statements.

You provide these site-specific values, also, in an FMN2SSDM macro in FMN2POPT. For more information, see "FMN2SSDM" on page 331. For information on the use of the site-specific LISTDEF and TEMPLATE libraries, and the OPTIONS EVENT statement, refer to the *File Manager User's Guide and Reference for DB2 Data*.

See "Examples of FMN2SSDM macros" for sample code showing the FMN2SSDM macros you would code to define a DB2 subsystem, and a DB2 group attachment identifier for a data sharing group.

If your site uses DB2 data sharing groups, review the explanatory information for the LIST parameter of the FMN2POPI macro in "LIST" on page 355. This information explains how FM/DB2 determines the DB2 systems that are candidates for connection on the local z/OS server.

Examples of FMN2SSDM macros

The following sample code shows the FMN2SSDM macro statements you would code to define DB2 subsystems to FM/DB2, for DB2 version 8, and data sharing groups for DB2 version 9. You would code FMN2SSDM macros similar to these, and include them in your version of FMN2POPT. (Do not change FMN2SSDM in FMN.SFMNMAC1.) For an explanation of the statements see the notes at the end of the example.

```
         FMN2SSDM SSID=DF8A,PLAN=FMN2PLN8,PLAN2=FMN2GEN8,              +    DF8A
               DB2LLIB=('DB2V810.DFA2.SDSNEXIT','DB2.V810.SDSNLOAD'),  +
               DB2RLIB=('DB2V810.DFA2.RUNLIB.LOAD'),                   +
               DB2CLIB=('DB2.V810.SDSNCLST'),                          +
               DB2PROC=('DB2V810.DFA2.PROCLIB'),                       +
               TMPDDLN=DF81TMDD,                                       +
               STMJCL1='DISP=SHR,DSN=FIRST.TEMPLATE.LIBRARY',          +
               STMJCL2=DSN=SECOND.TEMPLATE.LIBRARY,                    +
               OPTEVT1='ITEMERROR,HALT',                               +
               CPYCPYN=DF81CPYC,                                       +
               LODINDN=DF81LDIN,                                       +
               DESC='DB2 Version 8.1 - Development system'
*
         FMN2SSDM SSID=DF8B,PLAN=FMN2PLN8,PLAN2=FMN2GEN8,              +    DF8B
               TYPE=SUBSYS,                                            +
               AUDIT=(OPTIONAL,OFF),                                   +
               DB2LLIB=('DB2V810.DF61.SDSNEXIT','DB2.V6R1M0.SDSNLOAD'),+
               DB2RLIB=('DB2V810.DF61.RUNLIB.LOAD'),                   +
```

```
             DB2PLIB=('DB2.V8R1M0.SDSNSPFP','DB2.V8R1M0.SDSNPFPE'),  +
             DESC='DB2 Version 8.1 - System B'
*
        FMN2SSDM SSID=DG01,PLAN=FMN2PLN8,PLAN2=FMN2GEN8,             +    DG01
             TYPE=GROUP,                                             +
             DISPLAY=YES,                                            +
             AUDIT=NONE,                                             +
             DB2LLIB=('DB2V910.DF72.SDSNEXIT','DB2.V9R1M0.SDSNLOAD'),+
             DB2RLIB=('DB2V910.DF72.RUNLIB.LOAD'),                   +
             DB2CLIB=('DB2.V9R1M0.SDSNCLST'),                        +
             DB2MLIB=('DB2.V9R1M0.SDSNSPFM'),                        +
             DB2PLIB=('DB2.V9R1M0.SDSNSPFP','DB2.V9R1M0.SDSNPFPE'),  +
             DB2SLIB=('DB2.V9R1M0.SDSNSPFS'),                        +
             DB2TLIB=('DB2.V9R1M0.SDSNSPFT'),                        +
             DESC='DB2 Group Attach ID number 1'
*
        FMN2SSDM SSID=DG02,PLAN=FMN2PLN8,PLAN2=FMN2GEN8,             +    DG02
             TYPE=GROUP,                                             +
             DISPLAY=YES,                                            +
             AUDIT=(REQUIRED,SMF),                                   +
             DB2LLIB=('DB2.V9R1M0.DF82.SDSNEXIT','DB2.V7R1M0.SDSNLOAD+
             '),                                                     +
             DB2RLIB=('DB2V910.DF82.RUNLIB.LOAD'),                   +
             DESC='DB2 Group Attach ID number 2'
        FMN2SSDM SSID=DEFAULT                                             DG02S
*
```

Note: In this example the + characters are in column 72.

DF8A

1. These statements define a DB2 version 8 subsystem, called `DB2 Version 8.1 - Development system`. The SSID is DF81.
2. The default values for PLAN and PLAN2 are not used, they have been customized to FMN2PLN8 and FMN2GEN8 respectively. Therefore this change must also be made in sample job FMN2BIND.
3. In DF81 the TYPE keyword is not specified, therefore it defaults to a type of SUBSYS.
4. The AUDIT keyword is not specified, therefore it defaults to `AUDIT=(OPTIONAL,PROFILE)`. The creation of an audit trail will be optional to the user, and FM/DB2 will store the current value of the `Create an audit trail` setting from the global options in the user's ISPF profile at the end of every FM/DB2 session.
5. The TMPDDLN keyword specifies a DDNAME for the site-specific template library of DF81TMDD, and the STMJCL1 and STMJCL2 keywords specify two concatenated data sets named

   ```
   FIRST.TEMPLATE.LIBRARY and SECOND.TEMPLATE.LIBRARY
   ```

   for this DDNAME. The JCL generated by these keywords will be:

   ```
   //DF81TMDD DD DISP=SHR,DSN=FIRST.TEMPLATE.LIBRARY
   //         DD DSN=SECOND.TEMPLATE.LIBRARY
   ```

   Note that the JCL in STMJCL1 is enclosed in single quotation marks, as there is a comma in the statement. However, STMJCL2 does not need to be in quotation marks since there is no comma included.
6. The OPTEVT1 keyword will generate this OPTIONS EVENT statement:

   ```
   OPTIONS EVENT(ITEMERROR,HALT)
   ```

7. The CPYCPYN keyword specifies a template name of DF81CPYC to be used in a COPYDDN clause in a COPY utility statement. The LODINDN keyword specifies a template name of DF81LDIN to be used in an INDDN clause in the LOAD utility statement.

DF8B

1. These statements define a DB2 version 8 subsystem, called `DB2 Version 8.1 - System B`, specified on the DESC keyword. The SSID is DF8B. This is the value that will be shown on the FM/DB2 Primary Option Menu.
2. The default values for PLAN and PLAN2 are not used, they have been customized to FMN2PLN8 and FMN2GEN8 respectively. Therefore this change must also be made in sample job FMN2BIND.
3. The AUDIT keyword specifies that the creation of an audit trail is optional to the user, and that FM/DB2 will ignore any value for the `Create an audit trail` setting stored in the user's ISPF profile, and reset `Create an audit trail` to OFF at the start of every FM/DB2 session.
4. All library names specified are enclosed in quotation marks and each set of library names is enclosed in parentheses, even when there is only one library name in the set. Not all library types are specified, as some may be defined in the LINKLIST.
5. The DISPLAY keyword is not specified, therefore it defaults to YES, and the subsystem ID, DF8B, is displayed in the subsystem selection list.

DG01

1. These statements define a DB2 group attachment identifier called `DB2 Group Attach ID number 1`. The SSID is DG01.
2. The default values for PLAN and PLAN2 are not used, they have been customized to FMN2PLN8 and FMN2GEN8 respectively. Therefore this change must also be made in sample job FMN2BIND.
3. The TYPE=GROUP keyword indicates that this is a group attachment ID in a DB2 data sharing environment.
4. In this definition, the AUDIT keyword specifies that **no** audit trail will be produced, regardless of the setting of `Create an audit trail` in the user's Editor options.

DG02

1. These statements define a DB2 group attachment identifier called `DB2 Group Attach ID number 2`. The SSID is DG02.
2. The default values for PLAN and PLAN2 are not used, they have been customized to FMN2PLN8 and FMN2GEN8 respectively. Therefore this change must also be made in sample job FMN2BIND.
3. In this definition, the AUDIT keyword specifies that an audit trail **will** be produced, regardless of the setting of `Create an audit trail` in the user's Editor options, and that the audit trail will be written to SMF.
4. Notice that the names of the two data sets specified for DB2LLIB overflow onto two lines, observing the standard rules for coding assembler macros.

DG02S This is the last FMN2SSDM macro definition; its' only parameter is SSID=DEFAULT, since the last (or only) FMN2SSDM macro must contain this parameter. See “SSID” on page 345.

Customizing the FM/DB2 options

FM/DB2 also provides a number of processing options that you can customize to suit your installation requirements. You specify these options in FMN2POPI. For a description of the options in FMN2POPI, and the values you can specify, see "FMN2POPI" on page 348. These options determine how your users will access the DB2 subsystems available to FM/DB2.

You then include your FMN2POPI macros in FMN2POPT, using the usermod FMN2UMDP. See "Changing the default options" on page 126 for information about other changes you can make to FMN2POPT and how to install FMN2POPT.

Parameters for FMN2POPI are optional. If you do not want to change any of the options in FMN2POPI, you do not need to provide any parameters for FMN2POPI. However, you must supply an FMN2POPI statement, without parameters if appropriate.

Examples of FMN2POPI macros

The following examples show the FMN2POPI statements you would code to change the default values for some FM/DB2 options. You include this code in your version of FMN2POPT. (Do not change FMN2POPI in FMN.SFMNMAC1.)

Example 1

```
FMN2POPI                                                +
      LIST=ACTIVE,                                      +
      CONNECT=DEFINED,                                  +
      SSIDCMD1=CONNECT,                                 +
      SSIDCMD2=SSID,                                    +
      CATOWNER=SYSIBMV,                                 +
      EDITCAPS=(INITON,PROFILE,CAPSCMD)
```

Example 2

```
FMN2POPI                                                +
      LIST=ALL,                                         +
      CONNECT=ANY,                                      +
      SSIDCMD1=SSID,                                    +
      SSIDCMD2=DB2SYS,                                  +
      CATOWNER=SYSIBM,                                  +
      EDITCAPS=(INITON,FIXED,NOCAPSCMD)
```

Note: In these examples the + character is in column 72.

Example 1

1. LIST=ACTIVE specifies that only DB2 subsystems that are active (on the local z/OS system), and any group items, on the subsystem selection list, will be shown.
2. CONNECT=DEFINED specifies that FM/DB2 will only attempt to connect to active DB2 subsystems or groups for which there is an FMN2SSDM macro entry. The DB2 subsystem or group must be active for FM/DB2 to attempt the connection.
3. SSIDCMD1=CONNECT specifies that the word CONNECT can be used as the first command to change the currently connected DB2 subsystem, when entered from FM/DB2 panels that support the command. This replaces the default value for SSIDCMD1 of SSID. CONNECT can be abbreviated to CONN.

   SSIDCMD2=SSID specifies that the word SSID can be used as the second command to change the currently connected DB2 subsystem, when entered from FM/DB2 panels that support the command. This replaces the default value for SSIDCMD2 of DB2SYS.

The net effect of these two statements is to define two commands, CONNECT and SSID. CONNECT effectively replaces the DB2SYS command. The SSID command remains available.

4. CATOWNER=SYSIBMV specifies that the catalog owner for DB2 catalog tables will be SYSIBMV, replacing the default value of SYSIBM. You must ensure that suitable DB2 views, with an owner of SYSIBMV, have been defined for all DB2 catalog tables accessed by FM/DB2.
5. EDITCAPS=(INITON,PROFILE,CAPSCMD) sets the FM/DB2 editor caps option as follows:

   For new users the CAPS edit option is set to ON. The initial setting is stored in the user's ISPF profile and can be changed using either the global or local edit option's panels. Each FM/DB2 edit session starts with the CAPS command set to the current edit option's setting. The CASE command can be used within any FM/DB2 edit session. The only difference between this, and the installation default, (INITOFF,PROFILE,CAPSCMD), is that new users begin with the CAPS edit option set to ON. With the default setting new users begin with the CAPS edit option set to off.

Example 2

1. LIST=ALL specifies that all DB2 subsystems defined with an FMN2SSDM macro entry on the subsystem selection list, whether active or not, will be shown. The exception is any DB2 subsystem defined with DISPLAY=HIDDEN in the appropriate FMN2SSDM macro. Other DB2 systems defined on the local z/OS system, but for which there is no FMN2SSDM macro entry, will also appear on the subsystem selection list.
2. CONNECT=ANY specifies that FM/DB2 will attempt to connect to any active DB2 subsystem or group that appears on the subsystem selection list, regardless of whether there is an FMN2SSDM macro entry for the DB2 subsystem or group.
3. SSIDCMD1=SSID specifies that the default value (SSID) is to be used as a command that can be used to change the currently connected DB2 subsystem, when entered from FM/DB2 panels that support the command.

   SSIDCMD2=DB2SYS specifies that the default value (DB2SYS) is to be used as a second command that can be used to change the currently connected DB2 subsystem, when entered from FM/DB2 panels that support the command.
4. CATOWNER=SYSIBM specifies that the catalog owner for DB2 catalog tables will be SYSIBM, which is the default. FM/DB2 will access the DB2 catalog tables directly.
5. EDITCAPS=(INITON,FIXED,NOCAPSCMD) sets the FM/DB2 editor caps option as follows:

   Every FM/DB2 edit session starts with the CAPS set to ON. The initial setting cannot be altered using the Edit Options. The editor CAPS command is disabled. The editor CASE command is still available. Lower case data can only be entered by first issuing the editor CASE MIXED command.

Changing the default options

Default processing options are supplied with FM/DB2 in the module FMN2POPT. You can change these options to suit your installation requirements. You also use FMN2POPT to define your DB2 subsystems, and, in a DB2 data sharing environment, your DB2 group attachment identifiers. You use the usermod FMN2UMDP to install your version of FMN2POPT.

Note: The options in FMN2POPT are also available to File Manager base function in FMN0POPT. However, any option changed in the FM/DB2 options macro only takes effect in FM/DB2.

You change the options as follows:

1. Copy the member FMN2POPT from FMN.SFMNSAM1 into your own source library.
2. Change the options in FMN2POPT in your library according to your requirements. For a description of the options in FMN2POPT, and the values you can specify, see Appendix A, "File Manager options," on page 289.
3. Include your FMN2SSDM macros, (as described in "Identifying all the DB2 systems that FM/DB2 will access and defining these in the FMN2POPT module (required)" on page 121, and "FMN2SSDM" on page 331), in the FMN2POPT member in your library.
4. Include your FMN2POPI macros (if required), (as described in "Customizing the FM/DB2 options" on page 125, and "FMN2POPI" on page 348), in the FMN2POPT member in your library.
5. Modify the FMN2UMDP member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
6. Install SMP/E usermod FMN2UMDP.

Note: You can also use the sample job FMN2POPH to assemble FMN2POPT if you do not want to use SMP/E.

Setting the default national language

If you installed the Japanese or Korean component of FM/DB2, (or you provided other locally translated messages and panels), you can change the default national language for your installation, for FM/DB2 batch processing. The language used by FM/DB2 under ISPF depends on the language setting for your ISPF session. To change the language for FM/DB2 batch processing, set the LANGUAGE option in FMN2POPT to your language. See "LANGUAGE" on page 306 for more information about the LANGUAGE option.

For other customization you can do for FM/DB2 for national languages, refer to Chapter 17, "Customizing FM/DB2 for national languages," on page 151.

Changing the JCL skeletons for batch mode

Several functions in FM/DB2 are available in batch mode. For these functions to run successfully, appropriate job control must be provided. This is done by means of the set processing options, and skeletons in FMN.SFMNSLIB. The job control skeletons for FM/DB2 are the members FMN2FTSL and FMN2FTEX in FMN.SFMNSLIB.

FMN2FTSL

FMN2FTSL assigns a STEPLIB DD concatenation which comprises the FM/DB2 load library and the DB2 load libraries for the DB2 subsystem you are using. (The DB2 load libraries are specified by DB2LLIB in the FMN2SSDM macro for the DB2 subsystem you are using, in FMN2POPT. For more information, see "FMN2SSDM" on page 331.)

FMN2FTSL assumes that you have installed FM/DB2 into the default target libraries, and that the load library is FMN.SFMNMOD1. If you have installed FM/DB2 into a different library than FMN.SFMNMOD1, or if you do not want the load library in a STEPLIB DD concatenation (for example, if FMN.SFMNMOD1 is in your LINKLIST), you must modify the FMN2FTSL skeleton accordingly.

You modify FMN2FTSL using the usermod FMN2UMDB. FMN2UMDB is distributed in FMN.SFMNSAM1. To do this:

1. Copy the member FMN2FTSL from FMN.SFMNSLIB to your own source library.
2. To change the name of the load library, change the line:

   ```
   )SET SFMNMOD1 = FMN.SFMNMOD1
   ```

   to read:

   ```
   )SET SFMNMOD1 = your.loadlib
   ```

   where *your.loadlib* is the name of the load library where you installed FM/DB2.

   To change the skeleton so that no STEPLIB DD statement is generated for the load library, change the line:

   ```
   )SET SFMNMOD1 = FMN.SFMNMOD1
   ```

   to read:

   ```
   )SET SFMNMOD1 = &Z
   ```

   Note: A STEPLIB DD statement will always be generated for the DB2 load libraries for the DB2 subsystem you are using.
3. Modify the FMN2UMDB member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMN2UMDB.

Note: FM/DB2 does not provide support for the automatic generation of job routing control statements in the JCL generated by FMN2FTSL.

FMN2FTEX

FMN2FTEX provides printer information to FMN2FTSL. You might also need to modify FMN2FTEX, for example, to change the DCB information for SYSPRINT.

You modify FMN2FTEX using the usermod FMN2UMDE. FMN2UMDE is distributed in FMN.SFMNSLIB. To do this:

1. Copy the member FMN2FTEX from FMN.SFMNSLIB to your own source library.
2. Modify one or more of the following three statements in FMN2FTEX in your library, as required. Do not change any other statements in FMN2FTEX.

   ```
   //SYSPRINT DD SYSOUT=*
   //FMNTSPRT DD SYSOUT=*
   //SYSTERM  DD SYSOUT=*
   ```

   For example, you might want to change

   ```
   //SYSPRINT DD SYSOUT=*
   ```

   to

   ```
   //SYSPRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
   ```

3. Modify the FMN2UMDE member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMN2UMDE.

Customizing to protect update functions in FM/DB2

You can use an external security product to write facility class rules to protect update functions within FM/DB2.

The following FM/DB2 functions are considered to be update functions:

Table 23.

Function	Menu option	Description
D2E	2	DB2 edit
DBC	3.3	Copy utility
D2I	3.6	Import utility
D2G	3.8	DB2 data create
DBSBSP	4.1	Basic select prototyping
DBSASP	4.2	Advanced select prototyping
DBSEDX	4.4	DB2 edit and execute SQL
DBSENX	4.3	DB2 enter and execute SQL

If you require protection of update functions, specify SEC=YES in the FMN2POPT module (see the SEC parameter in Appendix A, "File Manager options," on page 289).

The facility class rules that are required are:
FILEM.DB2.UPDATE
FILEM.FUNCTION.*function_code*

Example 1

To protect all DB2 update functions:
Specify SEC=YES in the FMN2POPT
Write a facility class rule for FILEM.DB2.UPDATE

Example 2

To protect the DB2 editor function only:
Specify SEC=YES in the FMN2POPT
Write a facility class rule for FILEM.FUNCTION.D2E

The FILEM.FUNCTION facility class rule is checked first. If access is denied via this rule then no further checking is done.

If there is no FILEM.FUNCTION.*function_code* rule, the more general facility class rule FILEM.DB2.UPDATE is checked.

Chapter 15. Customizing the FM/DB2 audit facility (FMN2POPT controlled auditing)

You can use the audit logging facilities of FM/DB2 to create an audit trail of activity on DB2 objects by FM/DB2 users. You can create your audit trail either using SMF, or recording the activity to an audit log data set.

You have two choices for controlling FM/DB2 auditing. These are:
1. Auditing is controlled using options specified in the FMN2POPT module. This method was used in previous versions of FM/DB2. To use this method follow the directions in the remainder of this chapter.
2. Auditing is controlled using SAF FACILITY and XFACILIT rules, defined using an external security product such as RACF (or equivalent product). To use this method follow the directions in Chapter 16, "Customizing the FM/DB2 audit facility (SAF-rule controlled auditing)," on page 135.

When you select FMN2POPT-controlled auditing, ensure that the FMN2PARM member (in SYS1.PARMLIB) specifies:

```
FMAUDIT  SAF_CTRL=NO
```

See Appendix D, "File Manager options specified in PARMLIB members," on page 411 for more information.

FM/DB2 determines whether or not an audit trail is produced, and whether it will be written to an audit log data set or to SMF, by the settings of the AUDIT option in the FMN2SSDM macro, and `Create an audit trail` in the global Editor Options (option 0.3 from the FM/DB2 Primary Option Menu). For more information on changing the editor global options, the information collected by an audit trail, and how to report on the information, see the *File Manager User's Guide and Reference for DB2 Data*. See "FMN2SSDM" on page 331 for a description of the FMN2SSDM macro.

Note: You do not use the AUDITLOG option in FMN0POPT to determine whether or not to produce an audit trail in FM/DB2. The AUDITLOG option has no effect in FM/DB2.

Determining if an audit trail is to be produced

You can customize FM/DB2 for recording an audit trail (using the AUDIT option in FMN2SSDM) in one of four ways:

- No audit trail produced under any circumstances. Specify AUDIT=NONE.
- Optional for the current FM/DB2 session; an audit trail is produced if the individual user wants to record audit information, however the audit option is always reset to "not selected" whenever an FM/DB2 session ends. Specify AUDIT=(OPTIONAL,OFF).
- Optional for this and subsequent FM/DB2 sessions; an audit trail is produced if the individual user wants to record audit information. The current setting of the audit option is saved in the ISPF profile between FM/DB2 sessions. Specify AUDIT=(OPTIONAL,PROFILE).
- Required; an audit trail will be produced for each user, regardless of whether or not the user wants to record audit information. Specify AUDIT=(REQUIRED,....).

© Copyright IBM Corp. 2000, 2011

- Demand; an audit trail will be produced for each user, regardless of whether or not the user wants to record audit information and an audit report job will be submitted at conclusion of the edit function or when you change SSIDs. Specify AUDIT=(DEMAND,....). The job submitted will be determined by skeleton member FMN2FTAD found in FMN.SFMNSLIB. You should customize the job card and JCL to specify the reporting options you require. For changing the audit report options in the skeleton see "AUD (Print Audit Trail Report)" in the *File Manager User's Guide and Reference*.

In addition, if an audit trail is to be produced, you can select whether to record to SMF or user data sets.

When recording of audit information is optional to the user, (see above), this is determined by the individual setting of `Create an audit trail` in the Editor global options.

For optional auditing, AUDIT=(OPTIONAL,OFF) is suggested as the best option when audit data is written to data sets, rather than SMF, and when auditing is an exceptional, rather than a normal process.

Note: The AUDIT option is active only when data is modified via the editor (view, browse or edit) or copy utility. The setting of this option will be ignored for any other File Manager function that allows data to be modified, and for data modified by the ISPF editor, even though this is from within File Manager.

Auditing can be selectively disabled when using FM/DB2 view or browse by specifying AUDITBROWSE=N in the FMN2SSDM macro definition for a DB2 system. Specifying this option does not affect the audit records produced when using FM/DB2 edit. See "AUDITBROWSE" on page 333 for information about AUDITBROWSE.

Using System Management Facilities (SMF) for audit logging

If you intend to use SMF for audit logging, you must also do the following:

- Ensure that the load module FMNSMF is APF-authorized. For more information about using SMF for audit logging, and making FMNSMF APF-authorized, see page 49.
- Add the load module FMNSMF to the AUTHTSF list in member IKJTSO*xx* in SYS1.PARMLIB. If you do not do this, even if you have selected to record to SMF, and you have specified an SMF record type, no recording will be done.
- Select an SMF record type for the audit log records, and include it in your SMF parmlib member SMFPRM*xx*.
- Specify this SMF record type in the options macro, FMN2POPT. For information on how to do this, see "Changing the default options" on page 126.

To activate any changes you have made to SYS1.PARMLIB members, either restart your system, or use the appropriate commands for your site to dynamically activate the changes.

To report on the audit trail information collected by SMF, you must extract this information from SMF to your own data set. The information in this data set can then be printed by the FM/DB2 Print Audit Trail utility (option 3.10 from the FM/DB2 Primary Option menu).

A sample job, FMNSMFX, is provided in FMN.SFMNSAM1, to help you extract the SMF data to your own data set. See the comments in the job for information on changes you might need to make. For more information about SMF, see *z/OS MVS System Management Facilities (SMF)*.

Recording an audit trail to a data set

If you choose to record your audit trail to a data set, you can print this information using the FM/DB2 Print Audit Trail utility. To do this select option 3.10 from the FM/DB2 Primary Option menu.

The format of the audit log data set name is determined by the setting of the AUDITHLQ parameter in the FMN0POPI definition in FMN2POPT. See "AUDITHLQ" on page 291 for more information about the AUDITHLQ option.

The following data set name formats may be generated:

```
userid.FMN2AUD.<ssid>.Dyymmdd.Thhmmss              (when AUDITHLQ= (blank))
audithlq.FMN2AUD.<ssid>.Dyymmdd.Thhmmss            (when AUDITHLQ=audithlq)
qual1.<qual2.><qual3.>Dyymmdd.Thhmmss              (when AUDITHLQ=qual1.<qual2.><qual3>)
```

where:

audithlq	Is any 1-8 character constant that is valid in the context of a data set name
ssid	Is the name of the currently connected (local) DB2 system
userid	Is the userid creating the data set
Dyymmdd	Is the date of the activity
Thhmmss	Is the time of the activity

When AUDITHLQ contains one or more periods, the AUDITHLQ value is treated as a data set prefix, with one, two or three levels. Each level of the prefix can be:

XXX	Any 1-8 character constant that is valid in the context of a data set name.
&&PREFIX	Indicates that the user's TSO prefix should by used. This will be null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix will also be null.
&&USER	Indicates that the user's logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.
&&UID	Indicates that the user's TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user's TSO logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.
&&FUNCOD	Indicates that the File Manager internal function code should be used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.
&&SSID	Indicates that the currently connected (local) DB2 subsystem name should be used.

This data set can be printed using the FM/DB2 `Print Audit Trail` utility, (option 3.10 from the FM/DB2 Primary Option menu).

In this case, audit data sets are produced for each user, and are subject to the usual rules controlling user data sets. For this reason, if you intend that audit logging should always take place and be reported on, it is suggested that you select AUDIT=(REQUIRED,SMF) in your FMN2SSDM macro.

Chapter 16. Customizing the FM/DB2 audit facility (SAF-rule controlled auditing)

Controlling File Manager auditing using SAF (FM/DB2 component only)

You can use SAF to control the following aspects of FM/DB2 auditing:

- Whether FM/DB2 writes audit records to SMF, the user's audit log data set, or both.
- Whether FM/DB2 writes audit records when accessing DB2 objects or issuing DB2 commands.
- Whether FM/DB2 writes audit records only for update-related activities, or for all (read) activities.
- Whether FM/DB2 automatically produces the audit log report at the end of a FM/DB2 session ("Demand" audit logging).

You specify whether or not FM/DB2 is to use SAF to control auditing by the FMAUDIT parameter in the FMN2PARM member in SYS1.PARMLIB (or other library in the logical parmlib concatenation). See Appendix D, “File Manager options specified in PARMLIB members,” on page 411 for more information.

For users wanting to control auditing by means of parmlib, the users being audited must have read access or more to the SAF facility class profile FILEM.PARMLIB.DB2. In this case, auditing is controlled by the FMAUDIT parameter in the FMN2PARM member. See “FM/DB2 options specified in FMN2PARM” on page 422 for more information.

Note: File Manager for DB2 will not start if a user has read access to this facility class and the FMN2PARM member is unavailable from parmlib.

Example: User PROD1 to have auditing controlled from parmlib.

Write this RACF rule:

```
RDEF FACILITY FILEM.PARMLIB.DB2 AUDIT(NONE) UACC(NONE)
        OWNER(ownerid)
PE FILEM.PARMLIB.DB2 ACC(READ) ID(PROD1)  CLASS(FACILITY)
```

If SAF processing is not active, or the rule is not defined, or the rule is defined and the user has no access, then no parmlib processing is performed.

If there is no parmlib processing to specify SAF-controlled auditing, then SAF-controlled auditing is performed for users that have read access to the SAF facility class profile FILEM.SAFAUDIT.DB2.

Example: User PROD2 to have SAF auditing without parmlib.

Write this RACF rule:

```
RDEF FACILITY FILEM.SAFAUDIT.DB2 AUDIT(NONE) UACC(NONE)
        OWNER(ownerid)
PE FILEM.SAFAUDIT.DB2 ACC(READ) ID(PROD2)  CLASS(FACILITY)
```

Note: If SAF-controlled auditing is specified without parmlib processing, then the SMF record type is obtained from the options module SMFNO value.

© Copyright IBM Corp. 2000, 2011

When a security product other than RACF is in use

File Manager V11 APAR PM31842 introduced a change to remove the mandatory requirement for definition of members in the logical PARMLIB concatenation. In order to achieve this, File Manager makes RACROUTE calls, with STATUS=ACCESS, to SAF FACILITY profiles:

FILEM.PARMLIB.**
FILEM.SAFAUDIT.** (only if the first call fails)

STATUS=ACCESS is specified because the RACROUTE call should not audit the call. The documentation in the *z/OS V1R11 Security Server RACROUTE Macro Reference*, in the description of RACROUTE REQUEST=AUTH,STATUS=ACCESS, states:

ACCESS

The request is simply to return the user's highest current access to the resource specified. Upon successful completion, the user's access is returned in the RACF reason code. **No auditing is done for this request.**

When RACF is used, the STATUS=ACCESS request works as documented, and no security-related logging or abends are generated, even if you do not have access to the profile.

However, when non-RACF security products (such as ACF2) are used, S047 ABEND047 may be issued in response to the above RACROUTE request.

The ACF2 global system options (GSO) definitions shown here have been used to avoid the S047 abends:

```
SET CONTROL(GSO)
INSERT SAFDEF.APFFM PROGRAM(FMN*****) RB(FMN*****) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=FACILITY,STATUS=ACCESS)
INSERT SAFDEF.APFFM2 PROGRAM(FILEM***) RB(FILEM***) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=FACILITY,STATUS=ACCESS)
F ACF2,REFRESH(SAFDEF)
INSERT SAFDEF.APFFM PROGRAM(FMN*****) RB(FMN*****) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
INSERT SAFDEF.APFFM2 PROGRAM(FILEM***) RB(FILEM***) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
F ACF2,REFRESH(SAFDEF)
```

Implementing SAF-rule controlled auditing

Use the following checklist to implement SAF-rule controlled auditing:

__ 1. Determine the SAF FACILITY and XFACILIT rules that will be required. See "Understanding how FM/DB2 uses SAF rules to control auditing" on page 137 for more information.

__ 2. Write the relevant SAF rules. See the examples in "SAF rule examples" on page 142.

__ 3. Modify the FMN2PARM member to activate SAF-rule controlled auditing. See "Testing SAF-rule controlled auditing" on page 145 for more information.

__ 4. Test the configuration to ensure that auditing is occurring as required. You can use the facilities for customizing the FMN2PARM definitions (see Appendix D, "File Manager options specified in PARMLIB members," on page 411) to, for example, turn on SAF-rule controlled auditing for a specific logonid (or logonids), without changing the behaviour for other users.

__ 5. When testing is complete, activate SAF-rule controlled auditing for all FM/DB2 users by changing the FMN2PARM member.

Understanding how FM/DB2 uses SAF rules to control auditing

SAF (System Authorization Facility) allows applications, such as File Manager, to define "resources" that might need to be protected. The "resource" to be protected need not be something specific, such as a data set; it can be essentially any type of resource or facility that the application considers to be important. For FM/DB2 and auditing, the "resource" is the ability to write audit log records. The resource names reflect either the type of auditing that is to occur (eg to SMF), or the type of DB2 object, SQL statement, or DB2 command that is being processed, for example, a DB2 object name.

FM/DB2 uses two types of SAF resource names to control auditing. Note that a user's ability to write audit log records under SAF control is independent of, for example, the user's ability to access a particular DB2 object, issue a particular DB2 command and so on. A user may be able to write audit records when using the FM/DB2 editor to look at a particular DB2 object, but lack the DB2 authority to actually look at the object.

The SAF resource rules used by FM/DB2 to control auditing are shown in Table 28 on page 146 and Table 29 on page 146).

Understanding SAF rule access levels

SAF provides for five levels of access to any FACILITY or XFACILIT resource. The levels of access form a hierarchy, so that a user with the highest level of access to a resource also has access to all the lower levels. The levels of access are specified in RACF rules using the following mnemonics:

NONE No access
READ Level 1 access
UPDATE Level 2 access
CONTROL Level 3 access
ALTER Level 4 access.

It is important to understand that the mnemonics used (READ, ALTER and so on) can and do mean different things, depending on the context in which the SAF resource name is used. This can be confusing since READ and UPDATE have obvious meanings when it comes to, for example, accessing a data set. For SAF rules used to control FM/DB2 audit, it may aid understanding to think of the mnemonics as indicating level 1 access, level 2 access and so on.

For the SAF resource rules used by FM/DB2, the meanings of the various levels of access are:

NONE
The user does not have access to the resource; this typically means the user cannot write audit log records.

READ The user has level 1 access to the resource; this typically means that the user can write audit log records.

UPDATE
The user has level 2 access to the resource. This level of access only has meaning for FACILITY rule 2 (see Table 28 on page 146). A user with level 2 access can write audit log records to the user's audit log data set, and the audit log data set will be printed at the end of the user's session (on-line execution only). This is equivalent to the DEMAND audit option in the non-SAF case.

CONTROL
The user has level 3 access to the resource. This level of access only has

meaning with the XFACILIT rules described in Table 29 on page 146. A user with level 3 access can change the auditing requirement specified in the current resource name. Note that, in order to do so, the user must also have at least level 1 access to FACILITY rule 3 (see Table 28 on page 146).

ALTER

The user has level 4 access to the resource. This level of access is not used by FM/DB2.

How FM/DB2 determines whether audit log records should be written

The determination of whether audit records are to be written for a particular FM/DB2 function and a given TSO logonid follows this three step process:

Step 1. What is the FMAUDIT specification in the FMN2PARM member?

The FMAUDIT specification setting in the FMN2PARM member (in SYS1.PARMLIB) is the "master" switch for SAF-rule controlled auditing. Note that there are facilities available to specify different settings in the FMN2PARM member for different TSO logonids, see Appendix D, “File Manager options specified in PARMLIB members,” on page 411 for more information. For any given TSO logonid, there are two possibilities:

SAF_CTRL=NO

SAF-rule controlled auditing is not in effect. Auditing is determined by the settings in the FMN2POPT module, see Chapter 15, “Customizing the FM/DB2 audit facility (FMN2POPT controlled auditing),” on page 131.

SAF_CTRL=YES

SAF-rule controlled auditing is in effect. Processing continues to step 2.

Step 2. Does the user have access to write audit records?

This is determined by the user's access to rules 1 and 2 in Table 28 on page 146, the various outcomes are summarized in Table 24.

Table 24. Determination of a user's ability to write audit log records

TODSN access[1]	TOSMF access[2]	OPTION access[3]	Can write audit records?	Demand logging?	"Create audit trail" option[4]
NONE	NONE	ANY	No	No	Not visible
READ	NONE	NONE	Yes, data set only	No	Not visible
READ	NONE	READ	Yes, data set only	No	Visible
UPDATE	NONE	NONE	Yes, data set only	Yes	Not visible
UPDATE	NONE	READ	Yes, data set only	Yes	Visible
NONE	READ	NONE	Yes, SMF only	No	Not visible
NONE	READ	READ	Yes, SMF only	No	Visible
READ	READ	NONE	Yes, to data set and SMF	No	Not visible
READ	READ	READ	Yes, to data set and SMF	No	Visible
UPDATE	READ	NONE	Yes, to data set and SMF	Yes	Not visible

Table 24. Determination of a user's ability to write audit log records (continued)

TODSN access[1]	TOSMF access[2]	OPTION access[3]	Can write audit records?	Demand logging?	"Create audit trail" option[4]
UPDATE	READ	READ	Yes, to data set and SMF	Yes	Visible

Notes:

1. Refers to the level of access the user has to SAF FACILITY rule 1 in Table 28 on page 146.
2. Refers to the level of access the user has to SAF FACILITY rule 2 in Table 28 on page 146.
3. Refers to the level of access the user has to SAF FACILITY rule 3 in Table 28 on page 146.
4. The visibility of the "Create audit trail" option does not influence whether a user can write audit log records, although the user must have access to write audit log records (to either a data set or SMF), for the option to be visible.

If the user does not have the ability to write audit log records, then no check of SAF resource names in Step 3 occurs.

A user's access to write audit log records at Step 2 only indicates that auditing might occur, the final decision depends on the user's level of access to the XFACILIT resource name (or names) that apply to the particular FM/DB2 function.

Step 3. Does the user have access to write audit records for the current function?

The XFACILIT resource name(s) used by FM/DB2 to determine whether audit records should be written depend on the FM/DB2 function being executed.

The types of SQL statements and DB2 commands that might be issued by each FM/DB2 function are shown in Table 25.

The relationship between various SQL statements or DB2 commands and XFACILIT resource names is shown in Table 26 on page 140.

Table 25. Types of SQL (and DB2) statements issued by FM/DB2 functions

FM/DB2 function	Option number	SQL/DB2 statements
Browse	B	SELECT
View	1	SELECT
Edit	2	SELECT, DELETE, INSERT, UPDATE
Print	3.1	SELECT
DB2 Objects	3.2	CREATE, DROP
Copy	3.3	SELECT (source object) DELETE (target object) INSERT (target object) UPDATE (target object)
Object List	3.4	DROP, GRANT, REVOKE, FREE, BIND, REBIND
Object Privileges	3.5	GRANT, REVOKE
Import	3.6	DELETE, INSERT, UPDATE
Export	3.7	SELECT
Create	3.8	INSERT

Table 25. Types of SQL (and DB2) statements issued by FM/DB2 functions (continued)

FM/DB2 function	Option number	SQL/DB2 statements
Basic SELECT prototyping	4.1	SELECT (any editor mode), DELETE, INSERT and UPDATE (only when the Editor option for "Arbitrary SQL Select Statements" is set to edit).
Advanced SELECT prototyping	4.2	SELECT (any editor mode), DELETE, INSERT and UPDATE (only when the Editor option for "Arbitrary SQL Select Statements" is set to edit).
Enter, Execute and Explain SQL	4.3	For SELECT statements as per "Basic SELECT prototyping". Any other SQL statement that can be issued.
Edit/Execute SQL (Data Set)	4.4	For SELECT statements as per "Basic SELECT prototyping". Any other SQL statement that can be issued.
DB2 commands	6	-ACCESS -ALTER -ARCHIVE -CANCEL -DISPLAY -MODIFY -RECOVER -REFRESH -RESET -SET -START -STOP -TERM

Table 26. Relationship between SQL statement type and SAF resource names

SQL Statement Type	Audit resource name suffix (see note 1)
ALTER	DDL.<object_type>[7]
COMMENT	OTHER.ADHOCSQL
COMMIT	OTHER.ADHOCSQL
CREATE	DDL.<object_type>[7]
DELETE	UPDATE.ADHOCSQL
DROP	DDL.<object_type>[7]
EXCHANGE	OTHER.ADHOCSQL
EXPLAIN	OTHER.ADHOCSQL
GRANT	AUTH.<auth_type>[8]
INSERT	UPDATE.ADHOCSQL
LABEL	OTHER.ADHOCSQL
LOCK	OTHER.ADHOCSQL
MERGE	OTHER.ADHOCSQL
REFRESH	OTHER.ADHOCSQL
RENAME	DDL.<object_type>[7]
REVOKE	AUTH.<auth_type>[8]

Table 26. Relationship between SQL statement type and SAF resource names (continued)

SQL Statement Type	Audit resource name suffix (see note 1)
ROLLBACK	OTHER.ADHOCSQL
SELECT	READ.<object>[2] UPDATE.<object>[3] READ.ADHOCSQL[4,6] UPDATE.ADHOCSQL[5,6]
SET	OTHER.ADHOCSQL
TRUNCATE	UPDATE.ADHOCSQL
UPDATE	UPDATE.ADHOCSQL

Notes:

1. The prefix for all resource names in this table is FILEM.AUDIT.<ssid>, where ssid is the DB2 subsystem or group ID.
2. This resource name is used when processing a DB2 object via options B and 1.
3. This resource name is used when processing a DB2 object using option 2.
4. This resource name is used when processing a DB2 object (or objects) using a SELECT statement entered via options 4.1, 4.2, 4.3 or 4.4, when FM/DB2 browse or view is used to display the result table for the SELECT.
5. This resource name is used when processing a DB2 object (or objects) using a SELECT statement entered via options 4.1, 4.2, 4.3 or 4.4, when FM/DB2 edit is used to display the result table for the SELECT.
6. The choice of FM/DB2 editor mode (browse, view or edit) is determined by the setting of the "Arbitrary SQL Select Statements", Editor option, which can be found on the second FM/DB2 system options panel. This is accessed by typing 0.0.2 from the FM/DB2 main menu.
7. See Table 30 on page 147.
8. See Table 31 on page 148.

For some FM/DB2 functions only a single SAF XFACILIT rule needs to be checked to determine whether audit log records should be written. An example is the FM/DB2 editor, which processes a single DB2 object in either READ or UPDATE modes.

For other FM/DB2 functions multiple SAF XFACILIT rules may be checked.

Example 1: FM/DB2 Copy utility

- The access to write audit records for READ access to the source DB2 object is checked.
- The access to write audit records for UPDATE access to the target DB2 object is checked.

Example 2: FM/DB2 Edit/Execute SQL (Data Set) utility

This utility allows the execution of SQL statements coded in a data set. FM/DB2 checks the access to write audit records for each SQL statement in the data set as it is executed and audit records are written (or not) as appropriate.

Controlling where FM/DB2 writes audit log records

You can use SAF to control whether FM/DB2 writes audit log records to SMF, the user's audit log data set, or to both.

How FM/DB2 determines whether audit log records should be written

The following table shows the SAF FACILITY class resource names used to control FM/DB2 logging to SMF and the user's audit log data set.

Table 27. SAF FACILITY class resource names controlling disposition of FM/DB2 audit records

FACILITY Class Name	Purpose
FILEM.AUDIT2.<ssid>.TOSMF	Enables or disables auditing to SMF for FM/DB2.
FILEM.AUDIT2.<ssid>.TODSN	Enables or disables auditing to the user's data set for FM/DB2.

See the examples in "Controlling where FM/DB2 writes audit log records."

SAF rule examples

This section shows SAF rule examples under different conditions.

Controlling where FM/DB2 writes audit log records

You can use SAF to control whether FM/DB2 writes audit log records to SMF, the user's audit log data set, or to both.

Table 28 on page 146 shows the SAF FACILITY class resource names used to control FM/DB2 to logging and the user's audit log data set.

Example 1
- Disable audit logging to a user data set for all FM/DB2 users, for DB2 system DSNA.
- Enable FM/DB2 audit logging to SMF for the PROD logonid, for DB2 system DSNA.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT2.DSNA.TOSMF[1]
RDEL FACILITY FILEM.AUDIT2.DSNA.TODSN[1]

RDEF FACILITY FILEM.AUDIT2.DSNA.TOSMF UACC(NONE) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT2.DSNA.TODSN UACC(NONE) OWNER(XXXXXXX)[3]

PE FILEM.AUDIT2.DSNA.TOSMF ACC(READ) ID(PROD) CLASS(FACILITY)[4]
```

Explanation:
1. Delete any existing facility rule
2. Define the facility rule for DB2 system DSNA and audit logging to SMF (TOSMF suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
3. Define the facility rule for DB2 system DSNA and audit logging to the user's audit log data set (TODSN suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
4. Allow logonid PROD to write audit log records (ACC(READ)), to SMF.

Example 2
- Enable audit logging to a user data set for all FM/DB2 users, for DB2 system DSNB.
- Enable demand logging for the following users, PROD1, PROD2, PROD3
- Disable audit logging to SMF for all FM/DB2 users, for DB2 system DSNB.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT2.DSNB.TOSMF[1]
RDEL FACILITY FILEM.AUDIT2.DSNB.TODSN[1]

RDEF FACILITY FILEM.AUDIT2.DSNB.TOSMF UACC(NONE) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT2.DSNB.TODSN UACC(READ) OWNER(XXXXXXX)[3]

PE FILEM.AUDIT2.DSNB.TODSN ACC(UPDATE) ID(PROD1) CLASS(FACILITY)[4]
PE FILEM.AUDIT2.DSNB.TODSN ACC(UPDATE) ID(PROD2) CLASS(FACILITY)[4]
PE FILEM.AUDIT2.DSNB.TODSN ACC(UPDATE) ID(PROD3) CLASS(FACILITY)[4]
```

Explanation:

1. Delete any existing facility rule
2. Define the facility rule for DB2 system DSNB and audit logging to SMF (TOSMF suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
3. Define the facility rule for DB2 system DSNB and audit logging to the user's audit log data set (TODSN suffix). UACC(READ) is used so that any user, for which there is no specific rule, has read access, and can therefore write audit log records.
4. Allow logonids PROD1, PROD2, PROD3 to write audit log records with automatic printing of the audit report ("Demand logging") (ACC(UPDATE)), to SMF.

Example 3

- Disable audit logging completely for all FM/DB2 users, for DB2 system DSND.
- Enable dual logging for all FM/DB2 users for DB2 system DSNP.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT2.DSND.TOSMF[1]
RDEL FACILITY FILEM.AUDIT2.DSND.TODSN[1]
RDEL FACILITY FILEM.AUDIT2.DSNP.TOSMF[1]
RDEL FACILITY FILEM.AUDIT2.DSNP.TODSN[1]

RDEF FACILITY FILEM.AUDIT2.DSND.TOSMF UACC(NONE) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT2.DSND.TODSN UACC(NONE) OWNER(XXXXXXX)[3]
RDEF FACILITY FILEM.AUDIT2.DSNP.TOSMF UACC(READ) OWNER(XXXXXXX)[4]
RDEF FACILITY FILEM.AUDIT2.DSNP.TODSN UACC(READ) OWNER(XXXXXXX)[5]
```

Explanation:

1. Delete any existing facility rule
2. Define the facility rule for DB2 system DSND and audit logging to SMF (TOSMF suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
3. Define the facility rule for DB2 system DSND and audit logging to the user's audit log data set (TODSN suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
4. Define the facility rule for DB2 system DSNP and audit logging to SMF (TOSMF suffix). UACC(READ) is used so that any user, for which there is no specific rule, has access (and can therefore write audit records to SMF).
5. Define the facility rule for DB2 system DSNP and audit logging to to the user's audit log data set (TODSN suffix). UACC(READ) is used so that any user, for which there is no specific rule, has access (and can therefore write audit records to the user's audit log data set).

Controlling auditing of update access to DB2 objects

You can use SAF to control whether FM/DB2 writes audit log records for FM/DB2 functions that update, or have the potential to update, DB2 objects. Examples of such functions are:

- The FM/DB2 editor operating in edit mode (but not view or browse)

- FM/DB2 Copy, for the target DB2 object
- FM/DB2 Import, for the target DB2 object
- FM/DB2 Data Create

Table 29 on page 146 shows the SAF XFACILITY class resource names used to control FM/DB2 audit logging.

Example 1

- Enable audit logging for update access to DSN8910.EMP in DB2 system DSNA, for all users except TSO logonid MAINT1.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT.DSNA.UPDATE.OBJ.DSN8910.EMP[1]

RDEF XFACILIT FILEM.AUDIT.DSNA.UPDATE.OBJ.DSN8910.EMP +
     OWNER(XXXXXXXX) UACC(READ)[2]

PE FILEM.AUDIT.DSNA.UPDATE.OBJ.DSN8910.EMP +
     CLASS(XFACILIT) ID(MAINT1) ACC(NONE)[3]
```

Explanation:

1. Delete any existing XFACILIT rule
2. Define the XFACILIT rule for DB2 system DSNA and UPDATE access to DB2 object (OBJ) DSN8910.EMP. UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).
3. A specific rule for logonid MAINT1 to prevent audit log records being written.

Example 2

- Enable audit logging for update access to remote object DSN8910.EMP. This object is accessed from DB2 system DSNA. The location of the remote DB2 system is TEXAS. Audit logging is to be performed for all users except SERVIC1, SERVIC2.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT.DSNA.UPDATE.REMOBJ.TEXAS.DSN8910.EMP[1]

RDEF XFACILIT FILEM.AUDIT.DSNA.UPDATE.REMOBJ.TEXAS.DSN8910.EMP +
     OWNER(XXXXXXXX) UACC(READ)[2]

PE FILEM.AUDIT.DSNA.UPDATE.REMOBJ.TEXAS.DSN8910.EMP +
     CLASS(XFACILIT) ID(SERVIC1) ACC(NONE)[3]
PE FILEM.AUDIT.DSNA.UPDATE.REMOBJ.TEXAS.DSN8910.EMP +
     CLASS(XFACILIT) ID(SERVIC2) ACC(NONE)[3]
```

Explanation:

1. Delete any existing XFACILIT rule
2. Define the XFACILIT rule for DB2 system DSNA and UPDATE access to remote DB2 object (REMOBJ) TEXAS.DSN8910.EMP. UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).
3. Specific rules for logonids SERVIC1, SERVIC2 to prevent audit log records being written.

Controlling auditing of read access to DB2 objects

You can use SAF to control whether FM/DB2 writes audit log records for FM/DB2 functions that read data from DB2 objects. Examples of such functions are:

- The FM/DB2 editor operating in view or browse modes (but not edit mode)
- FM/DB2 print.

- FM/DB2 Copy, for the source DB2 object
- FM/DB2 Export

Table 29 on page 146 shows the SAF XFACILITY class resource names used to control FM/DB2 audit logging.

Example 1

- Enable audit logging for read access to DSN8910.DEPT in DB2 system DSNC, for all users except TSO logonid MASTER1.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT.DSNC.READ.OBJ.DSN8910.DEPT[1]

RDEF XFACILIT FILEM.AUDIT.DSNC.READ.OBJ.DSN8910.DEPT +
     OWNER(XXXXXXXX) UACC(READ)[2]

PE FILEM.AUDIT.DSNC.READ.OBJ.DSN8910.DEPT +
     CLASS(XFACILIT) ID(MASTER1) ACC(NONE)[3]
```

Explanation:

1. Delete any existing XFACILIT rule
2. Define the XFACILIT rule for DB2 system DSNC and READ access to DB2 object (OBJ) DSN8910.EMP. UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).
3. A specific rule for logonid MASTER1 to prevent audit log records being written.

Example 2

- Enable audit logging for update access to remote object DSN8910.ACT. This object is accessed from DB2 system DSNP. The location of the remote DB2 system is MONTANA. Audit logging is to be performed for all users except DEV1.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT.DSNP.READ.REMOBJ.MONTANA.DSN8910.ACT[1]

RDEF XFACILIT FILEM.AUDIT.DSNP.READ.REMOBJ.MONTANA.DSN8910.ACT +
     OWNER(XXXXXXXX) UACC(READ)[2]

PE FILEM.AUDIT.DSNP.READ.REMOBJ.MONTANA.DSN8910.ACT +
     CLASS(XFACILIT) ID(DEV1) ACC(NONE)[3]
```

Explanation:

1. Delete any existing XFACILIT rule
2. Define the XFACILIT rule for DB2 system DSNP and READ access to remote DB2 object (REMOBJ) MONTANA.DSN8910.ACR. UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).
3. Specific rule for logonid DEV1 to prevent audit log records being written.

Testing SAF-rule controlled auditing

You can test your SAF rules achieve the desired result using the following method:

1. Select a "test" TSO logonid, for example "TEST1".
2. Write the appropriate SAF FACILITY and XFACILIT rules for this logonid.
3. Modify the FMN2PARM member to specify:

```
FMAUDIT SAF_CTRL=NO
<U TEST1>
FMAUDIT SAF_CTRL=YES
</U>
```

4. Logon using TEST1.

   From the FM/DB2 main menu select the Help pull-down menu, option 8 (About DB2). This should show:

   ```
   Auditing  . . . . . : SAF-RULE CONTROLLED
   ```

   If it does not then TEST1 does not have access to the TOSMF or TODSN FACILITY rules.
5. Test individual FM/DB2 functions to confirm that audit log records are written (or not) as specified in the appropriate SAF rules.
6. When testing is complete and auditing is to be applied to all users, change the FMN2PARM member to:

   ```
   FMAUDIT  SAF_CTRL=YES.
   ```

FM/DB2 auditing FACILITY and XFACILIT class resource names

These two tables (and associated tables) list FACITITY and XFACILIT class resource names and details.

Table 28. FM/DB2 auditing FACILITY class resource names

Rule Number	Resource Name[1]	Purpose
1	<pfx>.TODSN	Allows a user to write audit log records to the user's audit log data set.
2	<pfx>.TOSMF	Allows a user to write audit log records to SMF.
3	<pfx>.OPTION	Allows the user access to the "Create audit trail" option on selected FM/DB2 panels.

Notes:

1. The prefix <pfx> for all resource names in this table is FILEM.AUDIT2.<ssid>, where ssid is the DB2 subsystem or group ID.

Table 29. FM/DB2 auditing XFACILIT class resource names

Rule Number	Resource Name suffix[1]	Purpose
1	<pfx>.READ.OBJ.<object>	Allows a user to write audit log records for functions that read data from the specified local object (object) in the specified DB2 system (ssid).
2	<pfx>.UPDATE.OBJ.<object>	Allows a user to write audit log records for functions that change data from the specified local object (object) in the specified DB2 system (ssid).
3	<pfx>.READ.REMOBJ.<object>	Allows a user to write audit log records for functions that read data from the specified remote object (object), when accessed from the specified DB2 system (ssid).

Table 29. FM/DB2 auditing XFACILIT class resource names (continued)

Rule Number	Resource Name suffix[1]	Purpose
4	<pfx>.UPDATE.REMOBJ.<object>	Allows a user to write audit log records for functions that update data from the specified remote object (object), when accessed from the specified DB2 system (ssid).
5	<pfx>.READ.ADHOCSQL	Allows a user to write audit log records for functions that read data from some result table in the specified DB2 system (ssid).
6	<pfx>.UPDATE.ADHOCSQL	Allows a user to write audit log records for functions that update data for some result table in the specified DB2 system (ssid). Alternatively, allows a user to write audit log records for SQL statements that might update data in the specified DB2 system (ssid).
7	<pfx>.OTHER.ADHOCSQL	Allows a user to write audit log records for functions that issue SQL statements that are not covered by the READ or UPDATE ADHOCSQL rules, or the DDL and AUTH rules, in the specified DB2 system (ssid).
8	<pfx>.DDL.<objecttype>[2]	Allows a user to write audit log records for functions that issue DDL statements (such as CREATE, DROP, ALTER and RENAME) in the specified DB2 system (ssid). The type of DB2 object is specified using the <object type> suffix.
9	<pfx>.AUTH.<privilege type>[3]	Allows a user to write audit log records for functions that issue SQL statements (such as GRANT, REVOKE) that explicitly alter DB2 privileges in the specified DB2 system (ssid). The type of DB2 privilege is specified using the <privilege type> suffix.
10	<pfx>.DB2CMD.<command type>[4]	Allows a user to write audit log records for functions that issue DB2 commands in the specified DB2 system (ssid).

Notes:

1. The prefix <pfx> for all resource names in this table is FILEM.AUDIT.<ssid>, where ssid is the DB2 subsystem or group ID.
2. See Table 30.
3. See Table 31 on page 148.
4. See Table 32 on page 148.

Table 30. Resource name suffixes for DB2 object types (DDL SQL statements)

DB2 Object Type	Resource Rule Name suffix
ALIAS	ALIAS
AUXILIARY TABLE	AUXTABLE
DATABASE	DATABASE
FUNCTION	FUNCTION

Table 30. Resource name suffixes for DB2 object types (DDL SQL statements) (continued)

DB2 Object Type	Resource Rule Name suffix
GLOBAL TEMPORARY TABLE	GBLTABLE
INDEX	INDEX
PROCEDURE	PROC
ROLE	ROLE
SEQUENCE	SEQUENCE
STOGROUP	STOGROUP
SYNONYM	SYNONYM
TABLE	TABLE
TABLESPACE	TBSPACE
TRIGGER	TRIGGER
TRUSTED CONTEXT	CONTEXT
TYPE	TYPE
VIEW	VIEW

Table 31. Resource name suffixes for DB2 privileges (GRANT and REVOKE SQL statements)

DB2 Authorization Type	Resource Rule Name suffix
COLLECTION	COLLECT
DATABASE	DATABASE
TYPE	TYPE
JAR	JAR
FUNCTION	FUNCTION
PACKAGE	PACKAGE
PLAN	PLAN
PROCEDURE	FUNCTION
SCHEMA	SCHEMA
SEQUENCE	SEQUENCE
SYSTEM	SYSTEM
TABLE	TABLE
USE	USE

Table 32. Resource name suffixes for DB2 commands

DB2 Command	Resource Rule Name suffix
ACCESS	ACCESS
ALTER	ALTER
ARCHIVE	ARCHIVE
BIND	BIND
CANCEL	CANCEL
DISPLAY	DISPLAY
FREE	FREE

Table 32. Resource name suffixes for DB2 commands (continued)

DB2 Command	Resource Rule Name suffix
MODIFY	MODIFY
REBIND	REBIND
RECOVER	RECOVER
REFRESH	REFRESH
RESET	RESET
RUN	RUN
SET	SET
START	START
STOP	STOP
TERM	TERM

FM/DB2 auditing FACILITY and XFACILIT class resource names

Chapter 17. Customizing FM/DB2 for national languages

You can customize FM/DB2 for national languages other than English. If you are using Japanese or Korean, and you have installed the FM/DB2 Japanese or Korean components, you might not need to perform any other customization for the Japanese or Korean national languages.

If you are using a language other than English, Japanese or Korean (for example, French), you will need to perform the customization tasks listed in Table 33. Perform **ALL** these steps (except step 9) if you plan to use a different language than English, Japanese or Korean.

You can also, optionally, translate the object list utility panel titles, column headings, and SQL statements. If you want to do this, perform step 9.

Table 33. Summary of steps for customizing FM/DB2 for a national language

Step	Description	Page
__ 1	Set the LANGUAGE option for batch processing, if required.	127
__ 2	Change the TERMTYPE option in FMN2POPT, if you are using a DBCS language.	322
__ 3	Confirm that you have the correct terminal type set in ISPF (ISPF option 0).	
__ 4	Create a print and display translation table for your language.	151
__ 5	Translate the File Manager message text to your language.	152
__ 6	Provide a version of FMN2MENU for your language.	152
__ 7	Translate the FM/DB2 ISPF messages to your language.	152
__ 8	Translate the FM/DB2 panels to your language.	153
__ 9	Translate the FM/DB2 object list utility headings and statements (FMN2DENU) to your language.	154

Changing the print and display translation tables for languages other than English

DBCS languages
This step is essential if you are using a DBCS language other than Japanese or Korean.

If you plan to use FM/DB2 with a national language other than English, you might need to provide a print and display translation table for your language. You do this as part of your customization for File Manager base function. See “Changing the print and display translation tables for languages other than English” on page 61.

You should also specify PRTTRANS=ON in FMN2POPT. If you are using any DBCS language you might also need to specify TERMTYPE=3270KN in FMN2POPT.

© Copyright IBM Corp. 2000, 2011

Translating the message text

All FM/DB2 messages are stored in the FMN2MENU source member. This CSECT is part of the root module so that an English version of the messages is always available. In addition, all messages used by FM/DB2 under ISPF are provided in the library, FMN.SFMNMENU. Using both FMN2MENU and members in FMN.SFMNMENU, you can provide your own set of translated messages. To use the messages you have translated, see "Using the translated messages and panels" on page 153.

To provide translated versions of the messages, you must provide a version of FMN2MENU for your language, as described in "Providing a national language version of FMN2MENU," **and** provide translated versions of the appropriate members in FMN.SFMNMENU, as described in "Translating the ISPF messages text." (You should not need to provide a Japanese or a Korean version of the messages if you have installed the FM/DB2 Japanese or Korean components.)

Providing a national language version of FMN2MENU

FMN2MENU contains the assembler source for the FM/DB2 messages. To provide translated versions of the messages:

1. Copy the member FMN2MENU from FMN.SFMNSAM1 to your own source library with the name FMN2M*yyy*, where *yyy* is one of the following language codes:

FRA	French
DEU	German
ITA	Italian
JPN	Japanese
PTG	Portuguese
ESP	Spanish
DAN	Danish
ENP	Upper case English
KOR	Korean
DES	Swiss German
CHT	Traditional Chinese
CHS	Simplified Chinese
XXX	Other

2. Change the message text in FMN2M*yyy* in your library.
3. Modify the FMN2UMDM member in FMN.SFMNSAM1 to meet your site's requirements, using the same language code as above. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN2UMDM.

Translating the ISPF messages text

All FM/DB2 ISPF messages are provided in English. They are also provided in Japanese if you have installed the FM/DB2 Japanese component, and in Korean if you have installed the FM/DB2 Korean component. You can translate some or all of these messages into another language.

All FM/DB2 ISPF messages are stored in FMN.SFMNMENU. You translate a message as follows:

1. Find the members in FMN.SFMNMENU that contain the messages you want to translate. The message members specific to FM/DB2 are all named FMND*zzzz* or FMN2*n*.

2. Create a library with the same characteristics as FMN.SFMNMENU, with the name FMN.SFMNM*yyy*, where *yyy* is the same language code you specified when you modified FMN2MENU. If you have already created a library with this name for translated File Manager base function messages, use that library. Copy the required message members from FMN.SFMNMENU to this library.
3. Change the required message texts in these members in your library.

To use the messages you have translated, see "Using the translated messages and panels." For more information about defining and using ISPF messages, see *z/OS ISPF Dialog Developer's Guide*.

Note: Be sure to include **ALL** the messages in the message members you copy to your own library. When FM/DB2 needs to display an ISPF message, it uses ISPF services to do this. Therefore the search for a message is made according to ISPF rules. Thus, if ISPF finds the message member it requires in your library, but the message number required is not in that member, ISPF will not look in any other library for the message, but will give an error. However, if you omit a complete message member from your library, then ISPF will use the English message member from the next library in the ISPMLIB concatenation. For more information about defining and using ISPF messages, see *z/OS ISPF Dialog Developer's Guide*.

Translating the panel text

All FM/DB2 panels are provided in English. They are also provided in Japanese if you have installed the FM/DB2 Japanese component, and in Korean if you have installed the FM/DB2 Korean component. You can translate some or all of these panels into another language. (If no translated version of a particular panel is available, FM/DB2 uses the English version.)

All FM/DB2 panels are stored in FMN.SFMNPENU. You translate a panel as follows:

1. Find the panel members in FMN.SFMNPENU that you want to translate. The panel members specific to FM/DB2 are all named FMN2*zzzz*.
2. Create a library with the same characteristics as FMN.SFMNPENU, with the name FMN.SFMNP*yyy*, where *yyy* is the same language code you specified when you modified FMN2MENU. If you have already created a library with this name for translated File Manager base function panels, use that library. Copy the required panel members from FMN.SFMNPENU to this library.
3. Change the required panel text in the members in your library. A panel may reference a **help** panel by means of a `.HELP` statement. If any panel you are changing contains any of these `.HELP` statements, also copy and change these referenced members in your library.

To use the panels you have translated, see "Using the translated messages and panels."

Using the translated messages and panels

To use your translated messages in a batch job, specify the appropriate language with the LANGUAGE processing option, using the keywords shown in Table 34 on page 154. See "Changing the default options" on page 126 for information on how to do this.

Table 34. Keyword values for the LANGUAGE option

Language	Code	Specify on LANGUAGE option...
French	FRA	FRENCH
German	DEU	GERMAN
Italian	ITA	ITALIAN
Japanese	JPN	JAPANESE
Portuguese	PTG	PORTUGUESE
Spanish	ESP	SPANISH
Danish	DAN	DANISH
Upper case English	ENP	UPPERENG
Korean	KOR	KOREAN
Swiss German	DES	SGERMAN
Traditional Chinese	CHT	CHINESET
Simplified Chinese	CHS	CHINESES
Other	XXX	OTHER

For example, to use French messages, specify LANGUAGE=FRENCH.

Under ISPF, the language used in messages and panels is determined by the setting of the national language for ISPF, for the current ISPF session. For information on changing the national language setting for your ISPF session, see *z/OS ISPF Dialog Developer's Guide*.

If your ISPF session is set up to use your language, you need to add your libraries to the appropriate ISPF concatenation, in front of any File Manager English libraries. For example, to use your translated messages, add FMN.SFMNM*yyy* to ISPMLIB, in front of FMN.SFMNMENU. To use your translated panels, add FMN.SFMNP*yyy* to ISPPLIB, in front of FMN.SFMNPENU.

Providing a national language version of FMN2DENU

FMN2DENU contains the assembler source for the FM/DB2 object list utility panel titles, column headings, and text used in expanded SQL statements. These are the titles, headings, and statements displayed when you select FM/DB2 option 3.4 and following options. These titles, headings, and statements can be translated to your national language.

Note: If you have installed the FM/DB2 Japanese component, these titles, headings and statements are provided in Japanese, in the module FMN2DJPN.

If you have installed the FM/DB2 Korean component, these titles, headings and statements are provided in Korean, in the module FMN2DKOR.

To translate these titles, headings or statements to your national language:

1. Copy the member FMN2DENU from FMN.SFMNSAM1 to your own source library with the name FMN2D*yyy*, where *yyy* is one of the following language codes:
   FRA French
   DEU German

ITA	Italian
JPN	Japanese
PTG	Portuguese
ESP	Spanish
DAN	Danish
ENP	Upper case English
KOR	Korean
DES	Swiss German
CHT	Traditional Chinese
CHS	Simplified Chinese
XXX	Other

2. Change the text in FMN2D*yyy* in your library as required, according to the following information:
   - Change the statement at about line 419, which reads:

     ```
     DC     CL8'FMN2DENU'
     ```

     to:

     ```
     DC     CL8'FMN2D yyy'
     ```

     where *yyy* is the language code you selected in step 1.
   - To change the panel headings, translate the wording in the statements starting `FMN2@TTL` and containing `TTL=...`
   - To change the column headings, translate the wording in the statements starting `FMN2@CHD` and containing `CHD=...`

     Where a ! separates two words, these two words appear on separate lines in the column heading.
   - To change the text used in expanded SQL statements, translate the wording between the quotation marks on the SETC statements, starting at about line 63.

   Do not change any statements other than those listed above.

   For example, in the `Tables, Views and Aliases` panel (FMN2POLT), to translate the panel title, you would change the statement:

   ```
   FMN2@TTL ISPFV=QSTBPT,TTL='Tables, Views and Aliases'
   ```

   To translate the column heading, TABLE SPACE NAME (where the words appear on three separate lines), you would change the statement:

   ```
   FMN2@CHD XRF=SCTSNAME,CHD='TABLE!SPACE!NAME'
   ```

   To translate the words Alias and Table for the OBJECT TYPE column, you would change the statements:

   ```
   &QTXT(4)   SETC 'Alias'
   ```

   and

   ```
   &QTXT(297) SETC 'Table'
   ```
3. Modify the FMN2UMDD member in FMN.SFMNSAM1 to meet your site's requirements, using the same language code as above. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN2UMDD.

Customizing for the Japanese national language

The other customization task you might need to do for the Japanese national language is to modify the supplied Japanese translation tables. If you want to do this, do so as part of the customization of the File Manager base function. See "Modifying the Japanese translation tables" on page 65.

Changing the Japanese message text

If you have installed the FM/DB2 Japanese component, all FM/DB2 Japanese messages are stored in the FMN2MJPN source member. Normally you should not need to modify this module. However, if you do want to modify it, you can do so by means of the usermod, FMN2UMDN.

To do this:

1. Copy the member FMN2MJPN from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN2MJPN in your library.
3. Modify the FMN2UMDN member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN2UMDN.

Customizing for the Korean national language

The other customization task you might need to do for the Korean national language is to modify the supplied Korean translation tables. If you want to do this, do so as part of the customization of the File Manager base function. See "Modifying the Korean translation tables" on page 66.

Changing the Korean message text

If you have installed the FM/DB2 Korean component, all FM/DB2 Korean messages are stored in the FMN2MKOR source member. Normally you should not need to modify this module. However, if you do want to modify it, you can do so by means of the usermod, FMN2UMDQ.

To do this:

1. Copy the member FMN2MKOR from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN2MKOR in your library.
3. Modify the FMN2UMDQ member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN2UMDQ.

Chapter 18. Verifying the customization of FM/DB2

After you have completed all the necessary steps involved in the initial installation and customization of FM/DB2, you need to perform the following steps to verify your customization, before completing the installation process. You also need to refer to the *File Manager User's Guide and Reference for DB2 Data*.

You verify the customization of FM/DB2 by running a batch job to define sample tables to your DB2 subsystem, and then using FM/DB2 to view and edit these tables, and define and drop new views. These jobs use sample data provided with DB2 in the DB2 sample library. It is also assumed that you have installed the EDITPROC DSN8EAE1 from DB2. Before starting the verification of FM/DB2, confirm that the EDITPROC, DSN8EAE1, is installed on your system. If it is not, refer to the DB2 publications for your DB2 system, to see how to do this.

Before you can do this, you must have customized FM/DB2 to access your DB2 subsystems. This means you must have bound the packages and plans for FM/DB2 in each DB2 subsystem, and customized the DB2 subsystem selection panel. If you have not already done this, refer to "Binding DB2 (required)" on page 119 and "Identifying all the DB2 systems that FM/DB2 will access and defining these in the FMN2POPT module (required)" on page 121, and carry out these tasks.

Step 1. Define DB2 objects to be used during verification

The first step in the verification is to define the DB2 objects to be used.

Edit and submit the batch job FMN2VER, to define various DB2 objects. Refer to the sample job for the changes you must make to this job before you can run it. You will have to provide the names of your DB2 libraries, sample plans and subsystem ID.

Check the results to ensure the job ran successfully. You should have created a storage group, a database, three tables, with indexes and views, and loaded these tables with some sample data.

Step 2. Start FM/DB2

Log on to a TSO userid that is enabled to access and use FM/DB2 and start FM/DB2.

If you have added an option for FM/DB2 to your ISPF Primary Options menu (see "Adding FM/DB2 to the ISPF menu" on page 107), enter this option value, and press `Enter`. For example, if you have assigned 'FD' to FM/DB2, enter 'FD' and press `Enter`.

If you defined FM/DB2 in an ISPF command table (see "Defining FM/DB2 in an ISPF command table" on page 107) verify that FM/DB2 can be started by entering the command FD on any ISPF command line.

If this is the first time you have used FM/DB2, the FM/DB2 DB2 Subsystem Selection menu is displayed, showing the DB2 subsystems you customized in your FMN2SSDM macro. Select the DB2 subsystem you want to use, and press Enter, to

© Copyright IBM Corp. 2000, 2011

go to the FM/DB2 Primary Option Menu. (If you have previously selected a DB2 subsystem, you will immediately see the Primary Option Menu.)

Note: The first time you use FM/DB2, a Copyright panel appears. After reading the panel text, press the Cancel key (PF12). In subsequent FM/DB2 sessions, this panel will not automatically appear.

Enter `ver` on the command line to display the release level and PTF level of FM/DB2. A panel is displayed, similar to this:

```
IBM File Manager for z/OS Version 9 Release 1
DB2 Component
(not APF authorized)

Service Levels of installed components

             Base       IMS        DB2        CICS
English      -NONE-     -NONE-     -NONE-     -NONE-
```

Notes:

1. This will always show FM/DB2 as `not APF authorized`, even if you have made File Manager APF-authorized, as File Manager cannot run APF-authorized under ISPF.
2. When you first install File Manager, `-NONE-` will be shown against each component. Subsequently, when you have applied service to File Manager, a PTF number will be shown, indicating the PTF level of each component you have installed. If you have not installed a component, that component will not be shown at all. If you have installed the Japanese or Korean language components of FM/DB2, another line will be displayed indicating the service level of that component.

Step 3. Edit table format and information

From the FM/DB2 Primary Option Menu, select option `2 Edit`. The **DB2 Edit** panel is displayed.

In the **owner** field enter `FMNDB2` and in the **name** field enter `DEPT`. Press `Enter`. You will see the DEPT table, in TABL mode.

Tab across to the **FORMAT** field at the top right hand side of the panel, and change `TABL` to `SNGL`. You will now see the first record with column headings and data.

Press PF11 repeatedly to see the rest of the rows in the table. Tab to the **FORMAT** field and change `SNGL` back to `TABL`.

Enter `I` in the prefix command area against a row, to insert another row, and press `Enter`.

Add new department information, tabbing across to each column. (In the LOCATION column ensure that there is an underscore, otherwise you could get an error when you attempt to commit the change.)

Press `PF3` to commit your change to the table. Press `PF3` again to return to the FM/DB2 Primary Option Menu.

From the FM/DB2 Primary Option Menu, select option 3 Utilities, and press Enter, and then 4 Object List, and press Enter.

In the **owner** field, enter FMNDB2 and leave the **name** field blank. Tab down to **Processing Options:** and under **Object Type** check that 3 is selected for **Table/view/alias**. (3 is the IBM default.) Press Enter.

The **Tables, Views and Aliases** panel is displayed, showing the tables and views you defined in FMN2VER, belonging to FMNDB2.

Tab down to the EMP table, in the prefix command area enter ?, and press Enter. The **Object List Line Commands** panel is displayed, showing the line commands available for a table.

Tab down to **COM Create a comment/remark** and enter S against this line, and press Enter. The **Comment Table** panel is displayed. Type a comment on the "Remarks" line and press Enter.

Press PF3 twice to return to the **Object List Utility** panel and then press Enter again. This will return you to the **Tables, Views and Aliases** panel.

Enter I against the EMP table, and press Enter. The **Table Details** panel for table EMP is displayed. At the bottom under **Associated Remarks:** you will see the comments you have entered for this table.

Press PF3 to return to the **Tables, Views and Aliases** panel.

Step 4. Display a view

In the **Tables, Views and Aliases** panel enter COL against the VHDEPT view, to show the columns for the view. There should be five columns, (DEPTNO, DEPTNAME, MGRNO, ADMRDEPT, and LOCATION) showing their names and characteristics. Press PF3 to return to the **Tables, Views and Aliases** panel.

Step 5. Change and run SQL

Now, in the **Tables, Views and Aliases** panel, enter VS ("Show how view created") against the VHDEPT view. Press Enter.

The **Edit and Execute SQL Statements from a Data Set** panel is displayed, showing a temporary data set that contains the SQL statements that were used to create the view.

Make sure that the option "Execute SQL from data set" is selected, and that the option "Re-edit data set after execution" is not selected. Press Enter.

You now have an ISPF edit session, where you can edit the SQL statements. Make a change to these statements, (for example, remove the MGRNO statement) and change the name of the view on the create statement. (For example, you could change the name VHDEPT to VZDEPT.)

Press PF3 to save your changes and execute the new SQL statements to create your new view. You are returned to the **Edit and Execute SQL Statements from a Data Set** panel and you should see the message CREATE statement executed at the bottom of the panel.

Step 5. Change and run SQL

Press PF3 to return to the **Tables, Views and Aliases** panel. Press PF3 again to return to the **Object List Utility** panel.

Press Enter to display a new list of objects belonging to FMNDB2, and your new view will be in the list.

Enter COL against the new view, VZDEPT, to show the columns for this view. There should now be four columns, (DEPTNO, DEPTNAME, ADMRDEPT, and LOCATION), since you removed the MGRNO statement. Press PF3 to return to the **Tables, Views and Aliases** panel.

Step 6. Drop the new view

Tab to this view, and enter DR against it, to drop the view, and press Enter. The **Drop View** panel is displayed.

Press Enter, and you will see the **Drop Confirmation** panel. Press Enter, and the view will be dropped.

Press PF3 three times, to return to the **Object List Utility** panel. Press Enter to display the **Tables, Views and Aliases** panel, and your new view will no longer be in the list.

Press PF3 repeatedly until you return to the FM/DB2 Primary Option Menu.

Complete the installation

You can now complete the installation of FM/DB2 by performing the ACCEPT processing. The steps involved are described in the *File Manager Program Directory*.

Part 3. Customizing File Manager IMS Component

© Copyright IBM Corp. 2000, 2011

Chapter 19. Preparing to customize FM/IMS

Before you can install and customize FM/IMS, you must have installed the File Manager base function. This and the following chapters assume you have installed FM/IMS into the same target and distribution libraries as File Manager base function. You could encounter problems using File Manager if you do not install FM/IMS in the same libraries as the base function.

Furthermore, before you can use FM/IMS you must customize FM/IMS and the operating environment. You may also need to customize the File Manager base function. For information on customizing the File Manager base function, see Part 1, "Customizing File Manager," on page 1.

Table 35 lists the customization tasks you can perform for FM/IMS. You should review the following sections in this chapter, and read the referenced sections in the table, to determine the customization you need to do.

Checklist for installing and customizing FM/IMS

Table 35. Summary of steps for customizing FM/IMS and the operating environment

	Description	Page
__ 1	Add FMN1PARM to SYS1.PARMLIB.	169
__ 2	Concatenate FM/IMS libraries to the LINKLIST.	171
__ 3	Modify the TSO logon procedure	171
__ 4	Authorize FM/IMS.	172
__ 5	Add FM/IMS to the ISPF menu.	173
__ 6	Define FM/IMS in an ISPF command table.	173
__ 7	Customize IMS to support the use of dynamic PSBs in BMP mode.	174
__ 8	Customize the FM/IMS installation options module	177
__ 9	Tailor the job control skeletons.	189
__ 10	Customize the FM/IMS security exit, FMN1SXT.	196
__ 11	Provide access to protected functions using RACF.	191
__ 12	Decide how to customize the FM/IMS audit facility.	171
__ 13	Customize FM/IMS for national languages.	213

Supported databases

FM/IMS supports the following databases:

HDAM Hierarchical Direct Access Method
HIDAM Hierarchical Index Direct Access Method
HISAM Hierarchical Index Sequential Access Method
SHISAM Simple Hierarchical Index Sequential Access Method
HSAM Hierarchical Sequential Access Method (BSAM or QSAM only)
SHSAM Simple Hierarchical Sequential Access Method (BSAM or QSAM)
INDEX Index database
DEDB Data Entry Database (Fast Path)

© Copyright IBM Corp. 2000, 2011

MSDB	Main Storage database (Fast Path)
PHDAM	Partitioned Hierarchical Direct Access Method (HALDB)
PHIDAM	Partitioned Hierarchical Index Direct Access Method (HALDB)
PSINDEX	Partitioned Secondary Index (HALDB)

Region types

FM/IMS offers two modes of accessing IMS databases - DLI mode and BMP mode.

DLI mode

When an FM/IMS function is run in DLI mode, a DL/I batch processing program issues the DL/I calls. The DL/I batch processing program executes as a subtask of the FM/IMS function.

You use DLI mode when databases are offline.

If you plan to run FM/IMS functions in DLI mode, you must specify the following details when you are customizing the FM/IMS installation options module:
- The IDs of the IMS subsystems you plan to access in DLI mode.
- The DLIBATCH parameters that are to be passed to the IMS region controller when accessing each subsystem.
- The names of the DFSVSAMP, RESLIB, IMS macros and staging ACBLIB data sets for each subsystem.
- The names of the PSB, DBD and Template libraries for each subsystem.
- Various processing options that you want the functions to use when accessing each subsystem in DLI mode.

For information on how you do this, see “Customizing the FM/IMS installation options module” on page 177.

BMP mode

When an FM/IMS function is run in BMP mode, a batch message processing (BMP) program issues the DL/I calls. The BMP executes as a subtask of the FM/IMS function.

You use BMP mode when databases are online.

If you plan to run FM/IMS functions in BMP mode, you must specify the following details when you are customizing the FM/IMS installation options module:
- The IDs of the IMS subsystems you plan to access in BMP mode.
- The IMSBATCH parameters that are to be passed to the IMS region controller when accessing each subsystem.
- The names of the RESLIB, IMS macros and DOPT ACBLIB data sets for each subsystem.
- The names of the PSB, DBD and Template libraries for each subsystem.
- Various processing options that you want the functions to use when accessing each subsystem in BMP mode.

For information on how you do this, see “Customizing the FM/IMS installation options module” on page 177.

PSB types

FM/IMS supports the use of two types of PSBs - dynamic PSBs and static PSBs.

Dynamic PSBs

Dynamic PSBs are temporary PSBs generated by FM/IMS when a function is started and deleted when it finishes. FM/IMS generates a PSB that is sensitive to all the segments in the specified DBD. If you don't want your function to be sensitive to all segments in a physical DBD, you can use a logical DBD that only includes the physical segments that you want the function to process. For the Extract and Load functions, the PSB is generated with PCBs for the primary database and all logically related databases.

Advantages in using a dynamic PSB

- A database administrator is not required to generate PSBs for the FM/IMS functions.
- FM/IMS generates a PSB that best suits the FM/IMS function.

Disadvantages in using a dynamic PSB

- The same processing option (PROCOPT) is used for all segments in the DBD. For example, you cannot prevent the Edit from updating one segment type in the DBD.
- A dynamic PSB can access any database in the subsystem, so unless special measures are taken they can be a security risk. See "Securing dynamic PSBs in BMP mode" on page 167 for a description of some special measures that can be taken to remove this security risk.
- The PSB is not available for use if Batch Backout is required.

IMS system considerations for BMP support

If you plan to use dynamic PSBs to access an IMS subsystem in BMP mode, you must:

- Include in the system definition an APPLCTN macro statement for each dynamic PSB name that FM/IMS will use. For information on how you do this, see "Declaring the dynamic PSBs" on page 174.
- Provide a DOPT ACBLIB data set into which FM/IMS can dynamically build PSBs. For more information on how you do this, see "Providing a DOPT ACBLIB data set" on page 174.

Static PSBs

Static PSBs are existing PSBs generated by the database administrator.

Advantages of using a static PSB

- It is easier to exclude segments from processing.
- Processing options (PROCOPTs) can be used to limit the update capability of the user.
- They are easily secured.
- The PSB is available for use if Batch Backout is required.

Disadvantages of using a static PSB

- A database administrator must generate the PSB.

IMS system considerations for BMP support

If you plan to use a static PSB to access an IMS subsystem in BMP mode, you must:

- Include an APPLCTN macro statement for the PSB in the system definition.
- Perform an ACBGEN for the PSB.

IMS region controller parameters

FM/IMS functions that access databases invoke the IMS region controller as a subtask. You specify the default values for the parameters passed to the IMS region controller when you customize the FM/IMS installation options module.

In the FM/IMS installation options module customization, you specify defaults for the following options:

- Whether or not FM/IMS functions use Database Recovery Control (DBRC) when they are run in DLI mode.
- Whether or not FM/IMS functions use an Internal Resource Lock Manager (IRLM) when they are run in DLI mode.
- The 4-byte z/OS subsystem name assigned to the Internal Resource Lock Manager.
- Whether or not dynamic backout is to be performed when an IMS pseudo-abend occurs in an FM/IMS function running in DLI mode.
- (If the IMS subsystem is part of a Remote Site Recovery (RSR) complex) the Global Service Group (GSG) name and the Transport Manager Instance (TMI) name that FM/IMS functions are to use when they are run in DLI mode.
- The parallel DL/I option that FM/IMS functions are to use when they are run in BMP mode.
- The number of Fast Path database buffers to be made available in the Common Service Area (CSA) when a Fast Path region is activated.
- The number of additional page-fixed buffers to be made available to a Fast Path region if the normal allotment is used.
- The maximum number of locks that an FM/IMS function is allowed to hold at one time.

In the FM/IMS installation options module customization, you also specify whether or not the user can override the values that you specify for the above options.

For more information on specifying the IMS region controller parameters, see “Specifying the parameters passed to the IMS region controller” on page 180.

Security

FM/IMS offers the following security features.

IMS subsystem and FM/IMS function security

FM/IMS allows you to protect these resources:

- The update or read-only functions
- Individual FM/IMS functions
- Access to individual IMS subsystems by the update or read-only functions
- Access to individual IMS subsystems by individual functions

For information on how you secure these resources, see “Controlling access to IMS subsystems and FM/IMS functions” on page 191.

Protected functions

A number of functions in FM/IMS are protected by default. They are listed in Table 43 on page 191. If you want to use these functions, you must give your users access to them using RACF (or an equivalent security product). For more information, see “Controlling access to IMS subsystems and FM/IMS functions” on page 191.

The FMN1SXT security exit

FMN1SXT is a customizable exit. During the processing of FM/IMS functions, FMN1SXT is invoked at the following points:

- Prior to invoking the IMS region controller.
- When allocating a database data set (DLI mode only).
- Prior to allocating an audit trail data set.
- After the IMS region controller terminates.

The version of FMN1SXT as distributed with FM/IMS performs no security checking and simply returns control. You can write your own version of FMN1SXT to perform various functions, for example to:

- Edit, modify or change the disposition of a database data set from OLD to SHR.
- Force the creation of an audit trail.

For more information on writing your own version of FMN1SXT, see "Customizing the FM/IMS security exit" on page 196.

AGN security

FM/IMS supports subsystems that use Application Group Names (AGNs) to secure dependent regions. You must specify whether or not a subsystem uses AGN security when you are customizing the FM/IMS installation options module. For information on how you do this, see "Specifying AGNS" on page 183. You also have the option of specifying the AGNs that you want FM/IMS to use. Information on how you do this can also be found in Specifying AGNS.

RAS security

FM/IMS supports subsystems that use Resource Access Security (RAS). No special customization steps are required to provide this support.

Securing dynamic PSBs in BMP mode

Dynamic PSB support enables a user to use FM/IMS to generate a PSB and an ACB that provide access to any database specified by the user. Furthermore, FM/IMS dynamically creates a PSB/ACB name using parameters supplied in the options macro, and these names are shared amongst all FM/IMS users. Consequently special methods must be employed to secure the BMP regions.

You can use one of the following methods to prevent unauthorized access to databases by the use of dynamic PSBs in BMP mode:

- Provide a version of FMN1SXT to prevent unauthorized users from accessing specific databases via a dynamic PSB in BMP mode.
- Authorize the ACBs dynamically generated by FM/IMS to an AGN and use RACF (or an equivalent security product) to give selected userids the authority to use that AGN. Thus only selected users have the authority to use a dynamic PSB in BMP mode to access any database.

IMS subsystem access

FM/IMS allows you to control the way in which your IMS subsystems are accessed as described in the following sections.

Read only subsystems

FM/IMS allows you to define a subsystem as read-only. If a subsystem is defined as read-only, only the FM/IMS non-update functions - Browse, Batch Browse, Extract and Print - can be run against databases in the subsystem; FM/IMS prevents users running the update functions - Edit, Batch Edit, Load, Initialize and Delete/Define.

BMP mode only and DLI mode only subsystems

FM/IMS allows you to define a subsystem as BMP mode only or DLI mode only. If a subsystem is defined as BMP mode only, databases in the subsystem can only be accessed in BMP mode. If a subsystem is defined as DLI mode only, databases in the subsystem can only be accessed in DLI mode.

Static PSB only and dynamic PSB only subsystems

FM/IMS allows you to define a subsystem as static PSB only or dynamic PSB only. If a subsystem is defined as static PSB only, then only static PSBs can be used to access databases in the subsystem. If a subsystem is defined as dynamic PSB only, then only dynamic PSBs can be used to access databases in the subsystem.

Dynamic allocation only subsystems

FM/IMS allows you to define a subsystem as dynamic allocation only. If a subsystem is defined as dynamic allocation only, FM/IMS enforces the use of the database data sets specified in the DFSMDA dynamic allocation modules, when accessing databases in the subsystem in DLI mode.

For more information on these controls, see “Controlling access to the subsystems” on page 181 and Appendix C, “FM/IMS options,” on page 361.

Releasing locks for unmodified data in Edit and Browse sessions

Before returning control to the user, the Edit and Browse functions release any locks that are being held for unmodified data.

If you are running IMS V11, IMS V10, IMS V9 with PTF UK25100 applied or IMS V8 with PTF UK23703 applied, then FM/IMS uses the RLSE call to release the locks.

However, if you are running IMS V9 without PTF UK25100 applied, or IMS V8 without PTF UK23703 applied, and the root segment of the database has a key, FM/IMS issues a GET UNIQUE call, qualified at the root level, with a greater-than operator or an equal-to operator for a key of X'FF' (high values).

This method of releasing the locks may fail, because not all randomizing routines for HDAM and DEDB databases can handle this key value.

The following are the possible outcomes of the randomizing routine not handling this key value:

1. A return code of 4 from the randomizing routine which results in IMS returning a status code of FM.
2. A return code from the randomizing routine that is not 0 or 4. This results in an IMS U1021 abend (if it is a DEDB database) or an IMS U0812 abend (if it is a HDAM database).
3. The randomizing routine abends.

FM/IMS can recover from case 1, but not case 2 and 3. In cases 2 and 3, FM/IMS displays the popup in Figure 8.

```
The following abend has occurred while FM/IMS was
attempting to free the lock the PCB has on the current
database record:

IMS Abend Code:         System Abend Code:

Reason Code . :

Apply APAR PQ92045(UK23703) to IMS V8 or
APAR PK24342(UK25100) to IMS V9.

Press ENTER, PF3 or PF12 to return to FM/IMS.
```

Figure 8.

If this panel is displayed, you should apply PTF UK25100 to IMS V9, or PTF UK23703 to IMS V8.

It is recommended that you apply these PTFs even if you are not getting an IMS abend, because the GU call that FM/IMS uses when the RLSE call is not available, is not as reliable as the RLSE call. In some circumstances, the GU call may not release the locks.

Releasing locks for modified data in Edit sessions

If you want the Edit to release any locks that are being held for modified data before returning control to the user, you should set AUTOSAVE=Y, EDITFREQ=1 and UAUTOSAV=N in the FM/IMS installation options module. You can specify these settings for some IMS subsystems, so all locks for modified data are released when the user is editing databases in those subsystems, and specify different settings for other IMS subsystems, where it is not as critical that these locks are released. You can also specify different settings for BMP and DLI mode. For more information on how to code these parameters, see "Specifying selected processing options" on page 182 and Appendix C.

Note: If you set AUTOSAVE=Y, EDITFREQ=1 and UAUTOSAV=N, editing changes are committed when the user presses `Enter`. Thus these settings preclude users from using the UNDO or CANCEL command to discard their editing changes.

Defining the FMN1PARM member

If auditing is to be controlled from parmlib (user has read access to FILEM.PARMLIB.IMS, see "Specifying that FM/IMS audit logging is to be controlled by SAF" on page 210), then member FMN1PARM must be defined in SYS1.PARMLIB (or any other library in the logical parmlib concatenation) as follows.

Default parmlib member FMN1PARM is provided in the SFMNSAM1 library. Copy this member to the appropriate system parmlib library.

Note: The sample FMN1PARM member supplied in SFMNSAM1 also includes an FMSECRTY statement. This option is not used in FM/IMS Version 11, and can be either omitted, or commented out. It has no effect.

FM/IMS will not start if the required parmlib member does not exist in the system logical parmlib concatenation for users to be audited via parmlib.

There are two methods that can be used to include the FMN1PARM member in a library in the logical parmlib concatenation. The choice of method depends on whether the installation's security software is configured to allow FM/IMS users READ access to the data set SYS1.PARMLIB.

Method 1 can only be used when FM/IMS users have read access to SYS1.PARMLIB.

Method 2 can be used regardless of whether FM/IMS users have READ access to SYS1.PARMLIB or not, and must be used when FM/IMS users do not have READ access to SYS1.PARMLIB.

Method 1

Place the FMN1PARM member in any library in the current logical parmlib concatenation. No IPL or other action is required to activate the new member (unless a new library was added to the logical parmlib concatenation).

Notes:

1. Method 1 cannot be used in any situation where FM/IMS users do not have READ access to SYS1.PARMLIB. For example, when FM/IMS users have READ access to another library in the logical parmlib concatenation, and the FMN1PARM member is placed in the latter library. This will not work. The key issue is whether the FM/IMS user has READ access to SYS1.PARMLIB.

Method 2

This method must be used when FM/IMS users do not have READ access to SYS1.PARMLIB.

1. Create a new library with dataset attributes similar to SYS1.PARMLIB.
2. The library name for this data set must include the string "FMNPARM" in one of the qualifiers. You can choose any data set name that meets this requirement. Examples of suitable data set names are:
   SYS1.PARMLIB.FMNPARM SYS8.FMNPARM.PARMLIB
   FMNPARM.SYS8.PARMLIB SYS2.FMNPARMS.LIB
   SYS8.XFMNPARM.PARMLIB
3. Add member FMN1PARM to the new library, specifying the appropriate FMAUDIT parameter.
4. Add the new library to the logical parmlib concatenation. This can be done dynamically, or by means of a system IPL.

Note: When Method 2 is used, the FMN1PARM member must be located in the library created in Step 1 (as described above). In addition, if the FMN1PARM member specifies any include statements (see “Facilities for customizing the FMN1PARM definitions” on page 419), all the included members must also reside in the library created in Step 1.

You use the FMN1PARM member to define:

- Whether FM/IMS uses SAF to control FM/IMS audit logging.
- The SAF resource name prefix to be used by FM/IMS when determining access to various resources.
- Whether FM/IMS loads the FMN1POPT module from a specific library.

For more information, see "Deciding how to customize the FM/IMS audit facility."

Alternatives for making FM/IMS available

You can make FM/IMS available to your users either by concatenating FMN.SFMNMOD1 to your linklist, or adding it to the STEPLIB DD statement in your TSO logon procedure.

To make FM/IMS readily available from ISPF, configure your ISPF environment as described in Chapter 2, "Customizing the operating environment for File Manager," on page 11.

Concatenating libraries to the LINKLIST

To make FM/IMS commonly available, add the FMN.SFMNMOD1 library to your concatenated LINKLIST. If you did not do this for File Manager base function, add this library to either your LNKLST*xx* or PROG*xx* member in SYS1.PARMLIB.

Modifying the TSO logon procedure

If you made File Manager base function available to TSO, then you do not need to make any further changes to TSO for FM/IMS. Otherwise, add the File Manager libraries to your TSO logon procedure, as described "Modifying the TSO logon procedure" on page 7.

Deciding how to customize the FM/IMS audit facility

You have two choices as to how you control auditing of FM/IMS audit activities:

Use FMN1POPT controlled audit logging

With this method, you control audit logging the same way you control it in previous versions of FM/IMS, that is, by specifying the required values for the IMSAUDLG parameter on the FMN1POPI macro statement and the SMFNO parameter on the FMN0POPI macro statement. See Appendix C, "FM/IMS options," on page 361 for information on how to code the IMSAUDLG parameter, and see Appendix A, "File Manager options," on page 289 for information on how to code the SMFNO parameter.

These points summarize the facilities available with FMN1POPT controlled auditing:

- The FM/IMS Edit function provides audit logging support, but the other FM/IMS functions do not create audit trails.
- You can specify different audit settings (such as whether or not auditing is required) for each IMS subsystem that FM/IMS accesses.
- The audit settings specified for any IMS subsystem apply equally to all FM/IMS users accessing that IMS subsystem.
- The audit settings specified for any IMS subsystem apply equally to all databases within that IMS subsystem.
- The **Create audit trail** option on the Edit Entry panel allows users to request audit logging of their Edit sessions when audit logging is not required.
- You can specify audit logging to SMF or to the user's audit log data set, but this is an installation-wide setting and you can only get logging to both the user's log data set and SMF if you specify logging to SMF and you request that the audit log is printed at the end of the Edit session.

(See the IMSAUDLG parameter in Appendix C, "FM/IMS options," on page 361 for information on how you do this.)

Use System Authorization Facility (SAF) controlled audit logging
With this method, audit logging is controlled by RACF (or an equivalent security product) and FACILITY and XFACILIT class profiles that you define.

These points summarize the facilities available with SAF-rule controlled auditing:

- All FM/IMS functions that access IMS databases provide audit logging support.
- You can specify different audit settings (such as whether or not auditing is required) for each IMS subsystem that FM/IMS accesses.
- You can specify different audit settings for different FM/IMS users.
- You can specify different audit settings for different databases.
- You can specify different audit settings for each FM/IMS function.
- You can control whether or not the **Create audit trail** option on the Edit Entry can be used:
  - To request an audit trail when one is not required.
  - To stop an audit trail being created when one is required.
- You can specify audit logging to SMF, to the user's audit log data set or, for Edit and Browse only, to the user's audit log data set with automatic (mandatory) printing of the audit log at the end of the session. You can also specify dual logging (to the user's audit log data set and to SMF).

Some other points to consider are:

- Audit logging to SMF requires additional set-up, but provides a more reliable and secure environment for capturing audit information than audit logging to the user's audit log data set.
- If an attempt to write an audit log record to SMF or the user's log data set fails, the FM/IMS function terminates.

If you want to implement FMN1POPT controlled audit logging at your installation, follow the instructions in Chapter 23, "Customizing the FM/IMS audit facility (FMN1POPT controlled auditing)," on page 201.

If you want to implement SAF-rule controlled auditing at your installation, follow the instructions in Chapter 24, "Customizing the FM/IMS audit facility (SAF-rule controlled auditing)," on page 205.

Planning for running FM/IMS with APF-authorization

If you intend to write FM/IMS audit log records to SMF, then you must ensure the File Manager load module, FMNSMF, is APF-authorized. If you have not made File Manager base function APF-authorized, then either add FMN.SFMNMOD1 to your list of authorized libraries, or copy FMNSMF into an authorized library.

Chapter 20. Customizing the operating environment for FM/IMS

This chapter describes how to customize the operating environment for FM/IMS. You do this after you have installed FM/IMS.

Modifying the ISPF environment

To make it easy to start FM/IMS under ISPF, configure your ISPF environment as described in the following sections.

Adding FM/IMS to your ISPF menu

To add FM/IMS to your ISPF Primary Option Menu panel (ISR@PRIM), insert the additional lines (◄ **New**) as shown in Figure 9. You could add FM/IMS in your Primary Option Menu after the File Manager base function.

```
   ⋮
)BODY  CMD(ZCMD)
   ⋮
 9  IBM Products  IBM program development products
 10 SCLM          SW Configuration Library Manager
 11 Workplace     ISPF Object/Action Workplace
 F  File Manager  File Manager for z/OS
 FI FM/IMS        File Manager/IMS              ◄ New
   ⋮
)PROC
   ⋮
 &ZSEL = TRANS( TRUNC (&ZCMD,'.')
   ⋮
  9,'PANEL(ISRDIIS) ADDPOP'
 10,'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
 11,'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
  F,'PANEL(FMNSTASK) SCRNAME(FILEMGR) NEWAPPL(FMN)' /* File Manager */
 FI,'PANEL(FMN1ST00) SCRNAME(FMIMS) NEWAPPL(FMN1)' /* File Manager IMS */ ◄ New
   ⋮
```

Figure 9.

For more information about configuring your ISPF Primary Option Menu panel, see *z/OS ISPF Planning and Customizing*.

Defining FM/IMS in an ISPF command table

ISPF supports four different command tables where you can define an ISPF command to invoke FM/IMS:

- Application command table
- User command table
- Site command table
- System command table

You can use the ISPF Command Table Utility (option 3.9) to create or change a command table that is not currently in use (the system command table, ISPCMDS,

© Copyright IBM Corp. 2000, 2011

is always in use). When you add a command for FM/IMS to one of these command tables, you can invoke FM/IMS from any ISPF panel without prefixing the command with TSO.

Add the following entry to a command table to enable FM/IMS to be run from any ISPF panel by typing 'FI' on the command line.

Verb: FI

Action: SELECT PANEL(FMN1ST00) OPT(&ZPARM) SCRNAME(FMIMS) SUSPEND NEWAPPL(FMN1)

Description: File Manager for z/OS IMS component

For information about ISPF command tables, see *z/OS ISPF Planning and Customizing*. For information about the ISPF Command Table Utility, see *z/OS ISPF User's Guide Vol II*.

Customizing IMS to support the use of dynamic PSBs in BMP mode

If you plan to use dynamic PSBs to access an IMS subsystem in BMP mode, there are two customizing tasks you must perform on the IMS subsystem:

- Declare the dynamic PSBs
- Provide a DOPT ACBLIB data set

Declaring the dynamic PSBs

You must include in the system definition an APPLCTN macro statement for each dynamic PSB name used by FM/IMS functions when run in BMP mode. You use:

- The DYNPRFN parameter on the FMN1POPI macro statement to specify the number of dynamic PSB names that FM/IMS can use to access the subsystem. (See "DYNPRFN" on page 368).
- The DYNPRFX parameter on the FMN1POPI macro statement to specify the first 1- to 5-characters of the dynamic PSB name. (See "DYNPRFX" on page 368).

The dynamic PSB names that FM/IMS uses are obtained by combining the 1- to 5-characters you specify in the DYNPRFX parameter with a three digit number in the range 001 to the number specified in the DYNPRFN parameter.

For example, if you specify DYNPRFN=3 and DYNPRFX=FMN on the FMN1POPI macro statement for the subsystem, you will need to include the following APPLCTN macro statements in the system definition:

```
APPLCTN PSB=FMN001,PGMTYPE=BATCH,DOPT
APPLCTN PSB=FMN002,PGMTYPE=BATCH,DOPT
APPLCTN PSB=FMN003,PGMTYPE=BATCH,DOPT
```

PGMTYPE=BATCH is required because the PSBs are for BMPs. The DOPT parameter specifies that it is a dynamic PSB, so it is also required.

Providing a DOPT ACBLIB data set

You must provide a DOPT ACBLIB data set into which FM/IMS can dynamically build PSBs.

You use the DYNACB parameter on the FMN1POPI macro statement to specify the DOPT ACBLIB data set. You must concatenate the specified data set after the primary ACBLIB data set in the IMS execution JCL (that is, the JCL that starts the IMS control region).

Chapter 21. Customizing FM/IMS

This chapter describes how to customize FM/IMS. You do this after you have installed FM/IMS.

Customizing the FM/IMS installation options module

Before you can use FM/IMS, you must customize the installation options module, FMN1POPT. In FMN1POPT you specify:

- The IDs of the IMS subsystems that are to be accessed by FM/IMS.
- The parameters that are to be passed to the IMS region controller when a function starts a BMP, Fast Path or DL/I batch processing region.
- The names of the PSBLIB, DBDLIB, DFSVSAMP, RESLIB, IMS macros, staging ACBLIB, RECON, DOPT ACBLIB and Template data sets for each subsystem.
- Selected processing options that you want the FM/IMS functions to use.
- Whether or not the user can override the values you specify for selected parameters, data sets, and processing options.

A version of FMN1POPT is distributed with FM/IMS, but you will need to replace this module with your own version of FMN1POPT.

You use the usermod FMN1UMDP to install your version of FMN1POPT. FMN1UMDP and the source for FMN1POPT are distributed in FMN.SFMNSAM1. To create and install your own version of FMN1POPT:

1. Copy the member FMN1POPT from FMN.SFMNSAM1 into your own source library. The member FMN1POPT contains the sample source for the FMN1POPT module.
2. Modify your copy of the FMN1POPT sample source, as described in the following sections.
3. Modify the FMN1UMDP member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMN1UMDP.

Note: You can also use the sample job FMN1POPH to assemble FMN1POPT if you do not want to use SMP/E.

© Copyright IBM Corp. 2000, 2011

Important information

Notes:

1. If you are migrating to FM/IMS version 11 from an earlier version of FM/IMS, you cannot use the FMN1POPT module that was created for the earlier version. If you do, you may get unpredictable results.
2. If you are migrating to FM/IMS Version 11 from FM/IMS version 10, the FMN1POPT module is most easily created by reassembling and link-editing the source code you used to create the version 10 FMN1POPT module.
3. If you are migrating to FM/IMS Version 11 from FM/IMS Version 8 or version 9, the FMN1POPT module is more easily created, if you use as a starting point the source code that you used to create the version 8 or Version 9 FMN1POPT module. The FM/IMS Version 11 macros do have some new parameters, however, and you - almost certainly - will need to specify values for some of these parameters. So, before you reassemble and link-edit the FM/IMS Version 8 or version 9 source code, review the list of new parameters on the FMN1POPD and FMN1POPI macro statements in Table 36 and when you don't want the default values for them, specify the required values for them on the relevant macro statement.
4. If you are migrating to FM/IMS Version 11 from a version of FM/IMS earlier than version 8, IBM recommends that you the FMN1POPT sample source provided and that you modify it as described in this chapter.

Table 36. FMN1POPD and FMN1POPI macro statement parameters added in FM/IMS Version 10

Parameter	Macro statement	Page number
DBDLIB1-6	FMN1POPD and FMN1POPI	364
LKEYxxxx	FMN1POPD	374 to 376
LOGxxxx	FMN1POPD	377 to 379
PSBLIB1-6	FMN1POPD and FMN1POPI	383
RKEYxxxx	FMN1POPD	387 to 389
SFMNSLIB	FMN1POPD	389
TPLLIB1-6	FMN1POPD and FMN1POPI	390
UDBDLIB	FMN1POPD and FMN1POPI	393
UPSBLIB	FMN1POPD and FMN1POPI	403
UTPLLIB	FMN1POPD and FMN1POPI	405
XKEYxxxx	FMN1POPD	407 to 409

The FMN1POPT sample source includes five FM/IMS macro statements:
- FMN0POPI
- FMN1POPD
- FMN1POPI
- FMN1AGNT
- FMN1END

You use these macro statements to specify the details that FM/IMS requires.

FMN0POPI macro

You use the FMN0POPI macro statement, to specify the non-IMS options. This macro statement is also included in the installation option modules for the other File Manager components. But you do not have to use the same options in FM/IMS as you do in the other components.

The FMN0POPI macro statement is required and it must be the first statement in the FMN1POPT module. You can only have one FMN0POPI macro statement in the FMN1POPT module.

The parameters on the FMN0POPI macro statement in the sample FMN1POPT specify the default values. The parameters that FM/IMS does not use are omitted. All the parameters are optional. For detailed information on these parameters, see Appendix A, “File Manager options,” on page 289. You use this information to decide the required settings for these parameters at your installation.

FMN1POPD macro

You use the FMN1POPD macro statement to specify the FM/IMS installation defaults. The FMN1POPD macro statement is required and it must be the second statement in the FMN1POPT module. You can only have one FMN1POPD macro statement in the FMN1POPT module.

The parameters on the FMN1POPD macro statement in the sample FMN1POPT specify the default values. All the parameters are optional. Each one is described in this chapter. For more detailed information, see Appendix C, “FM/IMS options,” on page 361.

FMN1POPI macro

You use the FMN1POPI macro statement to specify the values that are to be used when FM/IMS accesses an IMS subsystem. One FMN1POPI macro statement is required for each IMS subsystem that is to be accessed by FM/IMS. They must be included after the FMN1POPD macro statement and before all FMN1AGNT macro statements.

All the parameters on the FMN1POPI macro statement except the SSID parameter are optional. Each one is described in this chapter. For more detailed information, see Appendix C.

Note: Prior to FM/IMS version 8 the FMN1POPI macro statement specified values that were used when FM/IMS accessed a subsystem in BMP mode. Also an FMN1POPI macro statement was only required for a subsystem, if that subsystem was to be accessed in BMP mode. However, with FM/IMS version 8 and above this is no longer the case. In version 8, the FMN1POPI macro statement was changed to specify values for BMP mode and DLI mode.

FMN1AGNT macro

If a subsystem uses AGNs (Application Group Names) to secure dependent regions, you use FMN1AGNT macro statements to specify the AGNs that you want FM/IMS to use when it accesses this subsystem in BMP mode. One FMN1AGNT macro statement is required for each AGN you want to specify.

This macro statement is optional. Specifying AGNS for a subsystem is only required when you want to either:

- Limit the AGNS that FM/IMS functions can use when accessing this subsystem in BMP mode, or
- Allow users to be able to select from a list of AGNs when they specify this subsystem (and a BMP region type) on the FM/IMS functions entry panel.

If you do want to include FMN1AGNT macro statements in the FMN1POPT module, include them after all the FMN1POPI macro statements and before the FMN1END macro statement.

All the parameters on the FMN1AGNT macro statement except the DESC parameter are required. Each one is described in this chapter. For more detailed information, see Appendix C.

FMN1END macro

The FMN1END macro statement is required and it must be the last statement in FMN1POPT. You can only have one FMN1END macro statement in the FMN1POPT module. This macro statement does not have any parameters.

Specifying the IMS subsystems

Include in the FMN1POPT module, one FMN1POPI macro statement for each IMS subsystem that is to be accessed by FM/IMS and use the **SSID** parameter on each statement to specify the subsystem ID. The **SSID** parameter is required.

Use the **DESC** parameter to specify a description of the subsystem. The description is displayed on the Subsystem Selection panel. The **DESC** parameter is optional.

You can use the FMN1POPD macro statement to specify an installation default for all the other parameters of the FMN1POPI macro statement. These parameters are optional. When a parameter is not specified on the FMN1POPI macro statement, FM/IMS uses the installation default (if specified) or the FM/IMS default (otherwise). So you only need to specify a parameter on the FMN1POPI macro statement if it differs from the installation or FM/IMS default.

Specifying the parameters passed to the IMS region controller

Use the parameters in column 2 of Table 37 to specify values for the DLIBATCH parameters in column 1. The values you specify are passed to the IMS region controller when an FM/IMS function is run in DLI mode. Use the parameters in column 3 to specify whether or not the user can override the values you specify.

Table 37. DLIBATCH parameters

DLIBATCH parameter	FMN1POPD/FMN1POPI parameter	Use this parameter to allow override
BKO	IMSBKO	UIMSBKO
BUF		UBUF
DBRC	DBRC	UDBRC
GSGNAME	GSGNAME	URSR
IRLM	IRLM	UIRLM
IRLMNM	IRLMNAME	UIRLM
LOCKMAX	LOCKMAX	ULOCKMAX
TMINAME	TMINAME	URSR

Use the parameters in column 2 of Table 38 to specify values for the IMSBATCH parameters in column 1. The values you specify are passed to the IMS region controller when an FM/IMS function is run in BMP mode. Use the parameters in column 3 to specify whether or not the user can override the values you specify.

Table 38. IMSBATCH parameters

IMSBATCH parameter	FMN1POPD/FMN1POPI parameter	Use this parameter to allow override
LOCKMAX	LOCKMAX	ULOCKMAX
PARDLI	PARDLI	UPARDLI
NBA	IMSNBA	UIMSNBA
OBA	IMSOBA	UIMSNBA

All the parameters in columns 2 and 3 of Table 37 and Table 38 can be specified on the FMN1POPD and FMN1POPI macro statements. For detailed information on how to code these parameters, see Appendix C.

Important Information

The PARDLI parameter specifies the parallel DL/I option that FM/IMS functions use when they are run in BMP mode.

When this parameter is not specified on the FMN1POPD and FMN1POPI macro statements, FM/IMS passes a null value to the IMS region controller and IMS uses the default value, PARDLI=0.

With PARDLI=0, a system X22 abend in the BMP region causes the IMS control region to terminate with a U0113 abend.

A system X22 abend will occur in an FM/IMS BMP region when:
- A TSO session executing an Edit or Browse in BMP mode is cancelled or times out.
- A batch function running in BMP mode is cancelled or times out.

If you want to prevent the control region being terminated with abend U0113 in the above-mentioned instances, you should set PARDLI=1. But note that this setting can degrade performance.

Controlling access to the subsystems

Use the parameters in Table 39 to control the access that FM/IMS functions have to each subsystem.

Table 39. Control options

Parameter	Description
DYNALLOC	Enforce the use of the database data sets specified in the DFSMDA dynamic allocation members when accessing the subsystem in DLI mode.
IMSAUDLG	Enforce audit logging when editing databases in the subsystem.
MAXGN	Limit the number of DL/I calls that can be issued when searching a database in the subsystem.

Table 39. Control options (continued)

Parameter	Description
PSBTYPES	Prevent dynamic PSBs or static PSBs being used to access databases in the subsystem.
READONLY	Prevent users from running update functions (Edit, Batch Edit, Load, Initialize and Delete/Define) against databases in the subsystem.
REGTYPES	Prevent users from accessing the subsystem in BMP mode or DLI mode.
VCURULE	Control the views and criteria sets that can be used with a database in the subsystem.

All the parameters in Table 39 can be specified on the FMN1POPD and FMN1POPI macro statements. For detailed information on how to code these parameters, see Appendix C.

Specifying selected processing options

Use the parameters in column 1 of Table 40 to specify the processing options that you require. Use the parameters in column 3 to specify whether or not the user can override the values you specify.

Table 40. Processing options

Parameter	Description	Use this parameter to allow override
AUTOSAVE	Whether or not the automatic save function is set on during Edit.	UAUTOSAV
CHGAFREQ	The AUTOSAVE checkpoint frequency during Change All and Repeat All operations.	UAUTOSAV
EDITFREQ	The AUTOSAVE checkpoint frequency during an Edit.	UAUTOSAV
IEBFREQ	The checkpoint frequency during a Batch Edit.	UIEBFREQ
LOADFREQ	The checkpoint frequency during a database load.	ULOADFRQ
PROCOPTB	The PSB processing option (PROCOPT) that a dynamic PSB generated for a Browse uses to read databases.	UPROCOPB
PROCOPTP	The PSB processing option (PROCOPT) that a dynamic PSB generated for a Print uses to read databases.	UPROCOPP
PROCOPTX	The PSB processing option (PROCOPT) that a dynamic PSB generated for an Extract uses to read databases.	UPROCOPX
PROCOPTY	The PSB processing option (PROCOPT) that a dynamic PSB generated for a Batch Browse uses to read databases.	UPROCOPY
PSBTYPE	The type of PSB that batch functions use to access databases.	UPSBTYPE

All the parameters in columns 1 and 3 of Table 40 except PSBTYPE and UPSBTYPE can be specified on the FMN1POPD and FMN1POPI macro statements. You can only specify PSBTYPE and UPSBTYPE on the FMN1POPD macro statement.

For detailed information on how to code these parameters, see Appendix C.

Specifying IMS subsystem and template data sets

Use the parameters in column 2 of Table 41 to specify the data sets in column 1. Where applicable, use the parameters in column 3 to specify whether or not the user can override the values you specify.

Table 41. IMS subsystem and template data sets

Data set	Parameter	Use this parameter to allow override
DBDLIB	DBDLIB*n*	UDBDLIB
DFSVSAMP data set	DFSVSAMP	UDFSVSMP
DFSVSAMP member	VSMPMEM	UDFSVSMP
DOPT ACBLIB	DYNACB	
IMS macros	MACLIB	UMACLIB
PSBLIB	PSBLIB*n*	UPSBLIB
RESLIB	RESLIB*n*	URESLIB
RECON		URECON
Staging ACBLIB	ACBLIB	UACBLIB
Template	TPLLIB*n*	UTPLLIB

All the parameters in columns 2 and 3 of Table 41 can be specified on the FMN1POPD and FMN1POPI macro statements. For detailed information on how to code these parameters, see Appendix C.

Specifying miscellaneous IMS subsystem details

Use the parameters in Table 42 to specify some miscellaneous IMS subsystem details that FM/IMS requires.

Table 42. Miscellaneous IMS subsystem details

Parameter	Description
DYNPRFN	The maximum number of dynamic PSBs required by concurrent BMP users.
DYNPRFX	The first 1- to 5-characters of the dynamic PSB names used by FM/IMS functions when run in BMP mode.
DFSRRC00	The name of the program that FM/IMS attaches to invoke IMS.
UAGNS	Whether or not the subsystem uses AGNs to secure dependent regions.
XDOPTLB	Whether or not the members of the DOPT ACBLIB data set are deleted and the DOPT ACBLIB data set is compressed before FM/IMS executes an ACBGEN.

All the parameters in Table 42 can be specified on the FMN1POPD and FMN1POPI macro statements. For detailed information on how to code these parameters, see Appendix C.

Specifying AGNS

If a subsystem uses AGNs to secure dependent regions, you can use FMN1AGNT macro statements to specify the AGNs that you want FM/IMS to use when it

accesses this subsystem in BMP mode. One FMN1AGNT macro statement is required for each AGN you want to specify. You use:

- The **AGN** parameter to specify the Application Group Name.
- The **SSID** parameter to specify the ID of the subsystem that the AGN is for.
- The **DESC** parameter to specify a description of the AGN.

The **AGN** and **SSID** parameters are required. The **DESC** parameter is optional. For detailed information on how to code these parameters, see Appendix C.

Note: Specifying AGNs for a subsystem is optional unless you want to limit the AGNs that FM/IMS functions can use when accessing this subsystem in BMP mode, or you want users to be able to select from a list of AGNs when they specify this subsystem (and a BMP region type) on the FM/IMS function's entry panel.

You must, however, specify whether or not each subsystem uses AGNs to secure dependent regions. If a subsystem uses AGNs, specify UAGNS=Y on the FMN1POPI macro statement, or specify UAGNS=Y on the FMN1POPD macro statement and do not specify the UAGNS parameter on the FMN1POPI macro statement.

If a subsystem does not use AGNs, specify UAGNS=N on the FMN1POPI macro statement, or specify UAGNS=N on the FMN1POPD macro statement and do not specify the UAGNS parameter on the FMN1POPI macro statement, or do not specify the UAGNS parameter on the FMN1POPI and FMN1POPD macro statements.

Note: If the UAGNS parameter is not specified on the FMN1POPD and FMN1POPI macro statement, FM/IMS uses the default value, UAGNS=N, and any FMN1AGNT macro statements for the subsystem included in the FMN1POPT module are ignored.

Earlier versions of FM/IMS than version 8 give a different result when the UAGNS parameter is not specified on the FMN1POPD and FMN1POPI macro statements. The default for the UAGNS parameter in earlier versions was a null value which indicated:

- The subsystem used AGNs, if FMN1AGNT macro statements for the subsystem were specified in the FMN1POPT module.
- The subsystem did not use AGNs, if no FMN1AGNT macro statements for the subsystem were specified in the FMN1POPT module.

Setting the default national language

If you installed the Japanese or the Korean components of FM/IMS, (or you provided other locally translated messages and panels), you can change the default national language, for FM/IMS batch functions. The language used by FM/IMS under ISPF depends on the language setting for your ISPF session. To change the language for FM/IMS batch functions, set the LANGUAGE parameter on the FMN0POPI macro statement, to your language. See “LANGUAGE” on page 306 for more information about the LANGUAGE parameter.

For other customization you can do for FM/IMS for national languages, refer to Chapter 25, “Customizing FM/IMS for national languages,” on page 213.

Examples of FMN1POPD, FMN1POPI and FMN1AGNT macros

The following sample code shows the FMN1POPD, FMN1POPI and FMN1AGNT macros you would code to define IMS subsystems to FM/IMS. Five subsystems are defined, of which two use AGNs. For an explanation of the statements see the notes at the end of the example. Refer to Appendix C, "FM/IMS options," on page 361 for more information about FMN1POPD, FMN1POPI and FMN1AGNT.

Where a parameter value already coded in FMN1POPD is to be used by a subsystem, that parameter is not coded in the FMN1POPI macro for that subsystem.

FMN1POPD All BMP values are protected, all DLI values are unprotected. These will be the defaults unless overridden in an FMN1POPI macro for a subsystem.

FMN1POPI SSID=IF52, un-protect all BMP values such that everything is unprotected, and use Dynamic PSBs only.

SSID=IF42, protect all DLI values such that everything is protected, and use Static PSBs only.

SSID=IF32, only use database data sets that are dynamically allocated, and use AGNs.

SSID=IFA2, BMP only, use all values from FMN1POPD except data set names.

SSID=IFB2, READONLY, all other values from FMN1POPD.

```
FMN1POPD ACBLIB=IMSV11.IFB2.ACBLIB,                          +
               AUTOSAVE=(Y,Y),                               +
               CHGAFREQ=(100,100),                           +
               DBRC=Y,                                       +
               DBDLIB1=FMN1.PROD.DBDLIB,                     +
               DBDLIB2=FMNA.PROD.DBDLIB2,                    +
               DBDLIB3=FMN2.PROD.DBDLIB,                     +
               DBDLIB4=FMN5.PROD.DBDLIB,                     +
               DBDLIB5=FMNX.TEST.DBDLIB,                     +
               DBDLIB6=FMNY.TEST.DBDLIB,                     +
               DFSVSAMP=IMSV11.IFB2.PROCLIB,                 +
               DYNACB=IMSV11.IFB2.DOPTLIB,                   +
               DYNALLOC=N,                                   +      1
               DYNPRFN=50,                                   +
               DYNPRFX=FMN0,                                 +
               EDITFREQ=(1,1),                               +
               IEBFREQ=(100,100),                            +
               IMSAUDLG=N,                                   +
               IMSBKO=Y,                                     +
               IMSNBA=10,                                    +
               IMSOBA=6,                                     +
               IRLM=N,                                       +
               IRLMNAME=,                                    +
               LOADFREQ=(100,100),                           +
               MACLIB=IMS.V10.SDFSMAC,                       +
               MAXGN=(100,100),                              +
               PARDLI=0,                                     +
               PSBLIB1=FMN1.PROD.PSBLIB,                     +
               PSBLIB2=FMNA.PROD.PSBLIB2,                    +
               PSBLIB3=FMN2.PROD.PSBLIB,                     +
               PSBLIB4=FMN5.PROD.PSBLIB,                     +
               PSBLIB5=FMNX.TEST.PSBLIB,                     +
               PSBLIB6=FMNY.TEST.PSBLIB,                     +
               PROCOPTB=(G,G),                               +
               PROCOPTP=(G,G),                               +
               PROCOPTX=(G,G),                               +
```

```
                PROCOPTY=(G,G),                                 +
                PSBTYPE=DYNAMIC,                                +
                PSBTYPES=BOTH,                                  +       2
                READONLY=N,                                     +
                REGTYPES=BOTH,                                  +       3
                RESLIB1=IMSV11.IFB2.SDFSRESL,                   +
                RESLIB2=IMSV11.IFB2.USERLIB,                    +
                TPLLIB1=FMN1.PROD.TEMPLATE,                     +
                TPLLIB2=FMNA.PROD.TEMPLATE,                     +
                TPLLIB3=FMN2.PROD.TEMPLATE,                     +
                TPLLIB4=FMN5.PROD.TEMPLATE,                     +
                TPLLIB5=FMNX.TEST.TEMPLATE,                     +
                TPLLIB6=FMNY.TEST.TEMPLATE,                     +
                UACBLIB=Y,                                      +
                UAUTOSAV=(N,Y),                                 +
                UBUF=Y,                                         +
                UDBRC=Y,                                        +
                UDFSVSMP=Y,                                     +
                UIEBFREQ=(N,Y),                                 +
                UIMSBKO=Y,                                      +
                UIMSNBA=N,                                      +
                UIRLM=Y,                                        +
                ULOADFRQ=(N,Y),                                 +
                ULOCKMAX=(N,Y),                                 +
                UMACLIB=Y,                                      +
                UPARDLI=N,                                      +
                UPROCOPB=(N,Y),                                 +
                UPROCOPP=(N,Y),                                 +
                UPROCOPX=(N,Y),                                 +
                UPROCOPY=(N,Y),                                 +
                UPSBTYPE=Y,                                     +
                URECON=Y,                                       +
                URESLIB=Y,                                      +
                URSR=Y,                                         +
                VCURULE=T,                                      +
                VSMPMEM=DFSVSMDB,                               +
                DOPTLB=N
*
FMN1POPI SSID=IF52,                                             +       IF52
                DESC="Dynamic PSB only, unprotected",           +
                ACBLIB=IMSV910.IF52.ACBLIB,                     +       7
                DBDLIB1=FMN.IF52.DBDLIB,                        +       29
                DYNACB=IMSV910.IF52.DOPTLIB,                    +       8
                IRLM=Y,                                         +       6
                IRLMNAME=IRLM,                                  +
                MACLIB=IMS.V910.SDFSMAC,                        +       9
                MAXGN=(0,0),                                    +
                PSBTYPES=DYNAMIC,                               +       4
                RESLIB1=IMSV910.IF52.SDFSRESL,                  +       10
                UAUTOSAV=Y,                                     +
                UIEBFREQ=Y,                                     +
                UIMSNBA=Y,                                      +
                ULOADFRQ=Y,                                     +
                ULOCKMAX=Y,                                     +       5
                UPARDLI=Y,                                      +
                UPROCOPB=Y,                                     +
                UPROCOPP=Y,                                     +
                UPROCOPX=Y,                                     +
                UPROCOPY=Y,                                     +
                XDOPTLB=                                        +
*
         FMN1POPI SSID=IF42,                                    +       IF42
                DESC="Static PSB only, all protected",          +
                ACBLIB=IMSV910.IF42.ACBLIB,                     +       15
                DBRC=N,                                         +       14
                DYNACB=IMSV910.IF42.DOPTLIB,                    +       16
                PSBLIB1=FMN.IF42A.PSBLIB,                       +       30
```

```
            PSBLIB2=FMN.IF42B.PSBLIB,                    +
            PSBTYPES=STATIC,                             +      11
            RESLIB1=IMSV910.IF42.SDFSRESL,               +      17
            RESLIB2=IMSV910.IF42.USERLIB,                +      18
            UACBLIB=N,                                   +      12
            UAUTOSAV=(,N),                               +      13
            UBUF=N,                                      +
            UDBRC=N,                                     +
            UDFSVSMP=N,                                  +
            UIEBFREQ=(,N),                               +
            UIMSBKO=N,                                   +
            UIRLM=N,                                     +
            ULOADFRQ=(,N),                               +
            ULOCKMAX=(,N),                               +
            UMACLIB=N,                                   +
            UPROCOPB=(,N),                               +
            UPROCOPP=(,N),                               +
            UPROCOPX=(,N),                               +
            UPROCOPY=(,N),                               +
            URECON=N,                                    +
            URESLIB=N,                                   +
            URSR=N                                       +
*
      FMN1POPI SSID=IF32,                                +        IF32
            DESC="Dynamic allocation ds only",           +
            ACBLIB=IMSV810.IF32.ACBLIB,                  +
            AUTOSAVE=N,                                  +      20
            CHGAFREQ=(44,44),                            +
            DBRC=Y,                                      +      21
            DFSVSAMP=IMSV810.IF32.PROCLIB,               +
            DYNACB=IMSV810.IF32.DOPTLIB,                 +
            DYNALLOC=Y,                                  +      19
            RESLIB1=IMSV810.IF32.SDFSRESL,               +
            RESLIB2=,                                    +
            TPLLIB1=IFA2.XXX1.TEMPLATE,                  +      31
            TPLLIB2=IFA2.XXX2.TEMPLATE,                  +
            TPLLIB3=IFA2.XXX3.TEMPLATE,                  +

            UACBLIB=N,                                   +
            UAGNS=Y,                                     +      22
            XDOPTLB=Y                                    +      23
*
      FMN1POPI SSID=IFA2,                                +        IFA2
            DESC="BMP only system",                      +
            ACBLIB=IMSV11.IFA2.ACBLIB,                   +
            DYNACB=IMSV11.IFA2.DOPTLIB,                  +
            IMSAUDLG=Y,                                  +      26
            MACLIB=IMS.V10.BETA.SDFSMAC,                 +
            REGTYPES=BMP,                                +      24
            RESLIB1=IMSV11.IFA2.SDFSRESL,                +
            RESLIB2=IMSV11.IFA2.USERLIB,                 +
            UAGNS=Y                                      +      25
*
      FMN1POPI SSID=IFB2,                                +        IFB2
            DESC="READONLY system",                      +
            READONLY=Y                                   +      27
*
      FMN1AGNT SSID=IF52,AGN=FMN52                       +      28
      FMN1AGNT SSID=IFA2,AGN=FMNA0001,DESC="HR"          +
      FMN1AGNT SSID=IFA2,AGN=FMNA0002,DESC="Finance"     +
*
      FMN1END                                                   END
```

Notes:

1. In this example, the + characters are in column 72.

2. This example shows the statements you would code to define five IMS subsystems to be used by FM/IMS. Code for an FMN1POPD macro is also shown. Application group name security is in use in two of these IMS subsystems.
3. Notice that all the FMN1POPI macros are specified together, followed by the FMN1AGNT macros.
4. An FMN1END macro statement is coded at **END** .

FMN1POPD:

1. These parameters will serve as defaults for the parameters in the FMN1POPI macros, if not specified therein. For parameters that have two values, the first is for BMP mode, the second for DLI mode.
2. As a default, all subsystems can run in BMP and DLI mode [**3**], all subsystems can run with a Static or Dynamic PSB [**2**] and dynamic allocation [**1**] is not enforced.

FMN1POPI:

IF52:

1. These statements define an IMS subsystem, called **Dynamic PSB only, unprotected**, as specified on the DESC keyword. The SSID is **IF52**. This is the value that will be shown on the FM/IMS entry panels.
2. To force the use of dynamic PSBs, the parameter PSBTYPES is set to "DYNAMIC" [**4**].
3. To unprotect all parameters that have a default value of "protect" all of the usage parameters must be set to "Y" [**5**].
4. Furthermore this subsystem will use IRLM [**6**].
5. All data set names for this subsystem are different from those that are set as default and as such need to be specified [**7** , **8** , **9** , **10**].
6. When this one parameter is included [**29**], it overrides all six DBDLIB parameters set up in FMN1POPD.
7. The UAGNS keyword is not specified, and UAGNS=NO is specified in the FMN1POPD macro; therefore **IF52** will not use application group name security. The AGNs in the first FMN1AGNT macro will be ignored.

IF42:

1. These statements define an IMS subsystem, called **Static PSB only, all protected**, as specified on the DESC keyword. The SSID is **IF42**. This is the value that will be shown on the FM/IMS entry panels.
2. To force the use of static PSBs, the parameter PSBTYPES is set to "STATIC" [**11**].
3. To protect all parameters that have a default value of "unprotect" all of the usage parameters must be set to "N" [**12**] or "(,N)" [**13**], depending on whether or not that specific parameter has one or two values.
4. Furthermore this subsystem will not use DBRC [**14**]. This is only possible if the IMS system definitions allow for it.
5. Several data set names for this subsystem are different from those that are set as default and as such need to be specified [**15** , **16** , **17** , **18**].
6. When these parameters [**15** , **16**] are included, they override all six PSBLIB parameters set up in FMN1POPD.

IF32:

1. These statements define an IMS subsystem, called **Dynamic allocation ds only**, as specified on the DESC keyword. The SSID is **IF32**. This is the value that will be shown on the FM/IMS entry panels.
2. To force the use of dynamic allocation of all database data sets in DLI mode, the parameter DYNALLOC is set to "Y" [**19**].
3. When these parameters [**31**] are included, they override all six TPLLIB parameters set up in FMN1POPD.
4. The AUTOSAVE option is turned off [**20**].
5. DBRC is set to "Yes" [**21**], but is not needed as the default value is "Yes" also.
6. This subsystem will use AGNs [**22**].
7. When dynamic PSBs are used, the DOPT ACBLIB will be cleared of all members. This will be done as XDOPTLB is set to "Y" [**23**].

IFA2:

1. These statements define an IMS subsystem, called **BMP only system**, as specified on the DESC keyword. The SSID is **IFA2**. This is the value that will be shown on the FM/IMS entry panels.
2. To force the use of BMP mode, the parameter REGTYPES is set to "BMP" [**24**].
3. This subsystem will use AGNs [**25**].
4. To force the use of audit logging, the parameter IMSAUDLG is set to "Y" [**26**].

IFB2:

1. These statements define an IMS subsystem, called **READONLY system**, as specified on the DESC keyword. The SSID is **IFB2**. This is the value that will be shown on the FM/IMS entry panels.
2. To force this system to be a read-only system, the parameter READONLY is set to "Y" [**27**].

FMN1AGNT:

1. The first statement refers to subsystem **IF52**, however, this subsystem will not use AGNs and as such this statement will be ignored [**28**].
2. The next two statements refer to subsystem **IFA2** where UAGNS is set to "Y" and it will use these two and only these two AGNs.
3. Note that there is no FMN1AGNT macro for IF32. It is specified to use AGNs. Since there is no FMN1AGNT macro for IF32 the validity of the names will be controlled by IMS only and FM/IMS will allow any name in the **AGN** field.
4. Note that the advantage of specifying the FMN1AGNT macros is be able to display a selection list if the user is not sure about the name to use. This is not possible when these macros are not included.

Tailoring the job control skeletons

Several dialogs in FM/IMS generate JCL for batch functions. If you want these dialogs to generate JCL that will run at your installation, appropriate job control must be provided. This is done by means of the installation options, the Settings panels, and a job control skeleton. The job control skeleton for FM/IMS is the member FMN1FTEX in FMN.SFMNSLIB.

FMN1FTEX

There are some DD statements in the distributed version of the skeleton that you may need to change:

1. The STEPLIB DD statement specifies DSN=&FMNSMOD1. FM/IMS sets &FMNMOD1 to either the value specified for the FMNSMOD1 parameter in the FMN1POPT module (when specified) or FMN.SFMNMOD1 (otherwise). So if you haven't installed File Manager into the default target libraries, you need to specify the name of the target File Manager load library in the FMNSMOD1 parameter or modify this DD statement.
2. The FMN1JIN DD statement specifies DSN=&FMN1SLIB. FM/IMS sets &FMN1SLIB to either the value specified for the SFMNSLIB parameter in the FMN1POPT module (when specified) or FMN.SFMNSLIB (otherwise). So if you haven't installed File Manager into the default target libraries, you need to specify the name of the target File Manager skeleton library in the SFMNSLIB parameter or modify this DD statement.
3. The STEPLIB DD statement in this skeleton also concatenates two other statements

   ```
   //*        DD DSN=IGY.SIGYCOMP,DISP=SHR
   //*FMNCOB  DD DUMMY     Uncomment to force use of FM COBOL Compiler
   ```

   which appear as comments in FMN1FTEX.

   In the first statement shown above, IGY.SIGYCOMP is the supported, licensed COBOL compiler library. Some batch functions which make use of COBOL require this library. If you did not add your supported COBOL compiler library to your LINKLIST, remove the * to uncomment the line, and change the DSN to the name of your COBOL compiler library. All currently supported versions of IBM Enterprise COBOL for z/OS and OS/390 are supported by File Manager.

   If you have created a COBOL compiler library for File Manager with a special version of IGYCDOPT, you can use this DD statement to make it available to FM/IMS for all batch jobs using COBOL templates. See “Using COBOL compiler options with File Manager” on page 15 for information about a special version of IGYCDOPT for File Manager.

   File Manager provides an internal version of the COBOL compiler, for use in preference to a supported COBOL compiler, or when a supported COBOL compiler is not available to File Manager. (See “Using the File Manager COBOL compiler” on page 16.)

You modify the FMN1FTEX job control skeleton using the usermod FMN1UMDB. FMN1UMDB is distributed in FMN.SFMNSAM1.

If you need to modify the FMN1FTEX job control skeleton, do this:

1. Copy the member FMN1FTEX from FMN.SFMNSLIB to your own source library.
2. Modify the FMN1FTEX member in your own library as required.
3. Modify the FMN1UMDB member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMN1UMDB.

Note: FM/IMS does not provide support for the automatic generation of job routing control statements in the JCL generated by FMN1FTEX.

Chapter 22. Customizing the FM/IMS security environment

This chapter describes:

- How you can control access to the FM/IMS functions and IMS subsystems.
- How you can use user-written security exit routines to control access to various IMS resources.

Controlling access to IMS subsystems and FM/IMS functions

FM/IMS allows you to control which IMS subsystems a user can access when using each of the functions listed in Table 43. These functions are protected by default when you receive FM/IMS.

Table 43. Protected FM/IMS Functions

Function code	Description	UPDATE or READONLY
DBI	Initialize dialog - generates JCL for the initialize function	UPDATE
DDD	Delete or define dialog - generates JCL to delete or define database data sets	UPDATE
DIB	Initialize - initialize databases (batch)	UPDATE
IB	Browse - browse a database	READONLY
IBBO	Batch browse dialog - generate JCL for the batch browse function	READONLY
IBB	Batch browse - read a database in batch (batch)	READONLY
IE	Edit - edit a database	UPDATE
IEBO	Batch edit dialog - generates JCL for the batch edit function	UPDATE
IEB	Batch edit - edit a database in batch (batch)	UPDATE
IPRO	Print dialog - generates JCL for the print function	READONLY
IPR	Print - print data from databases (batch)	READONLY
IX	Extract dialog - generates JCL for the extract function	READONLY
IXB	Extract - extract data from databases (batch)	READONLY
IL	Load dialog - generates JCL for the load function	UPDATE
ILB	Load - load data into databases (batch)	UPDATE

Table 43 lists the function's code (column 1), a brief description of the function (column 2), and whether the function has been classified as an update or a read-only function (column 3).

The facility allows you to grant or deny some or all users access to:

1. Individual IMS subsystems by individual functions in Table 43.
2. Individual functions in Table 43. When you grant or deny users access to individual functions, they are granted or denied access to all IMS subsystems when using these functions.
3. Individual IMS subsystems by the update or read-only functions.
4. The update or read-only functions. When you grant or deny users access to the update or read-only functions, they are granted or denied access to all IMS subsystems when using the update or read-only functions.

© Copyright IBM Corp. 2000, 2011

FM/IMS provides security for these functions, in one of two ways, either through RACF (or an equivalent security product) or through the FMNSECUR exit.

If Security Server, RACF 1.9 (or later), or an equivalent security product is active, the System Authorization Facility (SAF) with the File Manager enhanced security facility is used for access control and authorization verification. Authorization is controlled by FM/IMS-specific profiles in the FACILITY class. This chapter describes the FM/IMS-specific profiles that you must define to RACF or your equivalent security product. It also describes how you define these profiles to RACF. If you use another security product, consult the documentation for your product to determine how to define these profiles to your product.

If SAF with RACF 1.9 (or an equivalent security product), is not active when an FM/IMS function is started, the function access control checks are passed to the FMNSECUR user exit instead of to SAF.

To use FMNSECUR, it must be installed in the LPA. If the FMNSECUR module is required and it cannot be found in the LPA, an error message is issued and the FM/IMS function will not start.

FMNSECUR is a customizable exit. It provides FMNS macros which allow you to define a table of user names or job names, File Manager-protectable resources (called profiles), and access levels. For information on FMNSECUR, see "Setting up the security environment by using FMNSECUR".

Notes:

1. The FMNSECUR module is not used (even if present) if SAF with RACF 1.9 (or an equivalent security product), is active when an FM/IMS function is started.
2. FM/IMS functions that are not listed in Table 43 cannot be protected by RACF (or an equivalent security product) or by the FMNSECUR exit.

Most installations will use RACF or an equivalent security product to protect their functions. The rest of this section describes how you implement RACF (or an equivalent security product) security for the functions in Table 43.

Controlling access to the update or read-only functions

The update functions in Table 43 are protected by the profile FILEM.IMS.UPDATE. The read-only functions in this table are protected by the profile FILEM.IMS.RDONLY. As a minimum, you need to define these profiles and grant or deny users access to them.

You define these profiles in the FACILITY class. You do this by entering the following RACF commands:

```
RDEFINE FACILITY FILEM.IMS.UPDATE UACC(READ or NONE)
RDEFINE FACILITY FILEM.IMS.RDONLY UACC(READ or NONE)
```

Specify:

- UACC(READ), if you want users or groups to be granted access to these resources, unless they are specifically denied access.
- UACC(NONE), if you want users or groups to be denied access to these resources, unless they are specifically granted access.

In the following, assume that:

- The RDEFINE for the FILEM.IMS.UPDATE profile specifies UACC(NONE), so users and groups are denied access to the update functions unless they are specifically granted access.
- The RDEFINE for the FILEM.IMS.RDONLY profile specifies UACC(READ), so users and groups are granted access to the read-only functions unless they are specifically denied access.

To grant a user (with userid *userid*) or a group (with groupid *groupid*) access to the update functions, you enter one of the following RACF commands:

```
PERMIT FILEM.IMS.UPDATE CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT FILEM.IMS.UPDATE CLASS(FACILITY) ID(groupid) ACCESS(READ)
```

To deny a user (with userid *userid*) or a group (with groupid *groupid*) access to the read-only functions, you enter one of the following RACF commands:

```
PERMIT FILEM.IMS.RDONLY CLASS(FACILITY) ID(userid) ACCESS(NONE)
PERMIT FILEM.IMS.RDONLY CLASS(FACILITY) ID(groupid) ACCESS(NONE)
```

Controlling access to individual IMS subsystems by the update or read-only functions

Access to individual IMS subsystems by the update functions in Table 43 on page 191 is protected by the profiles FILEM.IMS.UPDATE.ssid, where ssid is the IMS subsystem ID. Access to individual IMS subsystems by the read-only functions in Table 43 on page 191 is protected by the profiles FILEM.IMS.RDONLY.ssid, where ssid is the IMS subsystem ID.

If you want to grant or deny some or all users access to individual IMS subsystems by the update or read-only functions in Table 43 on page 191, you will need to define the aforementioned profiles.

You define these profiles in the FACILITY class. You do this by entering the following RACF commands:

```
RDEFINE FACILITY FILEM.IMS.UPDATE.ssid UACC(READ or NONE)
RDEFINE FACILITY FILEM.IMS.RDONLY.ssid UACC(READ or NONE)
```

Specify:

- UACC(READ), if you want users or groups to be granted access to these resources, unless they are specifically denied access
- UACC(NONE), if you want users or groups to be denied access to these resources, unless they are specifically granted access.

You use the PERMIT commands to grant or deny users and groups access to these resources, in the same way as for the FILEM.IMS.UPDATE and FILEM.IMS.RDONLY profiles described in "Controlling access to the update or read-only functions" on page 192.

Controlling access to individual functions

Individual functions in Table 43 on page 191 are protected by the profiles FILEM.FUNCTION.*fc*, where *fc* is the function code. If you want to grant or deny some or all users access to individual functions in Table 43, you will need to define the profiles for these functions.

You define these profiles in the FACILITY class, by entering the following RECF command:

```
RDEFINE FACILITY FILEM.FUNCTION.fc UACC(READ or NONE)
```

Specify:

- UACC(READ), if you want users or groups to be granted access to this function, unless they are specifically denied access
- UACC(NONE), if you want users or groups to be denied access to this function, unless they are specifically granted access.

You use the PERMIT commands to grant or deny users and groups access to these function, in the same way as for the *FILEM.IMS.UPDATE and FILEM.IMS.RDONLY* profiles.

Controlling access to individual IMS subsystems by individual functions

Access to individual IMS subsystems by individual functions in Table 43 is protected by the profiles FILEM.FUNCTION.*fc*.*ssid*, where *fc* is the function code and *ssid* is the IMS subsystem ID.

If you want to grant or deny some or all users access to individual IMS subsystems by individual functions in Table 43, you will need to define the aforementioned profiles.

You define these profiles in the FACILITY class. You do this by entering the following RACF command:

```
RDEFINE FACILITY FILEM.FUNCTION.fc,ssid UACC(READ or NONE)
```

Specify:

- UACC(READ), if you want users or groups to be granted access to these resources, unless they are specifically denied access
- UACC(NONE), if you want users or groups to be denied access to these resources, unless they are specifically granted access.

You use the PERMIT commands to grant or deny users and groups access to these resources in the same way as for the FILEM.IMS.UPDATE and FILEM.IMS.RDONLY profiles described in "Controlling access to the update or read-only functions" on page 192.

What governs whether access is granted or denied

Access to a subsystem *ssid* by an *update* function in Table 43 with function code *fc* is governed by the first profile in this list

- FILEM.FUNCTION.*fc*.*ssid*
- FILEM.FUNCTION.*fc*
- FILEM.IMS.UPDATE.*ssid*
- FILEM.IMS.UPDATE

Access to a subsystem *ssid* by a *read-only* function in Table 43 with function code *fc* is governed by the first profile in this list that has been defined in the FACILITY class:

- FILEM.FUNCTION.*fc*.*ssid*
- FILEM.FUNCTION.*fc*
- FILEM.IMS.RDONLY.*ssid*
- FILEM.IMS.RDONLY

Figure 10 illustrates the security checking that FM/IMS performs when a function in Table 43 attempts to access an IMS subsystem.

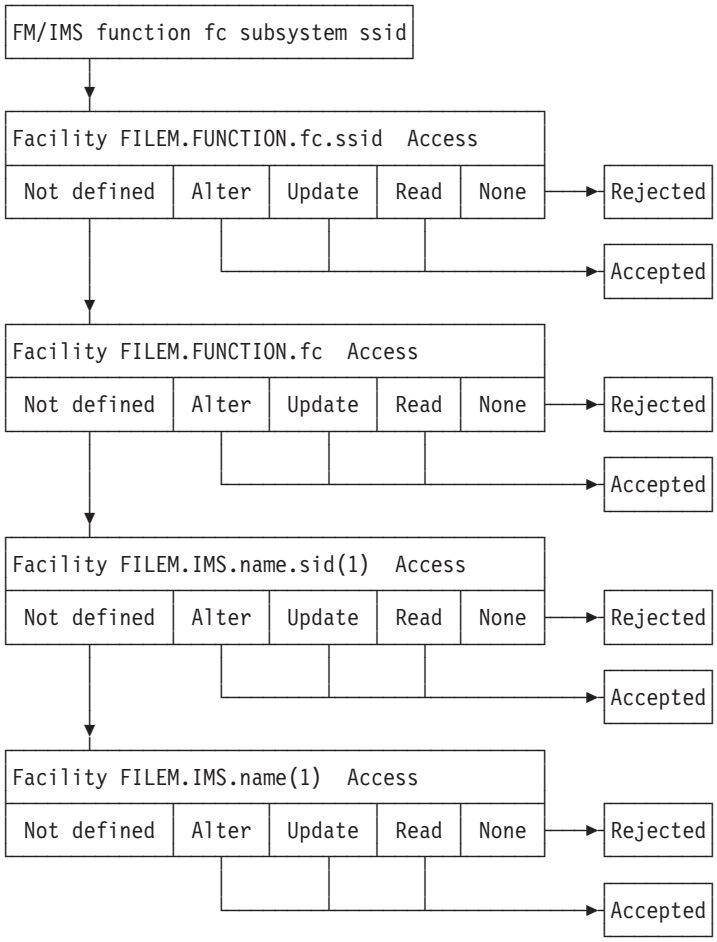


Figure 10. Security checking for FM/IMS functions

Notes:

1. FILEM.IMS.name is either FILEM.IMS.UPDATE or FILEM.IMS.RDONLY

ALTER, UPDATE or READ access means that the user can use the function. Access NONE means that the user cannot use the function.

Important information for users of non-IBM security products

FM/IMS issues a RACROUTE TYPE=AUTH for FILEM.FUNCTION.*fc*, and if this profile is not defined, RACF returns RC=4. If RC=4 is returned, FM/IMS issues a RACROUTE for either FILEM.IMS.UPDATE (if it is an update function), or FILEM.IMS.RDONLY (if it is a read-only function). If the return code from the RACROUTE TYPE=AUTH for FILEM.FUNCTION.*fc* is greater than 4, no RACROUTE is issued for FILEM.IMS.UPDATE or FILEM.IMS.RDONLY.

You should be aware that not all non-IBM security products issue RC=4 when a RACROUTE TYPE=AUTH for a profile fails; in this case no RACROUTE is issued for FILEM.IMS.UPDATE or FILEM.IMS.RDONLY, and the access request fails immediately. If this applies to your security product, you will have to provide individual profiles for all FM/IMS functions listed in Table 43 on page 191.

Customizing the FM/IMS security exit

FM/IMS provides a security exit module, FMN1SXT. FMN1SXT is called from four different points during processing.

- Exit Type A - Prior to allocating an audit trail data set when editing a database.
- Exit Type D - When allocating a database data set (DLI mode only).
- Exit Type I - Prior to invoking the IMS region controller.
- Exit Type T - After the IMS region controller terminates.

The default security program supplied returns control immediately for all exit types thus allowing normal processing to continue.

You can provide your own version of FMN1SXT in High Level Assembler or COBOL, using either of the sample source decks, FMN1XITA (HLASM) or FMN1XITC(COBOL), as a base. FMN1XITA and FMN1XITC are distributed in FMN1.SFMNSAM1.

Types of security exits

The following describes what you can use each exit type for.

Audit Trail Exit - Type A

The Audit Trail exit can be used to:

- Force the creation of an audit trail for a certain database or group of databases or for a certain user or group of users, or both.
- Override the standard audit trail data set name that is constructed by FM/IMS.
- Force the use of System Management Facilities (SMF) recording for the audit trail instead of using an audit trail data set. If you want to use SMF recording, the SMF record ID to be used must have been specified in the options macro, FMN1POPT.
- This exit is called only for the Edit function.

Database Data set Allocation Exit - Type D

The database data set allocation exit is called when the function runs in DLI mode. For example, this exit can be used to:

- Control access to database data sets.
- Override the database data set allocation status from OLD to SHR during database edit if IMS data sharing is used at your installation.

IMS Initialization Exit - Type I

The IMS Initialization exit can be used to:

- Control access to databases.
- Validate the value entered by the user for the IMS log data set. You can either accept, override, or disallow the value entered by the user.
- Override the standard FM/IMS log data set naming conventions.
- Override the Profile Option MAXGN.

IMS Termination Exit - Type T

The IMS Termination exit can be used to perform post-IMS processing of the log data set. This exit type has no parameters, it is provided as a point where you can add your own REXX code for termination processing

Invoking the security exit

The security exit program is invoked as follows:

```
CALL FMN1SXT (FMN_SECURITY_PARAMETERS, FMN_SECURITY_WORKAREA, FMN_IMS_SECURITY_PARAMETERS)
```

The security exit interface parameters are:

- The security parameter list passed to the security exit. This parameter list is documented in the following pages.
- A security area work area which is a 256-byte area initialized to binary zeroes. This area is unchanged by FM/IMS and can be used to pass information between multiple calls to the security exit.
- IMS security parameters which are passed to the IMS Initialization and Termination exits.

Common Exit Parameters

Table 44. Common Parameters - All Exit Types

Field	Update	Size	Description
Request Type	N	Char(1)	**A** Audit Trail allocation **D** Database data set allocation **I** IMS Initialization **T** IMS Termination
Option	N	Char(1)	**B** Browse **E** Edit **L** Extract/Load **P** Batch printing **U** Utilities
User Id	N	Char(7)	TSO userid
Permitted	Y	Char(1)	**Y** Allow intended action **N** Disallow intended action **Blank** N/A Not used for Exit Type A.

Security Exit Parameters

Table 45. Parameters - Exit Type A.

Field	Update	Size	Description
DB DSN	N	Char(44)	This field is obsolete and no longer populated.
DSN	Y	Char(44)	The Audit Trail DSN to be allocated.
SMF Record Id	Y	Binary(16)	Default is 00 (SMF not used for audit trail). Choose a number between 128 and 255. Only applies if the 'Create Indicator' (below) is set to Y.
Create	Y	Char(1)	Input value from the Edit Entry Panel. Override the value with one of the following: **Y** Create an audit trail. **N** Do not create an audit trail. **D** See IMSAUDLG=D description.
Keep	Y	Char(1)	**Y** Keep the audit trail data set. **N** Delete the audit trail data set after printing.
Report	Y	Char(1)	**Y** Produce the Audit Trail report at the end of the edit session. **N** Do not produce the Audit Trail report at the end of the edit session.
Job Type	N	Char(1)	**B** DLI mode **M** BMP mode
DBD DSN	N	Char(44)	The DBD library that has been allocated.
DBD Name	N	Char(8)	The DBD being processed.
IMS System Id	N	Char(4)	The IMS SYSTEM ID entered.
Appl. Group Name	N	Char(8)	The AGN used.
PSB Type	N	Char(1)	**S** Static **D** Dynamic
PSB Name	N	Char(8)	The name of the Program Specification Block (PSB).
PSB DSN	N	Char(44)	The name of the PSB library data set (static DLI).

Table 46. Parameters - Exit Type D.

Field	Update	Size	Description
Primary DBD Name	N	Char(8)	The primary DBD being processed.
Physical DBD Name	N	Char(8)	The physical DBD that has the DATASET= statement for this DDNAME.
DDNAME	N	Char(8)	The DDNAME of the database being allocated.
DSN Status	Y	Char(3)	SHR or OLD. Default is SHR for Browse and OLD for Edit.
DSN	N	Char(44)	The DSN of the database.

Table 47. Parameters - Exit Type I

Field	Update	Size	Description
DBD DSN	N	Char(44)	The DBD library that has been allocated.
DBD Name	N	Char(8)	The DBD being processed.
Subfunction	N	Char(1)	Option subfunction. Function L, Extract/Load: **E** Extract **L** Load Function U, Utilities: **D** Delete/define database data sets **I** Initialize IMS databases
IMS Log Indicator	Y	Char(1)	The user entered IMS log indicator. Only applicable during the online portion of the Edit and Load functions. Values are: **K** Allocate and keep log data set. **D** Allocate and delete log data set. **N** Do not use a log data set
IEFRDER DSN	Y	Char(44)	The IMS log data set name as constructed by FM/IMS when the 'IMS log data set' option has been selected in the Edit and Load online functions.
Job Type	N	Char(1)	**B** DLI mode **M** BMP mode
IMS System Id	N	Char(4)	The IMS SYSTEM ID entered.
Appl. Group Name	N	Char(8)	The AGN used.
Processing Option	N	Char(1)	The Database Load Processing Option as specified by the user. **1** Update and insert segments. **2** Insert new segments only.
Time	N	Char(6)	The time the exit call is made. Format is HHMMSS.
Date	N	Char(8)	The date the exit call is made. Format is YYYYMMDD.
PSB Type	N	Char(1)	**S** Static **D** Dynamic
PSB Name	N	Char(8)	The name of the Program Specification Block (PSB).
PSB DSN	N	Char(44)	The name of the PSB library data set (static DLI).
MAXGN Value	Y	Binary(16)	This value is only applicable during the Edit and Browse functions. The maximum number of GN (Get Next) calls allowed to satisfy a FIND or CHANGE command. Used to override the Profile Option MAXGN value.

Sample programs for a security exit

Copybooks, sample program source and JCL for the security exit are supplied for High Level Assembler and COBOL. The sample program source and JCL are distributed in FMN.SFMNSAM1, and the copybooks in FMN.SFMNMAC1. They are:

FMN1AXIT HLASM copybook for security exit parameters.
FMN1XITA Sample HLASM code for program FMN1SXT.
FMN1UMDS Usermod to install an HLASM version of FMN1SXT.
FMN1CXIT COBOL copybook for security exit parameters.
FMN1XITC Sample COBOL code for program FMN1SXT.
FMN1SECC Job control to install a COBOL version of FMN1SXT.

To provide your version of FMN1SXT in HLASM you use the usermod FMN1UMDS as follows:

1. Copy the member FMN1XITA from FMN.SFMNSAM1 to your own source library.
2. Code your version of FMN1XITA in your source library, using FMN1XITA from FMN.SFMNSAM1 as a base.
3. Modify the usermod FMN1UMDS member in FMN.SFMNSAM1 to meet your requirements. Refer to the usermod for information about changes you might need to make.
4. Install SMP/E usermod FMN1UMDS.

To provide your version of FMN1SXT in COBOL:

1. Copy the member FMN1XITC from FMN.SFMNSAM1 to your own source library.
2. Code your version of FMN1XITC in your source library, using FMN1XITC from FMN.SFMNSAM1 as a base.
3. Modify the sample job FMN1SECC in FMN.SFMNSAM1 to meet your site's requirements. Refer to the sample job for information about any changes you might need to make.
4. Run the job to FMN1SECC to compile and link your version of FMN1SXT. This job will link FMN1SXT into FMN.SFMNMOD1.

To implement your exit, add FMN.SFMNMOD1 to your LINKLIST or to the STEPLIB DD statement in your TSO logon procedure.

Note: If the security exit program is written in COBOL, the performance of the application may be impacted.

Chapter 23. Customizing the FM/IMS audit facility (FMN1POPT controlled auditing)

You can use the audit logging facilities of FM/IMS to create an audit trail of edit activity on IMS databases. You can create your audit trail either using SMF or recording the activity to an audit log data set.

FM/IMS determines whether or not an audit trail is produced by the setting of the `Create audit trail` option on the Edit Entry Panel (option 2 from the FM/IMS Primary Option Menu). The setting of the SMFNO option in FMN1POPT determines if the audit trail is written to SMF.

You can further customize how FM/IMS records an audit trail by means of the Audit Trail exit in the FM/IMS security exit, FMN1SXT. For example, using this exit you can:

- Force the creation of an audit trail for a certain database or group of databases or for a certain user or group of users, or both.
- Override the standard audit trail data set name that is constructed by FM/IMS.
- Override `Create audit trail` as set by your users.
- Force the use of SMF recording for the audit trail instead of using an audit trail data set.
- Force audit logging during Edit and at the conclusion of the Edit session submit an audit report job to report on the changes. The job submitted will be determined by skeleton member FMN1FTAD found in FMN.SFMNSLIB. You should customize the job card and JCL to specify the reporting options you require. For changing the audit report options in the skeleton see "AUD (Print Audit Trail Report)" in the *File Manager User's Guide and Reference*. For more information, see "IMSAUDLG" on page 371.

Note: You do not use the AUDITLOG option in FMN0POPT to determine whether or not to produce an audit trail in FM/IMS. The AUDITLOG option has no effect in FM/IMS.

Using System Management Facilities (SMF) for audit logging

If you plan to use SMF for audit logging, you must do the following:

- Ensure that the load module FMNSMF is APF-authorized. For more information about using SMF for audit logging, and making FMNSMF APF-authorized, see page 49.
- Add the load module FMNSMF to the AUTHTSF list in member IKJTSO*xx* in SYS1.PARMLIB. If you do not do this, even if you have selected to record to SMF and you have specified an SMF record type, no recording will be done.
- Select an SMF record number for the audit log records, and include it in your SMF parmlib member SMFPRM*xx*.
- Specify this SMF record number in the options module, FMN1POPT. For information on how to do this see "Customizing the FM/IMS installation options module" on page 177.

© Copyright IBM Corp. 2000, 2011

To report on the audit trail information collected by SMF, you must extract this information from SMF to your own data set. The information in this data set can then be printed by the FM/IMS Print Audit Trail utility. To do this select option 3.8 from the FM/IMS Primary Option Menu.

A sample job, FMNSMFX, is provided in FMN.SFMNSAM1 to help you extract the SMF data to your own data set. See the comments in the job for information about the changes you need to make to the job. For more information about SMF, see *z/OS MVS System Management Facilities (SMF)*.

Recording an audit trail to a data set

If you choose to record your audit trail to a data set, you can print this information using the FM/IMS Print Audit Trail utility. To do this select option 3.8 from the FM/IMS Primary Option Menu.

The format of the audit log data set name is determined by the setting of the AUDITHLQ parameter in the FMN0POPI definition in FMN1POPT. See "AUDITHLQ" on page 291 for more information about the AUDITHLQ option.

The following data set name formats may be generated:

`userid.IMSAUDIT.Dyymmdd.Thhmmss`	(when AUDITHLQ= (blank))
`audithlq``.IMSAUDIT.Dyymmdd.Thhmmss`	(when AUDITHLQ=*`audithlq`*)
`qual1.<qual2.><qual3.>Dyymmdd.Thhmmss`	(when AUDITHLQ=*`qual1.<qual2.><qual3>`*)

where:

audithlq Is any 1-8 character constant that is valid in the context of a data set name

userid Is the userid creating the data set

Dyymmdd Is the date of the activity

Thhmmss Is the time of the activity

When AUDITHLQ contains one or more periods, the AUDITHLQ value is treated as a data set prefix, with one, two or three levels. Each level of the prefix can be:

XXX Any 1-8 character constant that is valid in the context of a data set name.

&&PREFIX Indicates that the user's TSO prefix should be used. This will be null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix will also be null.

&&USER Indicates that the user's logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&UID Indicates that the user's TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user's TSO logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&FUNCOD Indicates that the File Manager internal function code should be used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.

&&SSID Indicates that the currently connected IMS subsystem name should be used.

Activating your changes to SYS1.PARMLIB

To activate any changes you have made to SYS1.PARMLIB members, either restart your system, or use the appropriate commands for your site to dynamically activate the changes.

Chapter 24. Customizing the FM/IMS audit facility (SAF-rule controlled auditing)

You can control FM/IMS audit logging in one of two ways. You can use the System Authorization Facility (SAF), or you can use the FM/IMS installation options module, FMN1POPT. In this chapter, we describe how you can use SAF to control FM/IMS audit logging.

For a comparison of these two methods of controlling FM/IMS audit logging, see "Deciding how to customize the FM/IMS audit facility" on page 171.

For information on how to implement FMN1POPT controlled audit logging, see Chapter 23, "Customizing the FM/IMS audit facility (FMN1POPT controlled auditing)," on page 201.

Implementing SAF-rule controlled auditing

Use the checklist shown here to implement SAF-rule controlled auditing:

__ 1. Determine the FACILITY and XFACILIT class profiles required to control FM/IMS audit logging at your installation. The information provided in this section should help you to do that:
   - "Understanding how SAF controls FM/IMS audit logging" on page 206.

__ 2. Determine which level of access to these profiles your users will require to meet the audit logging requirements at your installation. The information provided in these sections should help you to do that:
   - "Controlling where FM/IMS writes audit log records" on page 207.
   - "Controlling whether or not an audit trail is created" on page 209.
   - "Controlling if users can request an audit trail when one is not required (Edit function only)" on page 210.

__ 3. Define the required profiles to RACF or your equivalent security product and provide users with the required access to these profiles. The information provided in these sections describes how you do this when using RACF:
   - "FILEM.AUDIT1.*ssid*.TOSMF" on page 208.
   - "FILEM.AUDIT1.*ssid*.TODSN" on page 208.
   - "FILEM.AUDIT1.*ssid*.*fc*.*db*." on page 209.
   - "FILEM.AUDIT1.*ssid*.OPTION" on page 210.

   If you use another security product, consult the documentation for your product.

__ 4. Activate SAF-rule controlled auditing for the user IDs that you intend to use in step 5. The information provided in these sections describe how you do that:
   - "Specifying that FM/IMS audit logging is to be controlled by SAF" on page 210.
   - "FM/IMS options specified in FMN1PARM" on page 417.

__ 5. Test your configuration to ensure that audit logging occurs when, and only when, it is required.

__ 6. When you are satisfied that FM/IMS audit logging is only occurring when it should, activate SAF controlled audit logging for all FM/IMS users. The information provided in these sections describe how you do that:
   - "Specifying that FM/IMS audit logging is to be controlled by SAF" on page 210.

© Copyright IBM Corp. 2000, 2011

• "FM/IMS options specified in FMN1PARM" on page 417.

Understanding how SAF controls FM/IMS audit logging

Table 48 lists the FM/IMS functions that can be made to create an audit trail when audit logging is controlled by SAF.

Table 48. FM/IMS functions that support audit logging when audit logging is controlled by SAF

Function code	Function name	Description
IB	Browse	Browse a database
IBB	Batch Browse	Read a database in batch
IE	Edit	Edit a database
IEB	Batch Edit	Edit a database in batch
ILB	Load	Load data into databases (batch)
IPR	Print	Print data from a database (batch)
IXB	Extract	Extract data from databases (batch)

The following describes how SAF controls FM/IMS audit logging.

When functions in Table 48 are started, SAF is invoked to answer these audit queries:

- Whether or not audit logging is required.
- If audit logging is required, whether audit records should be written to SMF, the user's audit log data set, or both.
- For Edit and Browse function only, whether or not the user's audit log data set should be printed at the end of the Edit/Browse session.
- For Edit function only, whether or not the **Create audit trail** option on the Edit Entry panel can be used:
  - To request an audit trail when one is not required.
  - To stop an audit trail being created when one is required.

The responses to these queries are controlled by FACILITY and XFACILIT class profiles that you define.

Table 49 lists the profiles that control the responses when the query is from a given FM/IMS function that is being used to access a given database in a given IMS subsystem, where:

- *ssid* is the IMS subsystem ID.
- *fc* is the function code.
- *db* is the database name.

Table 49 lists the profile name (column 1), the class in which the profile must be defined (column 2), and what the profile controls (column 3).

Table 49. SAF profiles that control FM/IMS audit logging

SAF profile	Class	Description
FILEM.AUDIT1.*ssid*.TOSMF	FACILITY	Controls whether or not audit log records are written to SMF.

Table 49. SAF profiles that control FM/IMS audit logging (continued)

SAF profile	Class	Description
FILEM.AUDIT1.*ssid*.TODSN	FACILITY	1. Controls whether or not audit log records are written to the user's audit log data set. 2. Controls whether or not the user's audit log data set is printed at the end of the session (for Edit and Browse function only).
FILEM.AUDIT1.*ssid*.OPTION	FACILITY	Controls whether or not the **Create audit trail** option on the Edit Entry panel can be used to request an audit trail when one is not required (Edit function only).
FILEM.AUDIT1.*ssid.fc.db*	XFACILIT	1. Controls whether or not audit logging is required. 2. Controls whether or not the **Create audit trail** option on the Edit Entry panel can be used to stop an audit trail being created when one is required (Edit function only).

The following sections describe how you use these profiles to control FM/IMS audit logging and how you define these profiles to RACF.

Controlling where FM/IMS writes audit log records

When a function in Table 48 on page 206 accesses a given IMS subsystem, two profiles control where FM/IMS writes audit log records. These profiles are FILEM.AUDIT1.*ssid*.TOSMF and FILEM.AUDIT1.*ssid*.TODSN, where *ssid* is the IMS subsystem ID. Each of these profiles are discussed in turn.

FILEM.AUDIT1.*ssid*.TOSMF
: This profile controls whether any audit log records that this function may create are written to SMF. If the user has no access to this profile, then the function does not write audit log records to SMF. If the user has READ (or higher) access to this profile, then any audit log records that this function may create are written to SMF.

FILEM.AUDIT1.ssid.TODSN
: This profile controls whether any audit log records that this function may create are written to the user's audit log data set. For the Edit and Browse function, it also controls whether or not the user's audit log data set is printed at the end of the Edit/Browse session. If the user has no access to this profile, then the function does not write audit log records to the user's audit log data set. If the user has READ (or higher) access to this profile, then any audit log records that this function may create are written to the user's audit log data set. If the user has UPDATE (or higher) access to this profile, and it is the Edit or Browse function, and the function created an audit trail, then the user's audit log data set is printed at the end of the Edit/Browse session.

If the user does not have access to either of these profiles, then FM/IMS does not create an audit trail under any circumstances. So, apart from controlling where the audit log is written, these profiles can also affect whether or not an audit trail is created.

The following describes how you define these profiles to RACF.

FILEM.AUDIT1.*ssid*.TOSMF

You define this profile in the FACILITY class. You do this by entering this RACF command:

```
RDEFINE FACILITY FILEM.AUDIT1.ssid.TOSMF UACC(READ or NONE)
```

`UACC(READ)`
: Specify this if you want audit log records from users and groups to be written to SMF, unless they are specifically denied access to this profile.

`UACC(NONE)`
: Specify this if you do not want audit log records from users and groups to be written to SMF, unless they are specifically granted access to this profile.

In the following, assume that the RDEFINE for the profile specifies UACC(NONE), so audit log records from users and groups are not written to SMF unless they are specifically granted access.

To specify that the audit log records from a user (with userid *userid*) or a group (with groupid *groupid*) are to be written to SMF, you enter one of these RACF commands:

```
PERMIT FILEM.AUDIT1.ssid.TOSMF CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT FILEM.AUDIT1.ssid.TOSMF CLASS(FACILITY) ID(groupid) ACCESS(READ)
```

FILEM.AUDIT1.*ssid*.TODSN

You define this profile in the FACILITY class. You do this by entering this RACF command:

```
RDEFINE FACILITY FILEM.AUDIT1.ssid.TODSN UACC(UPDATE or READ or NONE)
```

`UACC(UPDATE)`
: Specify this if you want audit log records from users and groups to be written to their audit log data sets and, if it is an Edit or Browse, you want the audit log data sets printed at the end of the Edit/Browse session, unless they are specifically given some other access to this profile.

`UACC(READ)`
: Specify this if you want audit log records from users and groups to be written to their audit log data sets, unless they are specifically given some other access to this profile.

`UACC(NONE)`
: Specify this if you do not want audit log records from users and groups to be written to their audit log data set, unless they are specifically granted access to this profile.

In the following, assume that the RDEFINE for the profile specifies UACC(READ), so audit log records from users and groups are written to their audit log data set unless they are specifically denied access.

To specify that the audit log records from a user (with userid *userid*) or a group (with groupid *groupid*) are not to be written to their audit log data set, you enter one of these RACF commands:

```
PERMIT FILEM.AUDIT1.ssid.TODSN CLASS(FACILITY) ID(userid) ACCESS(NONE)
PERMIT FILEM.AUDIT1.ssid.TODSN CLASS(FACILITY) ID(groupid) ACCESS(NONE)
```

To specify that Browse and Edit audit log data sets from a user (with userid *userid*) or a group (with groupid *groupid*) are to be printed at the end of the Edit/Browse session, you enter one of these RACF commands:

```
PERMIT FILEM.AUDIT1.ssid.TODSN CLASS(FACILITY) ID(userid) ACCESS(UPDATE)
PERMIT FILEM.AUDIT1.ssid.TODSN CLASS(FACILITY) ID(groupid) ACCESS(UPDATE)
```

Controlling whether or not an audit trail is created

When a function in Table 48 on page 206 accesses a given database in a given IMS subsystem, three profiles control whether or not FM/IMS creates an audit trail. These profiles are:

FILEM.AUDIT1.*ssid*.TOSMF
FILEM.AUDIT1.*ssid*.TODSN
FILEM.AUDIT1.*ssid.fc.db*

where *ssid* is the IMS subsystem ID, *fc* is the function code, and *db* is the database name.

The FILEM.AUDIT1.*ssid*.TOSMF and FILEM.AUDIT1.ssid.TODSN profiles are discussed in "Controlling where FM/IMS writes audit log records" on page 207 where it mentions that if the user does not have access to either of these profiles, then no audit log is created under any circumstances. In the following, we assume that the user *does* have READ (or higher) access to one or both of these profiles.

The FILEM.AUDIT1.*ssid.fc.db* profile controls whether the function creates an audit trail.

If the user has no access to this profile, then the function does not create an audit trail. If the user has READ (or higher) access to this profile, then the function does create an audit trail. If the user has CONTROL (or higher) access to this profile and it is the Edit function, then the function creates an audit trail if, and only if, the user has selected the **Create audit trail** option on the Edit Entry panel.

The following describes how you define this profile to RACF.

FILEM.AUDIT1.*ssid.fc.db*.

You define this profile in the XFACILIT class. You do this by entering this RACF command:

```
RDEFINE XFACILIT FILEM.AUDIT1.ssid.fc.db UACC(READ or NONE)
```

`UACC(READ)`
: Specify this if you want audit trails created for the users and groups, unless they are specifically denied access to this profile.

`UACC(NONE)`
: Specify this if you do not want audit trails created for the users and groups, unless they are specifically granted access to this profile.

In the following, assume that the RDEFINE for the profile specifies UACC(NONE), so audit trails are not created for the users and groups, unless they are specifically granted access.

To specify that audit trails are to be created for a user (with userid *userid*) or a group (with groupid *groupid*), you enter one of these RACF commands:

```
PERMIT FILEM.AUDIT1.ssid.fc.db CLASS(XFACILIT) ID(userid) ACCESS(READ)
PERMIT FILEM.AUDIT1.ssid.fc.db CLASS(XFACILIT) ID(groupid) ACCESS(READ)
```

Controlling if users can request an audit trail when one is not required (Edit function only)

When the FM/IMS Edit function accesses a given IMS subsystem, the FILEM.AUDIT1.*ssid*.OPTION profile controls whether or not the **Create audit trail** option on the Edit Entry panel can be used to request an audit trail when one is not required.

If the user has no access to this profile, then the **Create audit trail** option cannot be used to request an audit trail when one is not required.

If the user has READ (or higher) access to this profile, then the **Create audit trail** option can be used to request an audit trail when one is not required, but only when the user has READ (or higher) access to either the FILEM.AUDIT1.*ssid*.TOSMF or the FILEM.AUDIT1.*ssid*.TODSN profiles. If the user does not have access to either of these profiles, then no audit log is created under any circumstances.

The following describes how you define this profile to RACF.

FILEM.AUDIT1.*ssid*.OPTION

You define this profile in the FACILITY class. You do this by entering this RACF command:

```
RDEFINE FACILITY FILEM.AUDIT1.ssid.OPTION UACC(READ or NONE)
```

`UACC(READ)`
: Specify this if you want to allow users and groups to use the **Create audit trail** option to request an audit trail when one is not required. unless they are specifically denied access to this profile.

`UACC(NONE)`
: Specify this if you don't want to allow users and groups to use the **Create audit trail** option to request an audit trail, unless they are specifically granted access to this profile.

Specifying that FM/IMS audit logging is to be controlled by SAF

For users wanting to control auditing by means of parmlib, the users being audited must have read access or more to the SAF facility class profile FILEM.PARMLIB.IMS. In this case, auditing is controlled by the FMAUDIT parameter in the FMN1PARM member. For SAF-rule controlled auditing, specify SAF_CTRL=YES. For more information, see "FM/IMS options specified in FMN1PARM" on page 417.

Note: File Manager for IMS will not start if a user has read access to this facility class and the FMN1PARM member is unavailable from parmlib.

Example: User PROD1 to have auditing controlled from parmlib

Write this RACF rule:

```
RDEF FACILITY FILEM.PARMLIB.IMS AUDIT(NONE) UACC(NONE)
    OWNER(ownerid) PE FILEM.PARMLIB.IMS ACC(READ) ID(PROD1)
    CLASS(FACILITY)
```

If SAF processing is not active, or the rule is not defined, or the rule is defined and the user has no access, then no parmlib processing is performed.

If there is no parmlib processing to specify SAF-controlled auditing, then SAF-controlled auditing is performed for users that have read access to the SAF facility class profile FILEM.SAFAUDIT.IMS.

Example: User PROD2 to have SAF auditing without parmlib

Write this RACF rule:

```
RDEF FACILITY FILEM.SAFAUDIT.IMS AUDIT(NONE) UACC(NONE)
    OWNER(ownerid) PE FILEM.SAFAUDIT.IMS ACC(READ) ID(PROD2)
    CLASS(FACILITY)
```

Note: If SAF-controlled auditing is specified without parmlib processing, then the SMF record type is obtained from options module SMFNO value.

When a security product other than RACF is in use

File Manager V11 APAR PM31842 introduced a change to remove the mandatory requirement for definition of members in the logical PARMLIB concatenation. In order to achieve this, File Manager makes RACROUTE calls, with STATUS=ACCESS, to SAF FACILITY profiles:

FILEM.PARMLIB.**
FILEM.SAFAUDIT.** (only if the first call fails)

STATUS=ACCESS is specified because the RACROUTE call should not audit the call. The documentation in the *z/OS V1R11 Security Server RACROUTE Macro Reference*, in the description of RACROUTE REQUEST=AUTH,STATUS=ACCESS, states:

ACCESS
The request is simply to return the user's highest current access to the resource specified. Upon successful completion, the user's access is returned in the RACF reason code. **No auditing is done for this request.**

When RACF is used, the STATUS=ACCESS request works as documented, and no security-related logging or abends are generated, even if you do not have access to the profile.

However, when non-RACF security products (such as ACF2) are used, S047 ABEND047 may be issued in response to the above RACROUTE request.

The ACF2 global system options (GSO) definitions shown here have been used to avoid the S047 abends:

```
SET CONTROL(GSO)
INSERT SAFDEF.APFFM PROGRAM(FMN*****) RB(FMN*****) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=FACILITY,STATUS=ACCESS)
INSERT SAFDEF.APFFM2 PROGRAM(FILEM***) RB(FILEM***) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=FACILITY,STATUS=ACCESS)
F ACF2,REFRESH(SAFDEF)
INSERT SAFDEF.APFFM PROGRAM(FMN*****) RB(FMN*****) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
INSERT SAFDEF.APFFM2 PROGRAM(FILEM***) RB(FILEM***) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
F ACF2,REFRESH(SAFDEF)
```

Chapter 25. Customizing FM/IMS for national languages

You can customize FM/IMS for national languages other than English. If you are using Japanese or Korean, and have installed the FM/IMS Japanese or Korean components, you might not need to perform any other customization for the Japanese or Korean national languages.

If you are using a language other than English, Japanese or Korean (for example, French), you will need to perform the customization tasks listed in Table 50. Perform **ALL** these steps if you plan to use a different language than English, Japanese or Korean.

Table 50. Summary of steps for customizing FM/IMS for a national language

Step	Description	Page
__ 1	Set the LANGUAGE option for batch processing, if required.	184
__ 2	Change the TERMTYPE option in FMN1POPT, if you are using a DBCS language.	322
__ 3	Confirm that you have the correct terminal type set in ISPF (ISPF option 0).	
__ 4	Create a print and display translation table for your language.	213
__ 5	Translate the File Manager message text to your language.	214
__ 6	Provide a version of FMN1MENU for your language.	214
__ 7	Translate the FM/IMS ISPF messages to your language.	214
__ 8	Translate the FM/IMS panels to your language.	215

Changing the print and display translation tables for languages other than English

DBCS languages
This step is essential if you are using a DBCS language other than Japanese or Korean.

If you plan to use FM/IMS with a national language other than English, you may need to provide a print and display translation table for your language. You do this as part of your customization for File Manager base function. See “Changing the print and display translation tables for languages other than English” on page 61.

You should also specify PRTTRANS=ON in FMN1POPT. If you are using any DBCS language you might also need to specify TERMTYPE=3270KN in FMN1POPT.

© Copyright IBM Corp. 2000, 2011

Translating the message text

All FM/IMS messages are stored in the FMN1MENU source member. This CSECT is part of the root module so that an English version of the messages is always available. In addition, all messages used by FM/IMS under ISPF are provided in the library, FMN.SFMNMENU. Using both FMN1MENU and members in FMN.SFMNMENU, you can provide your own set of translated messages. To use the messages you have translated, see "Using the translated messages and panels" on page 215.

To provide translated versions of the messages, you must provide a version of FMN1MENU in your language, as described in "Providing a national language version of FMN1MENU" **and** provide translated versions of the appropriate members in FMN.SFMNMENU, as described in "Translating the ISPF messages text." (You should not need to provide a Japanese or a Korean version of the messages if you have installed the FM/IMS Japanese or Korean components.)

Providing a national language version of FMN1MENU

FMN1MENU contains the assembler source for the FM/IMS messages. To provide translated versions of the messages:

1. Copy the member FMN1MENU from FMN.SFMNSAM1 to your own source library with the name FMN1M*yyy*, where *yyy* is one of the following language codes:

FRA	French
DEU	German
ITA	Italian
JPN	Japanese
PTG	Portuguese
ESP	Spanish
DAN	Danish
ENP	Upper case English
KOR	Korean
DES	Swiss German
CHT	Traditional Chinese
CHS	Simplified Chinese
XXX	Other

2. Change the message text in FMN1M*yyy* in your library.
3. Modify the FMN1UMDM member in FMN.SFMNSAM1 to meet your site's requirements, using the same language code as above. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN1UMDM.

Translating the ISPF messages text

All FM/IMS ISPF messages are provided in English. They are also provided in Japanese if you have installed the FM/IMS Japanese component, and in Korean if you have installed the FM/IMS Korean component. You can translate some or all of these messages into another language.

All FM/IMS ISPF messages are stored in FMN.SFMNMENU. You translate a message as follows:

1. Find the members in FMN.SFMNMENU that contain the messages you want to translate. The message members specific to FM/IMS are all named FMNI*zzzz*.

2. Create a library with the same characteristics as FMN.SFMNMENU, with the name FMN.SFMNM*yyy*, where *yyy* is the same language code you specified when you modified FMN1MENU. If you have already created a library with this name for translated File Manager base function messages, use that library. Copy the required message members from FMN.SFMNMENU to this library.
3. Change the required message texts in these members in your library.

To use the messages you have translated, see "Using the translated messages and panels." For more information about defining and using ISPF messages, see *z/OS ISPF Dialog Developer's Guide*.

Note: Be sure to include **ALL** the messages in the message members you copy to your own library. When FM/IMS needs to display an ISPF message, it uses ISPF services to do this. Therefore the search for a message is made according to ISPF rules. Thus, if ISPF finds the message member it requires in your library, but the message number required is not in that member, ISPF will not look in any other library for the message, but will give an error. However, if you omit a complete message member from your library, then ISPF will use the English message member from the next library in the ISPMLIB concatenation. For more information about defining and using ISPF messages, see *z/OS ISPF Dialog Developer's Guide*.

Translating the panel text

All FM/IMS panels are provided in English. They are also provided in Japanese if you have installed the FM/IMS Japanese component and in Korean if you have installed the FM/IMS Korean component. You can translate some or all of these panels into another language. (If no translated version of a particular panel is available, FM/IMS uses the English version.)

All FM/IMS panels are stored in FMN.SFMNPENU. You translate a panel as follows:

1. Find the panel members in FMN.SFMNPENU that you want to translate. The panel members specific to FM/IMS are all named FMN1*zzzz*.
2. Create a library with the same characteristics as FMN.SFMNPENU, with the name FMN.SFMNP*yyy*, where *yyy* is the same language code you specified when you modified FMN1MENU. If you have already created a library with this name for translated File Manager base function panels, use that library. Copy the required panel members from FMN.SFMNPENU to this library.
3. Change the required panel text in the members in your library. A panel may reference a **help** panel by means of a `.HELP` statement. If any panel you are changing contains any of these `.HELP` statements, also copy and change these referenced members in your library.

To use the panels you have translated, see "Using the translated messages and panels."

Using the translated messages and panels

To use your translated messages in a batch job, specify the appropriate language with the LANGUAGE processing option, using the keywords shown in Table 51 on page 216. See "Customizing the FM/IMS installation options module" on page 177 for information on how to do this.

Table 51. Keyword values for the LANGUAGE option

Language	Code	Specify on LANGUAGE option...
French	FRA	FRENCH
German	DEU	GERMAN
Italian	ITA	ITALIAN
Japanese	JPN	JAPANESE
Portuguese	PTG	PORTUGUESE
Spanish	ESP	SPANISH
Danish	DAN	DANISH
Upper case English	ENP	UPPERENG
Korean	KOR	KOREAN
Swiss German	DES	SGERMAN
Traditional Chinese	CHT	CHINESET
Simplified Chinese	CHS	CHINESES
Other	XXX	OTHER

For example, to use French messages, specify LANGUAGE=FRENCH.

Under ISPF, the language used in messages and panels is determined by the setting of the national language for ISPF, for the current ISPF session. For information on changing the national language setting for your ISPF session, see *z/OS ISPF Dialog Developer's Guide*.

If your ISPF session is set up to use your language, you need to add your libraries to the appropriate ISPF concatenation, in front of any File Manager English libraries. For example, to use your translated messages, add FMN.SFMNM*yyy* to ISPMLIB, in front of FMN.SFMNMENU. To use your translated panels, add FMN.SFMNP*yyy* to ISPPLIB, in front of FMN.SFMNPENU.

Customizing for the Japanese national language

The other customization task you might need to do for the Japanese national language is to modify the supplied Japanese translation tables. If you want to do this, do so as part of the customization of the File Manager base function. See "Modifying the Japanese translation tables" on page 65.

Changing the Japanese message text

If you have installed the FM/IMS Japanese component, all FM/IMS Japanese messages are stored in the FMN1MJPN source member. Normally you should not need to modify this module. However, if you do want to modify it, you can do so by means of the usermod, FMN1UMDN.

To do this:

1. Copy the member FMN1MJPN from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN1MJPN in your library.
3. Modify the FMN1UMDN member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.

4. Install SMP/E usermod FMN1UMDN.

Customizing for the Korean national language

The other customization task you might need to do for the Korean national language is to modify the supplied Korean translation tables. If you want to do this, do so as part of the customization of the File Manager base function. See "Modifying the Korean translation tables" on page 66.

Changing the Korean message text

If you have installed the FM/IMS Korean component, all FM/IMS Korean messages are stored in the FMN1MKOR source member. Normally you should not need to modify this module. However, if you do want to modify it, you can do so by means of the usermod, FMN1UMDQ.

To do this:

1. Copy the member FMN1MKOR from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN1MKOR in your library.
3. Modify the FMN1UMDQ member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN1UMDQ.

Changing the Korean message text

Chapter 26. Verifying the customization of FM/IMS

After you have completed all the necessary steps involved in the initial installation and customization of FM/IMS, you need to perform the following steps to verify your installation, before completing the installation process. You might need to refer to the *File Manager User's Guide and Reference for IMS Data*.

Step 1. Build the sample IMS databases to be used during verification

Member FMN1IVP in FMN.SFMNSAM1 contains a sample job that builds sample IMS databases. These databases are used in the installation verification process. Instructions in the sample job describe the modifications you must make to the JCL before you submit it

1. Following the instructions in the sample job, make the required modifications to the FMN1IVP job.
2. Run the FMN1IVP job. The expected return code is zero.

Member FMN1IVP1 in FMN.SFMNSAM1 contains a sample job that copies the sample database segment layouts to your COBOL and PL/I COPYLIB data sets. Instructions in the sample job describe the modifications you must make to the JCL before you submit it.

3. Following the instructions in the sample job, make the required changes to the FMN1IVP1 job.
4. Run the FMN1IVP1 job. The expected return code is zero.

Note
View and Template data sets are empty.

Step 2. Start FM/IMS

If you have added an option for FM/IMS to your ISPF Primary Options menu (see "Adding FM/IMS to your ISPF menu" on page 173), enter the option value you have assigned to FM/IMS and press `Enter`. For example, if you have assigned 'FI' to FM/IMS, enter 'FI' and press `Enter`.

If you defined FM/IMS in an ISPF command table (see "Defining FM/IMS in an ISPF command table" on page 173) verify that FM/IMS can be started by entering the command FI on any ISPF command line.

The FM/IMS Primary Option Menu should be displayed.

© Copyright IBM Corp. 2000, 2011

```
 Process   Options   Help
_____________________________________________________________________________
FM/IMS                       Primary Option Menu

0  Settings     Set processing options                User ID . : PERTHWA
1  Browse       Browse data                           System ID : FMD2
2  Edit         Edit data                             Appl ID . : FMN1
3  Utilities    Perform utility functions             Version . : 11.1.0
4  Templates    Template/view/criteria set utilities  Terminal. : 3278
X  Exit         Terminate FM/IMS                      Screen. . : 1
                                                       Date. . . : 2010/11/03
                                                       Time. . . : 16:25
Command ===> _______________________________________________________________
```

Figure 11. Primary Option Menu

Note: The first time you use FM/IMS, a Copyright panel is displayed. After reading the panel text, press `Enter`. In subsequent FM/IMS sessions, this panel will not automatically appear.

Enter `VER` on the command line to display the release level and PTF level of FM/IMS. A panel is displayed, similar to this:

```
IBM File Manager for z/OS Version 11 Release 1
IMS Component
(not APF authorized)

Service Levels of installed components

             Base       IMS        DB2        CICS
English      -NONE-     -NONE-     -NONE-     -NONE-
```

Notes:

1. This will always show FM/IMS as `not APF authorized`, even if you have made File Manager APF-authorized, as File Manager cannot run APF-authorized under ISPF.
2. When you first install File Manager, `-NONE-` will be shown against each component. Subsequently, when you have applied service to File Manager, a PTF number will be shown, indicating the PTF level of each component. If you have not installed a component, that component will not be shown at all. If you have installed the Japanese or Korean language components of FM/IMS, another line will be displayed indicating the service level of that component.

Step 3. Display settings menu

On the Primary Option Menu, enter **0** on the command line, for Settings, and press `Enter`.

The Settings Menu (Figure 12) is displayed.

```
 Process   Options   Help
_____________________________________________________________________________
FM/IMS                          Settings Menu

1  Print      Print settings
2  System     System settings
3  Batch      Job card specifications
4  Compiler   Language and compiler specifications
5  EDIT       Editor options
6  Subsystem  IMS subsystem settings
7  Temporary  Temporary Data Set Allocations
8  Output     Output Data Set Allocations
9  Trace      Trace options

Command ===> _____________________________________________________________
```

Figure 12. Settings Menu

Step 4. Display DLI mode settings menu

Enter **6** on the command line for Subsystem settings, and press `Enter`. The Subsystem Settings Menu (Figure 13) is displayed.

```
 Process   Options   Help
_____________________________________________________________________________
FM/IMS                       Subsystem Settings Menu

DLI mode settings
1  Parameters   Parameters passed to the IMS region controller
2  Data sets 1  DFSVSAMP, RESLIB and IMS macros data set names
3  Data sets 2  RECON and ACBLIB data set names
4  Options      Autosave, checkpoint frequencies and PROCOPTs

BMP mode settings
5  Parameters   Parameters passed to the IMS region controller
6  Options      Autosave, checkpoint frequencies and PROCOPTs

DLI and BMP mode settings
7  Data sets 1  PSB and DBD data set names
8  Data sets 2  Template data set names

IMS Subsystem Name   ____

Command ===> _____________________________________________________________
```

Figure 13. Subsystem Settings Menu

Enter **1** on the command line, for Parameters, and leave the **IMS Subsystem Name** field blank, and press `Enter`. The DLI Mode Parameters : Subsystem Selection panel (Figure 14) is displayed.

Step 5. Verify subsystem details

```
 Process   Options   Help
_______________________________________________________________________________
FMN1PSSL    DLI Mode Parameters : Subsystem Selection

                     Read  PSB    Region  AGNs
Cmd  SSID  Status    Only  Types  Types   Used  Description
___  IFA2  ACTIVE     N    BOTH   BMP      N    IFA2 - BMP only
___  IFB2  INACTIVE   Y    BOTH   BOTH     N    IFB2 - READONLY
___  IF92  ACTIVE     N    BOTH   BOTH     Y    IF92 - UAGNS=Y and 3 AGNs
___  IF82  ACTIVE     N    BOTH   BOTH     Y    IF82 - UAGNS=Y and no AGNs
___  IF72  ACTIVE     N    BOTH   BOTH     N    IF72 - UAGNS=N, READONLY=N
___  IF62  ACTIVE     N    BOTH   BOTH     Y    IF62 - UAGNS not spec, 6 AGNs
___  IF52  ACTIVE     N    DYN    BOTH     N    IF52 - DYNAMIC, all unprotect
___  IF42  ACTIVE     N    STAT   BOTH     N    IF42 - STATIC, all protected
___  IF32  ACTIVE     N    BOTH   BOTH     N    IF32 - DYNALLOC=Y
___  IF22  INACTIVE   N    BOTH   BOTH     N    IF22 - IMS V7
___  IF12  ACTIVE     Y    BOTH   BOTH     N    IF12 - UAGNS=N, READONLY=Y
    ****  End of data  ****

Command ===> _____________________________________________________________
```

Figure 14. DLI Mode Parameters

Check that all the subsystems that you specified in the FM/IMS installation options module are listed, and that the correct details are listed for each.

Select the subsystem that you are using for the verification process and press Enter. The DLI Mode Parameters panel (Figure 15) is displayed.

Step 6. Verify DLI mode parameters

```
 Process   Options   Help
_______________________________________________________________________________
 FM/IMS                        DLI Mode Parameters

 Subsystem IF52  IF52 - DYNAMIC, all unprotect

 Options:
    Enter "/" to select option
    /  Dynamic backout

    DBRC                          IRLM
    2  1. Use DBRC                2  1. Use IRLM
       2. Do not use DBRC            2. Do not use IRLM
       3. Subsystem default          3. Subsystem default

 Parameters:
    IRLMNM  . . . IRLM
    GSGNAME . . .
    TMINAME . . .
    BUF . . . . .
    LOCKMAX . . . 555

Command ===> _____________________________________________________________
```

Figure 15. DLI Mode Parameters

Enter **RESET** on the command line and press Enter.

Check that the settings displayed for the subsystem are the settings you specified in the FM/IMS installation options module.

Press **PF3** to save the settings in your profile and exit the panel. The Subsystem Settings Menu is again displayed.

Step 7. Verify DLI mode data sets

Enter **2** on the command line, for Data Sets 1, and press Enter. The DLI Mode Data Sets 1 panel (Figure 16) is displayed.

```
 Process   Options   Help
_______________________________________________________________________________
FM/IMS                       DLI Mode Data Sets 1

Subsystem IF52  IF52 - DYNAMIC, all unprotect

DFSVSAMP:
   Data set name . . 'IMSV11.IFB2.PROCLIB'
   Member  . . . . . DFSVSMDB

RESLIB:
   Data set name #1  'IMSV910.IF52.SDFSRESL'
   Data set name #2  'IMSV910.IF52.USERLIB'
   Data set name #3
   Data set name #4
   Data set name #5
   Data set name #6

IMS Macros:
   Data set name . . 'IMS.V910.SDFSMAC'

Command ===> _____________________________________________________________
```

Figure 16. DLI Mode Data Sets 1

Enter **RESET** on the command line and press Enter.

Check that the data set names displayed for the subsystem are the data set names you specified in the FM/IMS installation options module.

Press **PF3** to save the data set names in your profile and exit the panel. The Subsystem Settings Menu (Figure 13 on page 221) is redisplayed

Enter **3** on the command line, for Data Sets 2, and press Enter. The DLI Mode Data Sets 2 is displayed.

```
 Process   Options   Help
_______________________________________________________________________________
FM/IMS                       DLI Mode Data Sets 2

Subsystem IF52  IF52 - DYNAMIC, all unprotect

RECON:
   Primary data set  'IMSV910.IF52.RECON1'
   Secondary . . . . 'IMSV910.IF52.RECON2'
   Spare . . . . . . 'IMSV910.IF52.RECON3'

ACBLIB:
   Data set name . . 'IMSV910.IF52.ACBLIB'

Command ===> _____________________________________________________________
```

Figure 17. DLI Mode Data Sets 2

Enter **RESET** on the command line and press Enter.

Check that the RECON data set name fields are blank. Also check that the ACBLIB data set name displayed for the subsystem is the data set name you specified in the FM/IMS installation options module.

Press **PF3** to save the settings in your profile and exit the panel. The Subsystem Settings Menu is again displayed.

Step 8. Verify DLI mode options

Enter **4** on the command line, for Options, and press `Enter`. The DLI Mode Options panel (Figure 18) is displayed.

```
 Process   Options   Help
_______________________________________________________________________________
FM/IMS                          DLI Mode Options

Subsystem IF52  IF52 - DYNAMIC, all unprotect

Options:
   Enter "/" to select option
   _  Autosave

Checkpoint Frequencies:
   Edit . . . . . . . . . 5
   Change All/Repeat All  55
   Load . . . . . . . . . 155
   Batch Edit . . . . . . 255

PSB Processing Options:
   Browse         Extract        Print          Batch Browse
   1  1. G        1  1. G        2  1. G        2  1. G
      2. GO          2. GO          2. GO          2. GO


Command ===> _____________________________________________________________
```

Figure 18. DLI Mode Options

Enter **RESET** on the command line and press `Enter`.

Check that the settings displayed for the subsystem are the settings you specified in the FM/IMS installation options module.

Press **PF3** to save the settings in your profile and exit the panel. The Subsystem Settings Menu is again displayed.

Step 9. Verify DBD library

Enter **7** on the command line, for Data sets 1 PSB and DBD data set names, and press `Enter`. The PSB and DBD Data Sets panel (Figure 19) is displayed.

```
 Process   Options   Help
_______________________________________________________________________________
FM/IMS                       PSB and DBD Data Sets

Subsystem IF52    IF52 - DYNAMIC, all unprotect

PSB:
   Data set name #1  'FMN.PSBLIB'_____________________________________
   Data set name #2  _________________________________________________
   Data set name #3  _________________________________________________
   Data set name #4  _________________________________________________
   Data set name #5  _________________________________________________
   Data set name #6  _________________________________________________

DBD:
   Data set name #1  'FMN.DBDLIB'_____________________________________
   Data set name #2  _________________________________________________
   Data set name #3  _________________________________________________
   Data set name #4  _________________________________________________
   Data set name #5  _________________________________________________
   Data set name #6  _________________________________________________

Command ===> _________________________________________________________________
```

Figure 19. PSB and DBD Data Sets panel

Press PF3 to save the settings in your profile and exit the panel. The Subsystem Settings Menu is again displayed.

Press PF3 again to return to the Settings Menu (Figure 12 on page 221).

Press **PF3** again to return to the FM/IMS Primary Option Menu (Figure 11 on page 220).

Step 10. Start the browse dialog

Enter **1** on the command line, for Browse data, and press `Enter`. The Browse Entry Panel (Dynamic PSB) (Figure 20) is displayed.

```
 Process   Options   Help
_______________________________________________________________________________
FM/IMS                        Browse Entry Panel

IMS:
   Subsystem name  . . IF52          PSB name  . . . ________  (If static PSB)
   Database name . . . ________      AGN name  . . . ________  (If BMP)
View:
   Data set name . . . ___________________________________________
   Member  . . . . . . ________

Processing Options:
   PSB type            Region type         Fetch DB dsnames from (if DLI)
   1  1. Dynamic       1  1. DLI           1  1. User profile
      2. Static           2. BMP              2. DFSMDA members

   View usage                              Enter "/" to select option
   3  1. New                               _  Secondary index (if dynamic PSB)
      2. Existing                          _  Skip DB data set panel (if DLI)
      3. None

Command ===> _________________________________________________________________
```

Figure 20. Browse Entry Panel (Dynamic PSB)

The subsystem name you selected in Step 5. is displayed in the IMS **Subsystem name** field.

Start the browse dialog

Enter **DJ1E** in the **Database name** field.

Enter **1** in the **PSB type** field, **1** in the **Region type** field, **2** in the **Fetch DB dsnames from (if DLI)** field, a space in the **Secondary index (if dynamic PSB)** field, **3**in the **View usage** field, and press `Enter`.

Either the Browse : Database Data Set Specification panel or the Browse : Database Data Set Display panel is displayed. (The Browse Database Data Set Display panel is displayed if the FMN1POPI macro statement for the subsystem specifies DYNALLOC=Y. Otherwise, the Browse Database Data Set Specification panel is displayed.)

Figure 21 shows the Browse : Database Data Set Specification panel. Figure 22 on page 227 shows the Browse : Database Data Set Display panel.

Step 11. Verify database data sets

```
 Process   Options   Help
_____________________________________________________________________________
FM/IMS             Browse : Database Data Set Specification

Subsystem IF52  Database DJ1E

DBD name  DD name   Data set name
DJ1E      DJ1E      'FMIMS.DB.DJ1E'
DJ2E      DJ2E      'FMIMS.DB.DJ2E'
DJ2F      DJ2F      'FMIMS.DB.DJ2F'
DJ3E      DJ3E      'FMIMS.DB.DJ3E'
DJ3F      DJ3F      'FMIMS.DB.DJ3F'
DJ1F      DJ1F      'FMIMS.DB.DJ1F'
****  End of data  ****


Processing Options:
   Fetch dsnames from            Enter "/" to select option
   1  1. Profile                  _  Save dsnames in profile
      2. Dynamic Allocation data set


Command ===> ____________________________________________________________________
```

Figure 21. Browse : Database Data Set Specification

```
 Process   Options   Help
_____________________________________________________________________________
FMN1PE3H           Browse : Database Data Set Display

Subsystem IF32  Database DJ1E

DBD name  DD name   Data set name
DJ1E      DJ1E      'FMIMS.XTEST.DB.DJ1E'
DJ2E      DJ2E      'FMIMS.XTEST.DB.DJ2E'
DJ2F      DJ2F      'FMIMS.XTEST.DB.DJ2F'
DJ3E      DJ3E      'FMIMS.XTEST.DB.DJ3E'
DJ3F      DJ3F      'FMIMS.XTEST.DB.DJ3F'
DJ1F      DJ1F      'FMIMS.XTEST.DB.DJ1F'
****  End of data  ****




Command ===> _____________________________________________________________
```

Figure 22. Browse : Database Data Set Display

Check that the data set names displayed on the panels are the names given to the database data sets in the FMN1IVP job. If necessary, change the data set names to those shown in Figure 21 on page 226 and Figure 22. Then press `Enter`. The Browse : Database Positioning panel (Figure 23) is displayed.

Step 12. Database positioning

```
 Process   Options   Help
_____________________________________________________________________________
FM/IMS                 Browse : Database Positioning

Subsystem IF52  Database DJ1E      Key sequence                    Format CHAR
View      None                                 Key
Cmd SXE Level     Segment   Description     len  Key value
___ SX  1         SHIRE                     20  BROOME
___  X   2        SHIRENP                    0
___  X   2        LINKSUB                   18  ..................
   ****  End of data  ****


Command ===> _____________________________________________________________
```

Figure 23. Browse : Database Positioning

Enter an **S** in the **Cmd** field for the SHIRE segment and press `Enter`. The Browse : IMS Database panel (Figure 24) is displayed.

Step 13. Browse the database

```
Process   Options   Help
_____________________________________________________________________________
FM/IMS                   Browse : IMS Database DJ1E
                                                Scope DB  Col 1       Format CHAR
Cmd Level Segment   ----+----1----+----2----+----3----+----4----+----5---
                    ****   Top of window   ****
___ 1     SHIRE     BROOME              100100998.......p.......k......f.
___  2    SHIRENP   ..U........
___  2    LINKSUB   BROOME
___  2    LINKSUB   BROOME SHIRE
___ 1     SHIRE     DENMARK             130500244.h.....g/..............
___  2    SHIRENP   ...........
___  2    LINKSUB   BOW BRIDGE
___  2    LINKSUB   DENMARK
___  2    LINKSUB   DENMARK SHIRE
___  2    LINKSUB   GOLDEN HILL
___  2    LINKSUB   HAZELVALE
___  2    LINKSUB   KENT RIVER
___  2    LINKSUB   KENTDALE
___  2    LINKSUB   KENTON
___  2    LINKSUB   KORDABUP
___  2    LINKSUB   MOUNT SHADFORTH
___  2    LINKSUB   PARRY INLET


Command ===> _____________________________________________________________
```

Figure 24. Browse : IMS Database

This concludes the verification process.

Enter **=x** on the command line to return you to the z/OS Primary Option Menu. Alternatively, press PF3 repeatedly until you return to this panel. In the process, the message:

```
DFS627I IMS RTM CLEANUP ( EOT ) COMPLETE FOR ST PERTHWA .ISPFPROC.        ,RC=00
```

is displayed. Ensure that RC=00 is displayed. This message verifies that the connection between IMS and your ISPF session has terminated normally.

Complete the installation

You can now complete the installation of FM/IMS by performing the ACCEPT processing. The steps involved are described in the *File Manager Program Directory*.

Part 4. Customizing File Manager CICS Component

© Copyright IBM Corp. 2000, 2011

Chapter 27. Preparing to customize FM/CICS

FM/CICS runs by means of a CICS transaction, either by submitting a batch job or communicating with the FM Server, which communicates with the CICS transaction via TCP/IP. There are a number of steps you need to perform to customize and facilitate this process.

Before you can install and customize FM/CICS, you must have installed the File Manager base function. This and the following chapters assume you have installed FM/CICS into the same target and distribution libraries as File Manager base function, with one exception: FM/CICS load modules are installed by default into FMN.SFMNMOD**2**, whereas File Manager base function and the other File Manager component load modules are installed into FMN.SFMNMOD**1**.

Apart from the load modules, you could encounter problems using File Manager if you do not install FM/CICS in the same libraries as the base function.

Before you can use FM/CICS you must customize FM/CICS and the operating environment, as described in this section. You must also install, and optionally customize, IBM Problem Determination Tools Common Components, as described in Chapter 36, "Customizing IBM Problem Determination Tools Common Components for use with FM/CICS," on page 281.

The FM/CICS interface allows you to process CICS resources through a CICS terminal session. You can also optionally invoke the File Manager base function, FM/DB2, and FM/IMS interfaces through your FM/CICS session.

If you plan to use the FM Server method for the CICS FM transaction, you must also complete the customization of the FM Server, as described in Appendix E, "Customizing the FM Server," on page 435.

If you plan to run the File Manager base function interface through the FM/CICS session, you must first complete the installation and customization of the File Manager base function, as described in Part 1, "Customizing File Manager," on page 1.

If you plan to run the FM/IMS interface through the FM/CICS session, you must first complete the installation and customization of FM/IMS as described in Part 3, "Customizing File Manager IMS Component," on page 161.

If you plan to run the FM/DB2 interface through the FM/CICS session, you must first complete the installation and customization of FM/DB2 as described in Part 2, "Customizing File Manager DB2 Component," on page 97.

Table 52 on page 232 lists the customization tasks you can perform for FM/CICS. You should read the referenced sections in the table to determine the customization you need to do.

Table 53 on page 235 lists IBM publications that are referred to in this section, or which you might find useful.

© Copyright IBM Corp. 2000, 2011

Checklist for installing and customizing FM/CICS

Table 52. Summary of steps for customizing FM/CICS and the operating environment

	Description	Page
__ 1	Add FMN3PARM to SYS1.PARMLIB - **required**.	232
__ 2	Update your CICS startup procedures - **required**.	237
__ 3	Provide OMVS segments for the CICS region userid and other users - **required**.	237
__ 4	Customize TCP/IP, if necessary.	238
__ 5	Customize FM Server, if necessary.	435
__ 6	Customize to run base FM, FM/IMS and FM/DB2 through FM/CICS, if necessary.	239
__ 7	Customize for FM/CICS on interconnected CICS regions.	239
__ 8	Customize the FM/CICS batch procedure - **required**.	241
__ 9	Modify and run the installation job, FMN3INST - **required**.	242
__ 10	Customize to process COBOL copybooks.	245
__ 11	Customize to process PL/I include books.	245
__ 12	Customize to process HLASM copybooks.	245
__ 13	Customize to use DFSORT.	245
__ 14	Change the default options.	246
__ 15	Change the print and display tables.	250
__ 16	Customize to use an I/O exit.	250
__ 17	Customize the FM/CICS security environment.	251
__ 18	Customize CICS security for FM/CICS.	253
__ 19	Determine how to customize the FM/CICS audit facility.	234
__ 20	Customize FM/CICS for national languages.	269

Defining the FMN3PARM member

If auditing is to be controlled from parmlib (user has read access to FILEM.PARMLIB.IMS, see "Controlling FM/CICS auditing using SAF" on page 259), then member FMN1PARM must be defined in SYS1.PARMLIB (or any other library in the logical parmlib concatenation) as follows.

Default parmlib member FMN3PARM is provided in the SFMNSAM1 library. Copy this member to the appropriate system parmlib library. See below for details of methods that can be used to make this change.

Note: The sample FMN3PARM member supplied in SFMNSAM1 also includes an FMSECRTY statement. This option is not used in FM/CICS Version 11, and can be either omitted, or commented out. It has no effect.

FM/CICS will not start if the required parmlib member does not exist in the system logical parmlib concatenation for users to be audited by means of parmlib.

There are two methods that can be used to include the FMN3PARM member in a library in the logical parmlib concatenation. The choice of method depends on

whether the installation's security software is configured to allow FM/CICS users READ access to the data set SYS1.PARMLIB.

Method 1 can only be used when FM/CICS users have read access to SYS1.PARMLIB.

Method 2 can be used regardless of whether FM/CICS users have READ access to SYS1.PARMLIB or not, and must be used when FM/CICS users do not have READ access to SYS1.PARMLIB.

Method 1

Place the FMN3PARM member in any library in the current logical parmlib concatenation. No IPL or other action is required to activate the new member (unless a new library was added to the logical parmlib concatenation).

Notes:

1. Method 1 cannot be used in any situation where FM/CICS users do not have READ access to SYS1.PARMLIB. For example, when FM/CICS users have READ access to another library in the logical parmlib concatenation, and the FMN3PARM member is placed in the latter library. This will not work. The key issue is whether the FM/CICS user has READ access to SYS1.PARMLIB.
2. Using this method results in message IEE252I being written to the system log whenever a FM/CICS user accesses SYS1.PARMLIB. These messages cannot be suppressed. To avoid these messages use Method 2.

Method 2

This method must be used when FM/CICS users do not have READ access to SYS1.PARMLIB, or when suppression of the IEE252I messages is required.

1. Create a new library with dataset attributes similar to SYS1.PARMLIB.
2. The library name for this data set must include the string "FMNPARM" in one of the qualifiers. You can choose any data set name that meets this requirement. Examples of suitable data set names are:
   SYS1.PARMLIB.FMNPARM SYS8.FMNPARM.PARMLIB
   FMNPARM.SYS8.PARMLIB SYS2.FMNPARMS.LIB
   SYS8.XFMNPARM.PARMLIB
3. Add member FMN3PARM to the new library, specifying the appropriate FMAUDIT parameter.
4. Add the new library to the logical parmlib concatenation. This can be done dynamically, or by means of a system IPL.

Note: When Method 2 is used, the FMN3PARM member must be located in the library created in Step 1 (as described above). In addition, if the FMN3PARM member specifies any include statements (see “Facilities for customizing the FMN3PARM definitions” on page 430), all the included members must also reside in the library created in Step 1.

You use the FMN3PARM member to define:

- Whether FM/CICS uses SAF to control FM/CICS audit logging.
- The SAF resource name prefix to be used by FM/CICS when determining access to various resources.
- Whether FM/CICS loads the FMN3POPT module from a specific library.

For more information, see “Deciding how to customize the FM/CICS audit facility.”

Deciding how to customize the FM/CICS audit facility

You have two choices with respect to auditing of FM/CICS audit activities:

FMN3POPT controlled auditing
The facilities available with FMN3POPT controlled auditing are that you can specify auditing to the user's audit log data set, to the user's audit log data set with automatic (mandatory) printing of the audit log at the completion of the session, or to SMF.

SAF-rule controlled auditing
This relies on various SAF FACILITY and XFACILIT resource rules which you define with an external security product, such as RACF (or equivalent product).

These points summarize the facilities available with SAF-rule controlled auditing:

- Auditing can be (optionally) specified for all FM/CICS functions
- Different auditing requirements can be specified for different TSO userids.
- Different auditing requirements can be specified for access to different resources.
- You can provide FM/CICS users with a "Create audit trail" option for the FM/CICS edit functions. This is also SAF-rule controlled. The presence of the "Create audit trail" option does not guarantee that the user can switch off auditing, since this depends on the level of access the user has to the appropriate SAF resource names. When a user has access to the "Create audit trail" option, they can always turn on auditing, even if the relevant SAF resource rules do not require auditing.
- You can specify auditing to the user's audit log data set, to the user's audit log data set with automatic (mandatory) printing of the audit log at the completion of the session, or to SMF. Dual logging (to the user's audit log data set and to SMF) can also be specified.

Some other points to consider are:

- Auditing to the user's audit log data set can result in large numbers of audit log data sets. This may have disk space implications. You may need to consider implementing automatic purging or archiving of audit log data sets.
- Auditing to SMF (only) requires additional set-up, but provides a more reliable and secure environment for capturing audit information than audit logging to the user's audit log data set.

If you determine that SAF-rule controlled auditing is not appropriate, follow the instructions in Chapter 32, “Customizing the FM/CICS audit facility (FMN3POPT controlled auditing),” on page 255.

If you determine that SAF-rule controlled auditing is appropriate, follow the instructions in Chapter 33, “Customizing the FM/CICS audit facility (SAF-rule controlled auditing),” on page 259.

Publications

Table 53. Useful publications

Publication Title
IP CICS Sockets Guide
CICS Transaction Server for z/OS CICS Resource Definition Guide
CICS Transaction Server for z/OS Installation Guide
CICS Transaction Server for z/OS CICS Customization Guide
CICS Transaction Server for z/OS RACF Security Guide

Deciding how to customize the FM/CICS audit facility

Chapter 28. Customizing the operating environment for FM/CICS

This chapter describes how to customize the operating environment for FM/CICS. You do this after you have installed FM/CICS.

CICS resource definitions for FM/CICS

You will have to define resources to your CICS system before you can use FM/CICS. You do this by means of the job FMN3INST, distributed in FMN.SFMNSAM1. See "Modifying and submitting FMN3INST" on page 242 for information about FMN3INST and how to customize it.

Updating the CICS startup procedures

You need to make the following changes to your CICS startup procedures, before you can use FM/CICS.

- Add the group **FMNCICS** to a group list that is installed at CICS startup.
- Add the FM/CICS load library to the DFHRPL DD in your CICS startup procedure. This will be FMN.SFMNMOD2 if you installed FM/CICS into the default libraries.

  If you have installed the FM/CICS Japanese or Korean components, add the FM/CICS Japanese or Korean load library to the DFHRPL DD in front of FMN.SFMNMOD2. The default Japanese load library is FMN.SFMNMODJ and the default Korean load library is FMN.SFMNMODK.
- Add the following statement for the definition of the internal reader for job submission, to your startup procedure:

  ```
  //FMNRDR   DD   SYSOUT=(,INTRDR)
  ```

  If you change the name of the **FMNRDR** resource on the DEFINE TDQUEUE(FMNJ) statement in FMN3INST, then modify this statement in your startup procedure accordingly. See "Modifying and submitting FMN3INST" on page 242 for information about FMN3INST.

  You might also have to modify your CICS startup procedures for TCP/IP customization. See "Customizing for CICS TCP/IP" on page 238 for more information.

You should also ensure that your CICS system is set up to run with SEC=YES. Without this option the FM/CICS logon panel will not be able to verify or change passwords. For information about FM/CICS security, see Chapter 31, "CICS security and FM/CICS," on page 253.

When you have done all this, restart the CICS region.

Providing OMVS segments for FM/CICS users

The CICS region userid, and also each individual user of FM/CICS, must have a userid defined to z/OS with an OMVS segment defined. This allows usage of TCP/IP by FM/CICS. If an OMVS segment is not defined, message FMNCA021 may be issued with the `INITAPI` function when a user attempts to run the FM/CICS transaction.

© Copyright IBM Corp. 2000, 2011

Customizing for CICS TCP/IP

FM/CICS communicates with the CICS transaction via TCP/IP. Therefore you must set up and configure CICS TCP/IP if you have not already done so for another application. The following is an outline of the steps required to do this. For detailed information, see the *IP CICS Sockets Guide*.

1. Modify the startup JCL for the CICS region to include the DD statements required for TCP/IP. The required DD statements are listed in "Modifying CICS startup" in the *IP CICS Sockets Guide*.
2. Define the required TCP/IP resources to CICS. Refer to the member EZACICCT in the SEZAINST TCP/IP target library, or the RDO definitions listed in "Defining CICS TCP/IP Resources" in the *IP CICS Sockets Guide* for a complete list of all required resource definitions.
3. Optionally add the IP CICS entries to the PLT to automatically startup or shutdown the CICS Sockets Interface during CICS startup and shutdown. Refer to "CICS Program List Table (PLT)" in the *IP CICS Sockets Guide* for a description of the IP CICS PLT entries.

   Note: If the IP CICS entries are not added to the PLT, then the CICS Sockets interface must be started manually through the EZAO,START,CICS transaction after every time the CICS region is restarted.
4. Record the value of the TCPIPJOBNAME parameter specified in the TCPIP.DATA data set. This is the name of the started procedure used to start the TCP/IP Services address space and is required for the next step.
5. Create the TCP/IP configuration file, and then configure it using the EZACICD macro. A complete description and sample JCL for creating and building the configuration file can be found in "Configuring the CICS TCP/IP Environment" in the IP CICS Sockets Guide.
6. Restart the CICS region.
7. If you did not add the IP CICS entries to the PLT, start the CICS sockets interface by issuing the EZAO,START,CICS transaction (if available).

IPv6 support

The FM/CICS transaction communicates with either the FM Server, or a batch job via TCP/IP sockets and supports both IPv4 and IPv6 networks. If TCP/IP on the system where the CICS region is running supports Internet Protocol version 6 (IPv6), then the following additional requirements exist:

- TCP/IP and the resolver on the system where the CICS region is running must be properly configured so FM/CICS is able to obtain the correct host name of the system.

  You can test your TCP/IP configuration by running the FMNCHKH REXX exec member in FMN.SFMNSAM1.
- TCP/IP and the resolver on the system where the File Manager batch job is submitted must then be able to translate the host name that was obtained and passed by the CICS transaction into an appropriate IP address for the system where the CICS region is running.

Note: The requirements listed above apply for IPv6 networks even if the CICS region and File Manager batch job run on the same system.

Refer to *z/OS Communications Server: IP Configuration Guide* for further details.

Accessing other File Manager functions from the primary option menu

You can allow users to invoke File Manager base function, or FM/DB2 or FM/IMS (if installed), from the FM/CICS primary option menu.

To do this you customize your File Manager base function security, using RACF (or your equivalent security product), or FMNSECUR. For more information refer to Chapter 30, "Customizing the FM/CICS security environment," on page 251.

If you plan to access FM/DB2 from FM/CICS, then ensure that the SYSPROC DD statement in FMN3CICB references the File Manager CLIST library, SFMNCLIB. You should also ensure that the user's FM/CICS startup JCL specifies or defaults to a region size of at least 8M. For information about modifying FMN3CICB, see "Customizing the batch procedure" on page 241.

Providing CICS resource definitions for FM/CICS on interconnected regions

FM/CICS can be used to edit, list, and so on, CICS resources that are owned by remote CICS regions. An active connection must be available on the region where the FM/CICS interface is running, to the resource-owning CICS region. In addition, on each of the remote resource-owning CICS regions, you must:

- Add the FM/CICS load library to the DFHRPL DD statement in the CICS startup procedure. This will be FMN.SFMNMOD2 if you installed FM/CICS into the default libraries.

  If you have installed the FM/CICS Japanese or Korean components, add the FM/CICS Japanese or Korean load library to the DFHRPL DD in front of FMN.SFMNMOD2. The default Japanese load library is FMN.SFMNMODJ and the default Korean load library is FMN.SFMNMODK.
- Define and install the required FM/CICS load modules, FMN3CICS and FMN3LVL, in the CSD group, FMNCICS. You can do this using the job FMN3INSR. FMN3INSR is distributed in FMN.SFMNSAM1. Modify and run this job against the CSD of the interconnected regions to define the required FM/CICS resources on the remote regions. You do this as follows:
  1. Copy the member FMN3INSR from FMN.SFMNSAM1 into your own JCL library.
  2. Modify FMN3INSR in your library.
     - Change the job card to meet your site's requirements.
     - Change #*cicshlq* to the high level qualifier for your CICS libraries.
     - Change #*cee* to the high level qualifier for your Language Environment libraries. Remove the reference if Language Environment is available from LINKLIST.
     - Change #*csddsn* to the data set name of the DFHCSD file.

After you have done this, a resource owned by a remote CICS region can be processed through the FM/CICS interface by specifying the SYSID of the remote CICS region.

Note: The service level (PTF level) of the FM/CICS load modules must be the same (or higher) on all interconnected regions that you want the FM/CICS interface to link to, as on the local region.

This will automatically be true if you use the same FM/CICS load library for all your interconnected CICS regions. If you do not use the same load

library, you must ensure that all the load libraries have the same service level applied. If FM/CICS calls another region to process remote files, an error message is produced if the remote region is not at the required service level.

Limiting the regions to which FM/CICS can connect

By default, FM/CICS will attempt to link to each system that has an active connection defined on the region where the FM/CICS interface is running. You can limit the regions that FM/CICS attempts to link to by performing the following tasks:

1. Create a physical sequential data set (fixed, blocked, with 80 byte records) and enter the APPLID of each CICS region that FM/CICS should link to in the first column of a new line. Sample member FMN3CONN is provided in FMN.SFMNSAM1 which you can copy into your own sequential data set and modify for your requirements. Refer to the sample for more information.
2. An active connection must exist and FM/CICS must be installed on each CICS region entered in this data set.
3. Define the data set created in step 1 as an extrapartition transient data queue (tdqueue) to the CICS region where the FM/CICS interface is running. Refer to member FMN3INST in FMN.SFMNSAM1 library for a sample definition (tdqueue FMNC).
4. Specify the name of the extrapartition tdqueue defined in step 2 on the CONN option of the FM/CICS options module, FMN3POPT. See “Changing the default options” on page 246 for information about the FM/CICS options module.
5. Re-install usermod FMN3UMDP, and refresh FMN3POPT in your CICS region.

After performing these tasks, FM/CICS will only attempt to link to connected CICS regions that:

- Have been listed in the transient data queue defined to the CONN option.
- Have acquired a valid connection on the CICS region where FM/CICS is running.

Chapter 29. Customizing FM/CICS

This chapter describes how to customize FM/CICS. You do this after you have installed FM/CICS. If you plan to use the FM Server method for the CICS FM transaction, you must also perform the customization described in Appendix E, "Customizing the FM Server," on page 435.

Customizing the batch procedure

The batch procedure, FMN3CICB, is required to run the FM/CICS batch job. You will need to customize FMN3CICB before you can use it. You do this as follows:

1. Copy the member FMN3CICB from FMN.SFMNSAM1 into your own procedure library.
2. Modify FMN3CICB in your library.
   - Change #*fmnhlq* to the high level qualifier for the File Manager target libraries.
   - Change #*ipvhlq* to the high level qualifier for the IBM Problem Determination Tools Common Components target libraries. IBM Problem Determination Tools Common Components is required for the operation of FM/CICS. Information about installing IBM Problem Determination Tools Common Components is included in the *File Manager Program Directory*.
   - If the name of the TCP/IP address space is not the default value of 'TCPIP', uncomment the SYSTCPD DD statement and change #*tcpparms* to the data set and member name that contains the appropriate TCPIPJOBNAME parameter.
   - If you plan to access WebSphere MQ queues from FM/CICS, uncomment the three MQ target library statements in the STEPLIB concatenation and change #*mqhlq* to the high level qualifier of your MQ libraries. The required MQ libraries are: SCSQAUTH, SCSQANLE, and SCSQLOAD.
   - If you plan to access FM/DB2 from FM/CICS and use the DB2 DSN command processor, uncomment the DB2 load library statement in the STEPLIB concatenation and change #*db2hlq* to the high level qualifier of your DB2 load library. The DB2 load library is: SDSNLOAD

     If you are using more than one version of DB2, then specify the load library of the earliest version of DB2 you are using.

   Notes:

   a. The member name, FMN3CICB, is used on the **PROCNAME* statement in FMN3PRFD, which you customize for the job FMN3INST. If you change the name of the member here, you must also change the name specified on the **PROCNAME* statement to match when customizing FMN3INST. See "Modifying and submitting FMN3INST" on page 242 for information about FMN3PRFD and FMN3INST.

   b. For customization you can perform for IBM Problem Determination Tools Common Components, see Chapter 36, "Customizing IBM Problem Determination Tools Common Components for use with FM/CICS," on page 281.

© Copyright IBM Corp. 2000, 2011

Profile data sets

FMNPROF (#*fm.profile.dsn*) is used to populate the FM/CICS logon panel.

The data set specified on the *PROFILE statement in FMN3PRFD and in the FM/CICS logon panel is the IBM Problem Determination Tools Common Components profile data set used to store information about the FM/CICS session for the user (similar to an ISPF profile data set).

Modifying and submitting FMN3INST

You use the job FMN3INST (and FMN3PRFD) to provide the default job card and other default values that will be used to populate the logon panel the first time the FM/CICS transaction is run for any given user. After a user logs on for the first time the information will be saved in the profile data set for subsequent invocations.

FMN3INST will define the required CICS table entries, define the FM/CICS profile data set (a VSAM KSDS), and initialize this profile data set with the logon defaults.

The CICS resources defined by FMN3INST must be defined and installed locally on each CICS region where the FM/CICS interface will run.

The VSAM data set FMNPROF (#*fm.profile.dsn*) that is created and loaded by the last two steps of FMN3INST, specifies SHAREOPTIONS(2). However, you can share this FMNPROF data set between multiple CICS systems, if you change this to SHAREOPTIONS(**4**). In this case, you need only run the delete/define and load of the VSAM data set once. An example of how you might find this useful, is if you use a common DFHCSD data set across more than one CICS region.

Note: FMNPROF is used to populate the FM/CICS logon panel.

There are three steps to modifying and running FMN3INST:

- Modify FMN3PRFD
- Modify FMN3INST
- Run FMN3INST

FMN3PRFD

You modify FMN3PRFD as follows:

1. Copy the member FMN3PRFD from FMN.SFMNSAM1 into your own JCL library.
2. Change the following values in FMN3PRFD in your library, to customize the job card and other logon defaults:

   ***JOBCARD**

   Defines the default job card values.

   You can use up to four lines to provide the job card values. There are two variables you can use:

   - **&&TERM** is substituted with the 1 - 4 character terminal ID when the job is submitted.

- **&&USER** is substituted with the 1 - 7 character userid required by FM/CICS when the job is submitted.

Notes:

a. You can terminate the variable with a period which will be removed during substitution.

b. You can use these variables to ensure the default job card is unique for each user.

***PROCNAME=FMN3CICB**
The procedure name is the name of the procedure to be used by the submitted batch job. The IBM-supplied value is FMN3CICB. If you change the name of the procedure then change this statement accordingly. (See "Customizing the batch procedure" on page 241 for information about FMN3CICB.)

PROFILE=profile_dataset_name*
Specifies the default data set name for the user's IBM Problem Determination Tools Common Components profile data set. This value will be used to populate the profile data set name on the logon panel the first time a user invokes FM/CICS. The user can overwrite the system default on the logon panel. The submitted job will allocate the data set if it does not already exist. If it does exist it must be a partitioned data set with an LRECL of 80. The current userid will be substituted for the variable &&USER referenced in the name.

Notes:

a. This data set is the IBM Problem Determination Tools Common Components profile data set and is used to store information about the FM/CICS session for the user (similar to an ISPF profile data set).

b. It is recommended that you define this data set as a PDSE.

c. This is not the same data set as FMNPROF.

***USERID=SIGNON**
FM/CICS requires a userid to be provided when you run the transaction. If you specify *USERID=SIGNON then the userid will default to the CICS signed on userid value when a userid is not provided on transaction invocation. If you remove this statement and you invoke FM/CICS without a userid then FM/CICS will prompt you for a userid. If you specify *PASSWORD=REMEMBER you can remove this statement.

***PASSWORD=REMEMBER**
FM/CICS initially displays a logon panel where you provide userid/password and various other logon information. Specify this statement if, for subsequent invocations of FM/CICS, you want to bypass the logon panel. FM/CICS will remember your password, and if you sign on to CICS with the userid and simply enter the transaction FM, you will bypass the logon panel as long as the logon information recorded by FM/CICS is still valid. To invoke FM/CICS to get the logon panel simply enter:

```
FM userid.
```

The logon panel will also appear if the FM/CICS job does not respond to the CICS region for any reason.

FMN3INST

Modifying and submitting FMN3INST

You modify FMN3INST as follows:

1. Copy the member FMN3INST from FMN.SFMNSAM1 into your own JCL library.
2. Change FMN3INST in your library as follows:
   - Change the job card to meet your site's requirements.
   - Change #*cicshlq* to the high level qualifier for your CICS libraries.
   - Change #*cee* to the high level qualifier for your Language Environment libraries. Remove the reference if Language Environment is available from LINKLIST.
   - Change #*csddsn* to the data set name of the DFHCSD file.
   - Change #*fm.profile.dsn* to the data set name to be used for the VSAM logon profile data set for FM/CICS.
   - Change #*fmnhlq* to the high level qualifier for the File Manager target libraries.
   - Change #*hlq.yourlib*(FMN3PRFD) to refer to your customized version of FMN3PRFD.
   - If you are supplying FM/CICS with a list of APPLIDs of the connected CICS regions where FM/CICS is installed, change #*fm.conn.list* to the data set name that contains the list (a sample list is provided in the sample FMN3CONN, provided in FMN.SFMNSAM1).
   - Change the SHAREOPTIONS value in the second last step, or remove the last two steps completely, if you are sharing the FMNPROF profile data set. See "SHAREOPTIONS" on page 242 for more information.
   - You can also change the CICS resource definition names, listed in Table 54, in the DFHCSDUP step in FMN3INST. Note that changing some of these names also requires a change to the FM/CICS options in FMN3POPI. (See "FMN3POPI" on page 247 for information about changing FMN3POPI.)
   - If necessary, change RESSEC for the transactions FM or FMVU to RESSEC(YES), if you plan to use CICS security to protect resources accessed by these transactions.

Table 54. FM/CICS resource definition names

Resource name	Description	FMN3POPI change required...
FMNPROF	File - logon profile data set	Yes
FM	Transaction - FM/CICS invocation	No
FMVU	Transaction - verify userid	Yes
FMNJ	Transient Data queue name for job submission.	Yes
FMNM	Transient Data queue for FM/CICS message log.	Yes
FMNRDR	DDNAME for transient data queue FMNJ.	No
FMNC	Transient data queue with a list of APPLIDs of connected CICS regions.	Yes

3. When you have made all the required changes to FMN3INST, submit the job. It should finish with return code zero.

Customizing for processing COBOL copybooks

If you want to use COBOL copybooks with FM/CICS, a COBOL compiler must be made available to FM/CICS. File Manager provides an internal version of the COBOL compiler and you can also use another supported COBOL compiler, if one is made available to FM/CICS. Refer to the instructions in "Customizing for processing COBOL copybooks" on page 13, substituting all references to the TSO logon procedure with the FM/CICS batch procedure, FMN3CICB.

For example, a supported COBOL compiler can be made available to FM/CICS by adding the FMNCOB DD statement to FMN3CICB.

This step must be completed if you are using COBOL copybooks through the FM/CICS interface, or through the File Manager base function interface running under FM/CICS.

Note: This functionality requires ADATA, therefore the COBOL compiler must be one that has the ADATA capability. COBOL compilers such as VS COBOL II, and OS/COBOL, which do not support ADATA, cannot be used. All currently supported versions of IBM Enterprise COBOL for z/OS and OS/390 provide this support, and can therefore be used.

Customizing for processing PL/I include books

If you want to use PL/I include books with FM/CICS, the Language Environment runtime library must be available to FM/CICS. Refer to the instructions in "Customizing for processing PL/I include books" on page 17, substituting all references to the TSO logon procedure with the FM/CICS batch procedure, FMN3CICB.

For example, you can add the Language Environment runtime library, SCEERUN, to the STEPLIB DD statement in FMN3CICB.

This step must be completed if you are using PL/I include books through the FM/CICS interface, or through the File Manager base function interface running under FM/CICS.

Customizing for processing HLASM copybooks

If you want to use HLASM copybooks with FM/CICS, ensure that the HLASM load library, ASM.SASMMOD1, is available to FM/CICS. You can do this either by adding it to your linklist, or to the STEPLIB DD statement in FMN3CICB.

You must do this if you are using HLASM copybooks through the FM/CICS interface, or through the File Manager base function interface running under FM/CICS.

Customizing to use DFSORT to improve File Manager performance

You can optionally enable FM/CICS to use the DFSORT COPY function to improve the performance of the data set copy and print functions through the FM/CICS interface. Refer to the instructions in "Customizing to use DFSORT to improve File Manager performance" on page 17, substituting all references to the TSO logon procedure with the FM/CICS batch procedure, FMN3CICB.

For example, you can enable FM/CICS to use DFSORT by adding the SICELINK and SORTLPA libraries, in that order, to the STEPLIB DD statement in FMN3CICB.

This step must be completed if you plan to use DFSORT when running the FM/CICS interface, or through the File Manager base function interface running under FM/CICS.

Changing the default options

Default processing options are supplied with FM/CICS in the module FMN3POPT. You can change these options to suit your installation requirements.

The first part of this module invokes the FMN0POPI macro and allows you to change options related to File Manager processing (for example, audit logging and the print data set). Refer to Appendix A, "File Manager options," on page 289 for a description of the available File Manager options and the values you can specify.

Any option changed in FMN3POPT will be used by the FM/CICS interface and when running the File Manager base interface under FM/CICS.

Notes:

1. If you have previously customized the File Manager base function, the options specified in the FMN0POPT options module (not FMN3POPT) are used when running File Manager base function under ISPF or batch. Therefore, if you want to run File Manager base under ISPF and FM/CICS with similar installation requirements, you must ensure that the options specified in FMN0POPT match the options specified in FMN3POPT.

   For example, if you want File Manager running through ISPF and FM/CICS to use the same audit log HLQ, ensure that the value specified for the AUDITHLQ option is identical in both FMN0POPT and FMN3POPT.

   If you do not plan to run File Manager base function under ISPF or batch, then only FMN3POPT needs to be configured.
2. When running FM/IMS through ISPF or through FM/CICS, all FM/IMS options are taken from FMN1POPT. Therefore you must ensure that the FM/IMS component is installed and configured to your requirements. Refer to "Customizing the FM/IMS installation options module" on page 177 for information about changing options in FMN1POPT.

The second part of the FMN3POPT options module invokes the FMN3POPI macro and allows you to change options specific to the CICS environment.

You use the usermod FMN3UMDP to install your version of FMN3POPT. FMN3UMDP is distributed in FMN.SFMNSAM1.

The options you can change in FMN3POPI are:

- The file name of the CICS profile data set (PROF option).
- The queue name of the transient data queue required for submission of CICS batch jobs (QRDR option).
- The transaction name for the userid verification program (TRANVU option).
- The queue name of the FM/CICS message log transient data queue (MSGL option).
- The name of the transient data queue with a list of APPLIDs of connected CICS regions (CONN option).
- The FM/CICS start type (START option).

- The port of File Manager server if START=TASK is specified (PORT option).
- The host name of File Manager server (HOST option).

You change the options as follows:

1. Copy the member FMN3POPT from FMN.SFMNSAM1 into your own source library.
2. Change the options in FMN3POPT in your library, according to your requirements. For a description of the options in FMN3POPT, and the values you can specify, see Appendix A, "File Manager options," on page 289.
3. Modify FMN3POPI in the copy of FMN3POPT in your library, if required. See "FMN3POPI" for information about FMN3POPI.
4. Modify the FMN3UMDP member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
5. Install SMP/E usermod FMN3UMDP.

Note: You can also use the sample job FMN3POPH to assemble FMN3POPT if you do not want to use SMP/E.

FMN3POPI

You use the options in FMN3POPI to override the FM/CICS defaults for the profile data set file name, the transient data queue name for job submission, and the transaction identifier for the user verification program. Refer to the description of each option for details.

PROF

PROF Specifies the 1-8 character file name used to define the profile data set to CICS. The default is FMNPROF.

QRDR

QRDR
Specifies the 1-4 character transient data queue name that is used by FM/CICS for job submission.

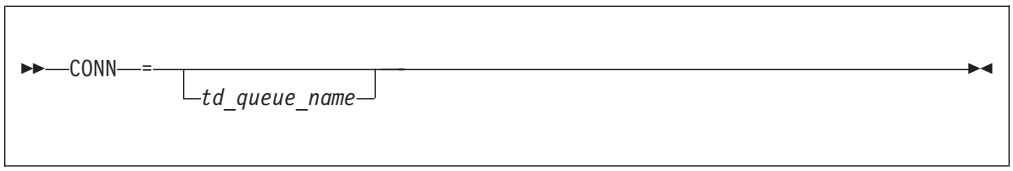
TRANVU

TRANVU
Specifies the 1-4 character transaction ID that is used to run the user verification program.

MSGL

MSGL
Specifies the 1-4 character transient data queue name that is used by FM/CICS as the message log.

CONN



CONN
Specifies the 1-4 character transient data queue name that contains the list of APPLIDs of permitted CICS regions for FM/CICS to access. This parameter is optional.

START

START
Specifies how the FM/CICS address space is to be started.
BATCH
A batch job will be submitted to start the address space.
TASK The address space will be started by connecting to a running FM Server task.

Note: If you specify START=TASK then you must specify a value for PORT and a value for HOST.

TRAN

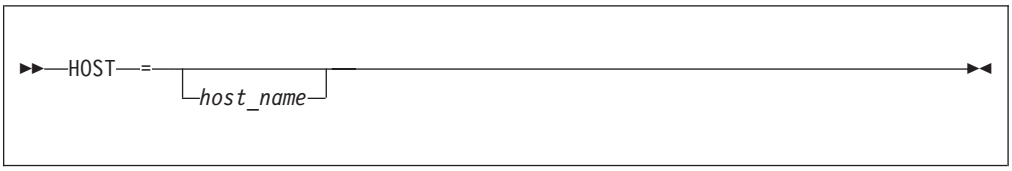
TRAN Specifies the 1-4 character transaction ID for FMN3CICS. The default is FM.

PORT

PORT Specifies a 1-4 digit port number to be dedicated to the FM Server task. FM/CICS will attempt to connect to the task using the specified port.

Note: If you specified TASK for the START option, then you must provide a value for PORT. If you specified START=BATCH then you will need to provide a value for PORT if your users will use the option (START=TASK) to invoke File Manager. Otherwise, any value you specify for PORT is ignored.

HOST



HOST Specifies the host name of the system where the FM server is running when START=TASK has also been specified. The default is no name, and the maximum length of the name is 255 characters.

Note: If you specified TASK for the START option, then you must provide a value for HOST. If you specified START=BATCH then any value you specify for HOST is ignored.

Changing the print and display tables

By default, FM/CICS translates all unprintable characters in printed output to blanks (option PRTTRANS=ON), and non-displayable characters in displayed output to periods. If you want to change the default File Manager translation tables, and you have not already performed this task when configuring File Manager base function for ISPF or batch, then refer to “Changing the print and display translation tables” on page 28.

Customizing to use an I/O exit

FM/CICS allows you to provide a user I/O exit for use when processing data. This enables you to process records in resources which are subject to any kind of pre- or post-processing not offered directly by File Manager. Refer to Chapter 9, “Customizing File Manager to use an I/O exit,” on page 73 for further details about the exit.

To specify that an I/O exit can be used through the FM/CICS interface, or through the File Manager base function interface under FM/CICS, set the USEIOX option in FMN3POPT. If required, you can also provide the name of a site-specific I/O exit. For more information about USEIOX, see “USEIOX” on page 326. For information about changing the options in FMN3POPT, see “Changing the default options” on page 246. This exit name can also be overridden by the user interactively. See the *File Manager User's Guide and Reference* for information about how to use the I/O exit.

Chapter 30. Customizing the FM/CICS security environment

Security can be provided for File Manager functions, including the functions available through the FM/CICS interface, using RACF (or an equivalent security product) or through the FMNSECUR exit. If you have not already configured your File Manager base function security, see Chapter 4, "Customizing the File Manager security environment," on page 31 for further information.

Customizing to protect update functions in FM/CICS

You can protect update functions in FM/CICS by means of the facility class FILEM.CICS.UPDATE, together with the SEC option in FMN3POPT.

If you set SEC=YES in FMN3POPT and then give or deny users access to the FILEM.CICS.UPDATE facility class, you can protect the following functions when running FM/CICS.

DSC	Data set copy
CDE	CICS transient data edit
CFE	CICS file edit
CTE	CICS temporary storage edit
DSE	Data set edit
DSEB	Data set edit batch
DSG	Data set create
DSU	Data set update
FCH	Find/change - only protected if the function attempts the update command

The normal security checking of FILEM.FUNCTION.*function_code* applies to these functions when SEC=YES has been specified in FMN3POPT. If you do not specify SEC=YES, then no checking is done for these functions.

Customizing to invoke other File Manager components from FM/CICS

You can allow users to invoke File Manager base function, or FM/DB2 or FM/IMS (if installed), from the FM/CICS primary option menu. To do this you customize your File Manager base function security, using RACF (or your equivalent security product), or FMNSECUR, to give the userids running FM/CICS a minimum of read access to the facility groups, FILEM.CICS.BASE, FILEM.CICS.IMS or FILEM.CICS.DB2, as appropriate. For information about these facility groups, see page 32 or page 39. For more information about FM/IMS security, see Chapter 22, "Customizing the FM/IMS security environment," on page 191.

If you provide this access then the FM/CICS primary option menu will include the functions FM (for File Manager base function), FM/IMS, and FM/DB2. These functions can then be invoked through the FM/CICS interface.

If you do not provide this access, then these functions will not appear on the menu, and will not be available.

Note: If you plan to run the File Manager base function interface through the FM/CICS session, you must first complete the installation and customization of File Manager base function as described in Part 1, "Customizing File Manager," on page 1. If you plan to run the FM/IMS

© Copyright IBM Corp. 2000, 2011

interface through the FM/CICS session, you must first complete the installation and customization of FM/IMS as described in Part 3, "Customizing File Manager IMS Component," on page 161. If you plan to run the FM/DB2 interface through the FM/CICS session, you must first complete the installation and customization of FM/DB2 as described in Part 2, "Customizing File Manager DB2 Component," on page 97.

Chapter 31. CICS security and FM/CICS

Introduction

This chapter describes CICS security as it applies to FM/CICS. For complete information about CICS security, refer to the *CICS Transaction Server for z/OS RACF Security Guide* for your CICS release.

To allow FM/CICS to process userids and passwords correctly, the CICS system should be set up to run with SEC=YES. Without this option the FM/CICS logon panel will not be able to verify or change passwords.

This chapter provides an overview of CICS security that applies to the functions performed by FM/CICS. It is intended to help you:

- Ensure proper security is implemented in your environment to prevent unauthorized access to CICS resources or commands through FM/CICS.
- Determine why a NOTAUTH response was returned from an FM/CICS function.

  Note that a resp2 value of 100 indicates a command security failure, whereas a resp2 value of 101 indicates a resource security failure.

Notes:

1. CICS Security is turned on for the region by specifying SEC=YES as a SIT parameter.
2. To use CICS Security to protect resources accessed in a transaction, you must specify RESSEC(YES) for the transactions FM or FMVU. You can do this in FMN3INST (see page 243). The IBM-supplied value for RESSEC in FMN3INST for FM and FMVU, is RESSEC(NO).

CICS security considerations for FM/CICS

To provide the necessary security for your CICS regions, CICS uses the z/OS system authorization facility (SAF) to route authorization requests to an external security manager (ESM), such as RACF, at appropriate points within CICS transaction processing. FM/CICS does not use any undocumented or unsupported interfaces that bypass CICS security. Therefore, CICS security can be utilized to protect your system from unauthorized access to CICS resources through FM/CICS.

When CICS Security prevents successful execution of an FM/CICS function, a NOTAUTH response is displayed. To ensure proper authorization to CICS resources through FM/CICS, consider the following aspects of CICS Security in your environment:

Logon and user security

CICS security must be turned on in the region (SEC=YES) for FM/CICS to verify and change passwords from the FM/CICS logon panel.

© Copyright IBM Corp. 2000, 2011

Resource security

All files, temporary storage queues, and transient data queues can be protected through CICS resource security. Appropriate authority must be assigned to the userid for a user to process (edit and view) resources through FM/CICS.

Note the following exception to CICS resource security for extrapartition transient data queues. When browsing an extrapartition transient data queue through FM/CICS, the batch FM address space will read the records directly from the data set the queue is defined to. Therefore, the z/OS userid supplied to FM/CICS must have a minimum of READ authority to the appropriate RACF data set profile.

Command security

CICS command security allows CICS security checking to be performed on system programming commands. FM/CICS issues the following commands that are subject to command security checking when listing and processing CICS resources:

- INQUIRE
- SET

Therefore, the userid must have sufficient authority to issue the commands listed above to successfully perform the FM/CICS function.

Note: A userid must have authority to issue the CICS INQUIRE command to run the FM/CICS interface.

Transaction security

CICS transaction security prevents unauthorized users from running specific CICS transactions. Therefore, a userid must have appropriate authority to run the transaction that executes the FM/CICS program. Also, FM/CICS will link to other CICS regions to process remote resources. This requires the userid to have appropriate authority to run the mirror transaction that executes the FM/CICS program on the remote region.

Intercommunication security

When multiple CICS regions are interconnected, FM/CICS can be used to access (list and edit) resources on multiple regions. To process remote resources, FM/CICS will function ship requests and link to the remote CICS regions. Therefore, the userid must pass the CICS Link Security check and optional User Security check on the remote region to successfully complete the FM/CICS function.

Chapter 32. Customizing the FM/CICS audit facility (FMN3POPT controlled auditing)

You can use File Manager audit logging to create an audit trail of editing activity against resources processed through the FM/CICS interface. You can create your audit trail using either SMF or recording the activity to an audit log data set.

FM/CICS determines whether or not an audit trail is produced based on the settings of the AUDITLOG and SMF options in the FMN3POPT options module.

If audit logging is enabled, audit log records will be produced for all edit activity against resources through the FM/CICS interface. This includes both CICS resources and non-CICS data sets (if running the File Manager base function interface under FM/CICS).

You can force the production of an audit trail, and report on the changes made, by specifying `AUDITLOG=DEMAND` in FMN3POPT. The job submitted will be determined by skeleton member FMN3FTAD found in FMN.SFMNSLIB. You should customize the job card and JCL in this skeleton to specify the reporting options you require.

Important Information

If you plan to enable audit logging for FM/CICS edit activity or for File Manager base function edit activity when running under the FM/CICS interface, then you must specify values for AUDITLOG and (if required) AUDITHLQ and SMFNO, in FMN**3**POPT.

If you want to enable audit logging for File Manager base functions in batch and under ISPF, then you must specify values for AUDITLOG, AUDITHLQ and SMFNO in the base function options module, FMN**0**POPT.

Audit logging under FM/CICS and CICS logging

FM/CICS uses standard CICS APIs to process CICS resources. Therefore, any established CICS logging is performed when running FM/CICS.

However, FM/CICS audit logging is completely independent of CICS logging. Therefore, CICS logging (including CICS audit logging) can be used with FM/CICS. You could use CICS audit logging as an alternative to using FM/CICS audit logging.

Using System Management Facilities (SMF) for audit logging

If you plan to use SMF for audit logging, you must do the following:

- Ensure that the load module FMNSMF is APF-authorized. For more information about using SMF for audit logging, and making FMNSMF APF-authorized, see page 49.
- Add the load module FMNSMF to the AUTHTSF list in member IKJTSO*xx* in SYS1.PARMLIB. If you do not do this, even if you have selected to record to SMF and you have specified an SMF record type, no recording will be done.

© Copyright IBM Corp. 2000, 2011

- Select an SMF record number for the audit log records, and include it in your SMF parmlib member SMFPRM*xx*.
- Specify this SMF record number in the options module, FMN3POPT. For information on how to do this see "Changing the default options" on page 246.

To activate any changes you have made to SYS1.PARMLIB members, either restart your system, or use the appropriate commands for your site to dynamically activate the changes.

To report on the audit trail information collected by SMF, you must extract this information from SMF to your own data set. The information in this data set can then be printed by the FM/CICS Print Audit Trail utility. To do this select option 3.6 from the FM/CICS Primary Option menu.

A sample job, FMNSMFX, is provided in FMN.SFMNSAM1 to help you extract the SMF data to your own data set. See the comments in the job for information about the changes you need to make to the job. For more information about SMF, see *z/OS MVS System Management Facilities (SMF)*.

Recording an audit trail to a data set

If you choose to record your audit trail to a data set, you can print this information using the FM/CICS Print Audit Trail utility. To do this select option 3.6 from the FM/CICS Primary Option menu.

The format of the audit log data set name is determined by the setting of the AUDITHLQ parameter in the FMN0POPI definition in FMN3POPT. See "AUDITHLQ" on page 291 for more information about the AUDITHLQ option.

The following data set name formats may be generated:

```
userid.FMNLOG.Dyymmdd.Thhmmss                    (when AUDITHLQ= (blank))
audithlq.userid.FMNLOG.Dyymmdd.Thhmmss           (when AUDITHLQ=audithlq)
qual1.<qual2.><qual3.>Dyymmdd.Thhmmss            (when AUDITHLQ=qual1.<qual2.><qual3>)
```

where:

audithlq Is any 1-8 character constant that is valid in the context of a data set name

userid Is the userid creating the data set

Dyymmdd Is the date of the activity

Thhmmss Is the time of the activity

When AUDITHLQ contains one or more periods, the AUDITHLQ value is treated as a data set prefix, with one, two or three levels. Each level of the prefix can be:

XXX Any 1-8 character constant that is valid in the context of a data set name.

&&PREFIX Indicates that the user's TSO prefix should be used. This will be null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix will also be null.

&&USER Indicates that the user's logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&UID Indicates that the user's TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user's TSO logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&FUNCOD Indicates that the File Manager internal function code should be

used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.

Notes:

- The format of the data set name for audit log data sets created under FM/CICS is the same as that for audit log data sets created by the File Manager base function. To distinguish between the two, use the AUDITHLQ option in FMN0POPT and FMN3POPT as appropriate.
- If you specify a value for AUDITHLQ in FMN3POPT, this value will apply to audit data sets for FM/CICS and for data sets for File Manager base function running under FM/CICS.

Recording an audit trail to a data set

Chapter 33. Customizing the FM/CICS audit facility (SAF-rule controlled auditing)

Controlling FM/CICS auditing using SAF

You can use SAF to control the following aspects of FM/CICS auditing:

- Whether FM/CICS writes audit records to SMF, the user's audit log data set, or both.
- Whether FM/CICS writes audit records when accessing resources.
- Whether FM/CICS writes audit records only for update-related activities, for read and update-related activities, or functional information only.
- Whether FM/CICS automatically produces the audit log report at the end of a FM/CICS session ("Demand" audit logging).

For users wanting to control auditing by means of parmlib, the users being audited must have read access or more to the SAF facility class profile FILEM.PARMLIB.CICS. In this case, auditing is controlled by the FMAUDIT parameter in the FMN3PARM member. For more information, see "FM/CICS options specified in FMN3PARM" on page 428.

Note: File Manager for CICS will not start if a user has read access to this facility class and the FMN3PARM member is unavailable from parmlib.

Example: User PROD1 to have SAF processing determined from parmlib

Write this RACF rule:

```
RDEF FACILITY FILEM.PARMLIB.CICS AUDIT(NONE) UACC(NONE)
       OWNER(ownerid)
PE FILEM.PARMLIB.CICS ACC(READ) ID(PROD1)  CLASS(FACILITY)
```

If SAF processing is not active, or the rule is not defined, or the rule is defined and the user has no access, then no parmlib processing is performed.

If there is no parmlib processing to specify SAF-controlled auditing, then SAF-controlled auditing is performed for users that have read access to the SAF facility class profile FILEM.SAFAUDIT.CICS.

Example: User PROD2 to have SAF auditing without parmlib:

Write this RACF rule:

```
RDEF FACILITY FILEM.SAFAUDIT.CICS AUDIT(NONE) UACC(NONE)
       OWNER(ownerid)
PE FILEM.SAFAUDIT.CICS ACC(READ) ID(PROD2)  CLASS(FACILITY)
```

Note: If SAF-controlled auditing is specified without parmlib processing, then the SMF record type is obtained from the options module SMFNO value.

© Copyright IBM Corp. 2000, 2011

When a security product other than RACF is in use

File Manager V11 APAR PM31842 introduced a change to remove the mandatory requirement for definition of members in the logical PARMLIB concatenation. In order to achieve this, File Manager makes RACROUTE calls, with STATUS=ACCESS, to SAF FACILITY profiles:

FILEM.PARMLIB.**
FILEM.SAFAUDIT.** (only if the first call fails)

STATUS=ACCESS is specified because the RACROUTE call should not audit the call. The documentation in the *z/OS V1R11 Security Server RACROUTE Macro Reference*, in the description of RACROUTE REQUEST=AUTH,STATUS=ACCESS, states:

ACCESS
The request is simply to return the user's highest current access to the resource specified. Upon successful completion, the user's access is returned in the RACF reason code. **No auditing is done for this request.**

When RACF is used, the STATUS=ACCESS request works as documented, and no security-related logging or abends are generated, even if you do not have access to the profile.

However, when non-RACF security products (such as ACF2) are used, S047 ABEND047 may be issued in response to the above RACROUTE request.

The ACF2 global system options (GSO) definitions shown here have been used to avoid the S047 abends:

```
SET CONTROL(GSO)
INSERT SAFDEF.APFFM PROGRAM(FMN*****) RB(FMN*****) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=FACILITY,STATUS=ACCESS)
INSERT SAFDEF.APFFM2 PROGRAM(FILEM***) RB(FILEM***) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=FACILITY,STATUS=ACCESS)
F ACF2,REFRESH(SAFDEF)
INSERT SAFDEF.APFFM PROGRAM(FMN*****) RB(FMN*****) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
INSERT SAFDEF.APFFM2 PROGRAM(FILEM***) RB(FILEM***) NOAPFCHK -
RACROUTE(REQUEST=AUTH,CLASS=XFACILIT,STATUS=ACCESS)
F ACF2,REFRESH(SAFDEF)
```

Implementing SAF-rule controlled auditing

Use the following checklist to implement SAF-rule controlled auditing:

__ 1. Determine the SAF FACILITY and XFACILIT rules that will be required. See "Understanding how FM/CICS uses SAF rules to control auditing" on page 261 for more information.

__ 2. Write the relevant SAF rules. See the examples in "SAF rule examples" on page 264.

__ 3. Modify the FMN3PARM member to activate SAF-rule controlled auditing. See "FM/CICS options specified in FMN3PARM" on page 428 for more information.

__ 4. Test the configuration to ensure that auditing is occurring as required. You can use the facilities for customizing the FMN3PARM definitions to, for example, turn on SAF-rule controlled auditing for a specific logonid (or logonids), without changing the behaviour for other users.

__ 5. When testing is complete, activate SAF-rule controlled auditing for all FM/CICS users by changing the FMN3PARM member.

Understanding how FM/CICS uses SAF rules to control auditing

SAF (System Authorization Facility) allows applications, such as FM/CICS, to define "resources" that might need to be protected. The "resource" to be protected need not be something specific, such as a data set; it can be essentially any type of resource or facility that the application considers to be important. For FM/CICS and auditing, the "resource" is the ability to write audit log records. The resource names reflect either the type of auditing that is to occur (eg to SMF), or the FM/CICS function and resource.

FM/CICS uses two types of SAF resource names to control auditing. The SAF resource rules used by FM/CICS to control auditing are shown in Table 58 on page 266 and Table 59 on page 266).

Understanding SAF rule access levels

SAF provides for five levels of access to any FACILITY or XFACILIT resource. The levels of access form a hierarchy, so that a user with the highest level of access to a resource also has access to all the lower levels. The levels of access are specified in RACF rules using the following mnemonics:

NONE No access
READ Level 1 access
UPDATE Level 2 access
CONTROL Level 3 access
ALTER Level 4 access.

It is important to understand that the mnemonics used (READ, UPDATE and so on) can and do mean different things, depending on the context in which the SAF resource name is used. This can be confusing since READ and UPDATE have obvious meanings when it comes to, for example, accessing a data set. For SAF rules used to control FM/CICS audit, it may aid understanding to think of the mnemonics as indicating level 1 access and level 2 access.

For the SAF resource rules used by FM/CICS, the meanings of the various levels of access are:

NONE
The user does not have access to the resource; this typically means the user cannot write audit log records.

READ The user has level 1 access to the resource; this typically means that the user can write audit log records.

UPDATE
The user has level 2 access to the resource. This level of access only has meaning for FACILITY rule 2 (see Table 58 on page 266). A user with level 2 access can write audit log records to the user's audit log data set, and the audit log data set will be printed at the end of the user's session (on-line execution only). This is equivalent to the DEMAND audit option in the non-SAF case.

CONTROL
The user has level 3 access to the resource. This level of access is not used by FM/CICS.

ALTER
The user has level 4 access to the resource. This level of access is not used by FM/CICS.

How FM/CICS determines whether audit log records should be written

The determination of whether audit records are to be written for a particular FM/CICS function and a given TSO logonid follows this three step process:

Step 1. What is the FMAUDIT specification in the FMN3PARM member?

The FMAUDIT specification setting in the FMN3PARM member (in SYS1.PARMLIB) is the "master" switch for SAF-rule controlled auditing. Note that there are facilities available to specify different settings in the FMN3PARM member for different TSO logonids, see “FM/CICS options specified in FMN3PARM” on page 428 for more information. For any given TSO logonid, there are two possibilities:

SAF_CTRL=NO

SAF-rule controlled auditing is not in effect. Auditing is determined by the settings in the FMN3POPT module, see Chapter 32, “Customizing the FM/CICS audit facility (FMN3POPT controlled auditing),” on page 255.

SAF_CTRL=YES

SAF-rule controlled auditing is in effect. Processing continues to step 2.

Step 2. Does the user have access to write audit records?

This is determined by the user's access to rules 1 and 2 in Table 58 on page 266; the various outcomes are summarized in Table 55.

Table 55. Determination of a user's ability to write audit log records

TODSN access[1]	TOSMF access[2]	OPTION access[3]	Can write audit records?	Demand logging?	"Create audit trail" option[4]
NONE	NONE	ANY	No	No	Not visible
READ	NONE	NONE	Yes, data set only	No	Not visible
READ	NONE	READ	Yes, data set only	No	Visible
UPDATE	NONE	NONE	Yes, data set only	Yes	Not visible
UPDATE	NONE	READ	Yes, data set only	Yes	Visible
NONE	READ	NONE	Yes, SMF only	No	Not visible
NONE	READ	READ	Yes, SMF only	No	Visible
READ	READ	NONE	Yes, to data set and SMF	No	Not visible
READ	READ	READ	Yes, to data set and SMF	No	Visible
UPDATE	READ	NONE	Yes, to data set and SMF	Yes	Not visible
UPDATE	READ	READ	Yes, to data set and SMF	Yes	Visible

Table 55. Determination of a user's ability to write audit log records (continued)

TODSN access[1]	TOSMF access[2]	OPTION access[3]	Can write audit records?	Demand logging?	"Create audit trail" option[4]

Notes:

1. Refers to the level of access the user has to SAF FACILITY rule 1 in Table 58 on page 266.
2. Refers to the level of access the user has to SAF FACILITY rule 2 in Table 58 on page 266.
3. Refers to the level of access the user has to SAF FACILITY rule 3 in Table 58 on page 266.
4. The visibility of the "Create audit trail" option does not influence whether a user can write audit log records, although the user must have access to write audit log records (to either a data set or SMF), for the option to be visible.

If the user does not have the ability to write audit log records, then no check of SAF resource names in Step 3 occurs.

A user's access to write audit log records at Step 2 only indicates that auditing might occur, the final decision depends on the user's level of access to the XFACILIT resource name (or names) that apply to the particular FM/CICS function.

Step 3. Does the user have access to write audit records for the current function and data set? The XFACILIT resource name(s) used by FM/CICS to determine whether audit records should be written depend on the FM/CICS function being executed and the data set being accessed.

Table 56 shows the function codes which are supported.

Table 56. FM/CICS function codes that can be audited using SAF

Function code	Online option	Description
CSL	Delete prefix command	Delete queue
CTB	Browse prefix command	Browse temporary storage queue
CTE	2	Edit temporary storage queue
CTV	1	View temporary storage queue
CTP	3.2	Print temporary queue
CDB	Browse prefix command	Browse transient data queue
CDE	2	Edit transient data queue
CDV	1	View transient data queue
CDP	3.2	Print transient data queue
CFB	Browse prefix command	Browse file
CFE	2	Edit file
CFV	1	View file
CFP	3.2	Print file

Controlling where FM/CICS writes audit log records

You can use SAF to control whether FM/CICS writes audit log records to SMF, the user's audit log data set, or to both.

How FM/CICS determines whether audit log records should be written

The following table shows the SAF FACILITY class resource names used to control FM/CICS logging to SMF and the user's audit log data set.

Table 57. SAF FACILITY class resource names controlling disposition of FM/CICS audit records

FACILITY Class Name	Purpose
FILEM.AUDIT3.TOSMF	Enables or disables auditing to SMF for FM/CICS.
FILEM.AUDIT3.TODSN	Enables or disables auditing to the user's data set for FM/CICS.

SAF rule examples

This section shows SAF rule examples under different conditions.

Controlling where FM/CICS writes audit log records

You can use SAF to control whether FM/CICS writes audit log records to SMF, the user's audit log data set, or to both.

Table 58 on page 266 shows the SAF FACILITY class resource names used to control FM/CICS to logging and the user's audit log data set.

Example 1
- Disable audit logging to a user data set for all FM/CICS users.
- Enable FM/CICS audit logging to SMF for the PROD logonid.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT3.TOSMF[1]
RDEL FACILITY FILEM.AUDIT3.TODSN[1]
RDEF FACILITY FILEM.AUDIT3.TOSMF UACC(NONE) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT3.TODSN UACC(NONE) OWNER(XXXXXXX)[3]
PE FILEM.AUDIT3.TOSMF ACC(READ) ID(PROD) CLASS(FACILITY)[4]
```

Explanation:
1. Delete any existing facility rule.
2. Define the facility rule for audit logging to SMF (TOSMF suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
3. Define the facility rule for audit logging to the user's audit log data set (TODSN suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
4. Allow logonid PROD to write audit log records (ACC(READ)) to SMF.

Example 2
- Enable audit logging to a user data set for all FM/CICS users.
- Enable demand logging for the following users, PROD1, PROD2, PROD3.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT3.TOSMF[1]
RDEL FACILITY FILEM.AUDIT3.TODSN[1]
RDEF FACILITY FILEM.AUDIT3.TOSMF UACC(NONE) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT3.TODSN UACC(READ) OWNER(XXXXXXX)[3]
PE FILEM.AUDIT3.TODSN ACC(UPDATE) ID(PROD1) CLASS(FACILITY)[4]
PE FILEM.AUDIT3.TODSN ACC(UPDATE) ID(PROD2) CLASS(FACILITY)[4]
PE FILEM.AUDIT3.TODSN ACC(UPDATE) ID(PROD3) CLASS(FACILITY)[4]
```

Explanation:
1. Delete any existing facility rule.
2. Define the facility rule for audit logging to SMF (TOSMF suffix). UACC(NONE) is used so that any user, for which there is no specific rule, has no access.
3. Define the facility rule for audit logging to the user's audit log data set (TODSN suffix). UACC(READ) is used so that any user, for which there is no specific rule, has read access, and can therefore write audit log records.
4. Allow logonids PROD1, PROD2, PROD3 to write audit log records with automatic printing of the audit report ("Demand logging") (ACC(UPDATE)), to SMF.

Example 3
- Disable audit logging completely for all FM/CICS users.
- Enable dual logging for all FM/CICS users.

You could write the following RACF rules:

```
RDEL FACILITY FILEM.AUDIT3.TOSMF[1]
RDEL FACILITY FILEM.AUDIT3.TODSN[1]
RDEF FACILITY FILEM.AUDIT3.TOSMF UACC(READ) OWNER(XXXXXXX)[2]
RDEF FACILITY FILEM.AUDIT3.TODSN UACC(READ) OWNER(XXXXXXX)[3]
```

Explanation:
1. Delete any existing facility rule.
2. Define the facility rule for audit logging to SMF (TOSMF suffix). UACC(READ) is used so that any user, for which there is no specific rule, has access (and can therefore write audit records to SMF).
3. Define the facility rule for audit logging to to the user's audit log data set (TODSN suffix). UACC(READ) is used so that any user, for which there is no specific rule, has access (and can therefore write audit records to the user's audit log data set).

Controlling auditing of FM/CICS functions

You can use SAF to control whether FM/CICS audit log records for FM/CICS functions which access resources. Table 56 on page 263 shows FM/CICS functions codes which may be logged.

Example 1
- Enable audit logging of all modifications to temporary storage queue TS01 using the FM/CICS temporary storage queue edit function for all users except TSO logonid MAINT1.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT3.CTE.UPDATE.TS01[1]
RDEF XFACILIT FILEM.AUDIT3.CTE.UPDATE.TS01 OWNER(XXXXXXXX) UACC(READ)[2]
PE FILEM.AUDIT3.CTE.UPDATE.TS01 CLASS(XFACILIT) ID(MAINT1) ACC(NONE)[3]
```

Explanation:
1. Delete any existing XFACILIT rule.
2. Define the XFACILIT rule to log all modifications to temporary storage queue TS01 using the FM/CICS temporary storage queue edit function (CTE). UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).
3. A specific rule for logonid MAINT1 to prevent audit log records being written.

Example 2

- Enable audit logging of all modifications together with all records read for transient data set TD01 using the FM/CICS transient data queue edit function (CDE) for user SERVIC1.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT3.CDE.ALL.TD01[1]
RDEF XFACILIT FILEM.AUDIT3.CDE.ALL.TD01 OWNER(XXXXXXXX) UACC(NONE)[2]
PE FILEM.AUDIT3.CDE.ALL.TD01 CLASS(XFACILIT) ID(SERVIC1) ACC(READ)[3]
```

Explanation:

1. Delete any existing XFACILIT rule.
2. Define the XFACILIT rule to log all modification and all records read for transient data queue TD01 using the FM/CICS transient data queue edit function (CDE). Uacc(NONE) specified that no TSO userids write audit log records (in the absence of any over-riding more specific rule).
3. A specific rule for logonid SERVIC1 to write audit log records.

Example 3

- Enable audit logging of functional information for data set FMN.CICS.KSDS using the FM/CICS file print utility for all users.

You could write the following RACF rules:

```
RDEL XFACILIT FILEM.AUDIT3.CFP.FUNCTION.FMN.CICS.KSDS[1]
RDEF XFACILIT FILEM.AUDIT3.CFP.FUNCTION.FMN.CICS.KSDS OWNER(XXXXXXXX) UACC(READ)[2]
```

Explanation:

1. Delete any existing XFACILIT rule.
2. Define the XFACILIT rule to log function information when data set FMN.CICS.KSDS is printed using the FM/CICS file print utility (CFP). UACC(READ) allows all TSO userids to write audit log records (in the absence of any over-riding more specific rule).

FM/CICS auditing FACILITY and XFACILIT class resource names

These two tables (and associated tables) list FACITITY and XFACILIT class resource names and details.

Table 58. FM/CICS auditing FACILITY class resource names

Rule Number	Resource Name[1]	Purpose
1	FILEM.AUDIT3.TODSN	Allows a user to write audit log records to the user's audit log data set.
2	FILEM.AUDIT3.TOSMF	Allows a user to write audit log records to SMF.
3	FILEM.AUDIT3.OPTION	Allows the user access to the "Create audit trail" option on the FM/CICS Edit panels.

Table 59. FM/CICS auditing XFACILIT class resource names

Resource Name	Purpose
FILEM.AUDIT3.functioncode.ALL[1].resource[2]	Allows users to write audit log records for all records which are read and modified for the specified data set (resource) using the FM/CICS function (function code).

Table 59. FM/CICS auditing XFACILIT class resource names (continued)

Resource Name	Purpose
FILEM.AUDIT3.functioncode.UPDATE.resource	Allows users to write audit log records for all modifications to the specified data set (resource) using the FM/CICS function (function code).
FILEM.AUDIT3.functioncode.FUNCTION.resource	Allows users to write audit log records containing information for the specified data set (resource) using the FM/CICS function (function code).

Notes:

1. It is recommended that this option be used with caution. The size of the data set being accessed and the editing technique used will have an impact on the number of read records logged. And this may have an impact on the performance of FM/CICS.
2. If the resource is a file, the data set name and not the CICS resource name must be specified. For temporary storage queues, transient data queues, and enqueues, the CICS resource name must be specified.

Chapter 34. Customizing FM/CICS for national languages

If you are using a language other than English (for example, French), you will need to perform some or all of the customization tasks listed in Table 60.

Unlike other File Manager components, if you are using Japanese or Korean you will have to perform the first task, *Setting the LANGUAGE option in FMN3POPT*, even if you have installed the FM/CICS Japanese or Korean components.

You will also have to do this if you plan to use the File Manager base function interface under FM/CICS.

Table 60. Summary of steps for customizing FM/CICS for a national language

Step	Description	Page
__ 1	Set the LANGUAGE option in FMN3POPT.	269
__ 2	Verify that your CICS terminal is defined as DBCS-capable, if necessary.	270
__ 3	Translate the FM/CICS logon messages to your language.	270
__ 4	Provide a version of FMN3MENU (FM/CICS logon messages) for your language.	270
__ 5	Translate the FM/CICS panels to your language.	271

Changing the print and display translation tables for languages other than English

DBCS languages
This step is essential if you are using a DBCS language other than Japanese or Korean.

If you plan to use FM/CICS with a national language other than English, you might need to provide a print and display translation table for your language. You do this as part of your customization for File Manager base function. See "Changing the print and display translation tables for languages other than English" on page 61.

You should also specify PRTTRANS=ON in FMN3POPT. If you are using any DBCS language you might also need to specify TERMTYPE=3270KN in FMN3POPT.

Setting the LANGUAGE option

FM/CICS uses the LANGUAGE option set in the FM/CICS options module, FMN3POPT. The setting of LANGUAGE in FMN3POPT is used for all FM/CICS messages, and panels.

Therefore, if you are using a language other than English (for example, French), you will need to set the LANGUAGE option in FMN3POPT to one of the values shown in Table 61 on page 270.

© Copyright IBM Corp. 2000, 2011

You will need to do this if you are using Japanese or Korean, even if you have installed the File Manager and FM/CICS Japanese or Korean components.

See "Changing the default options" on page 23 and "Changing the default options" on page 246 for information on how to do this.

Table 61. Keyword values for the LANGUAGE option

Language	Code	Specify on LANGUAGE option...
French	FRA	FRENCH
German	DEU	GERMAN
Italian	ITA	ITALIAN
Japanese	JPN	JAPANESE
Portuguese	PTG	PORTUGUESE
Spanish	ESP	SPANISH
Danish	DAN	DANISH
Upper case English	ENP	UPPERENG
Korean	KOR	KOREAN
Swiss German	DES	SGERMAN
Traditional Chinese	CHT	CHINESET
Simplified Chinese	CHS	CHINESES
Other	XXX	OTHER

For example, to use French messages, specify LANGUAGE=FRENCH

Verifying that the CICS terminal is DBCS-capable

If you are using a DBCS language, verify that the CICS terminals are defined as DBCS-capable. To do this, issue the CICS transaction

```
CECI ASSIGN
```

and verify that the *SOSI* data area contains X'FF'.

Translating the FM/CICS logon messages

The language used in FM/CICS logon messages is determined by the setting of the LANGUAGE option in FMN3POPT. All FM/CICS messages are stored in the FMN3MENU source member. This CSECT is part of the root FM/CICS module so that an English version of the messages is always available. If you are using a language other than English, Japanese or Korean, you must provide a version of FMN3MENU for your language, as described in "Providing a national language version of FMN3MENU."

Providing a national language version of FMN3MENU

FMN3MENU contains the assembler source for the FM/CICS messages. To provide translated versions of the FM/CICS logon messages:

1. Copy the member FMN3MENU from FMN.SFMNSAM1 to your own source library with the name FMN3M*yyy*, where *yyy* is one of the following language codes:
   FRA French
   DEU German

ITA	Italian
JPN	Japanese
PTG	Portuguese
ESP	Spanish
DAN	Danish
ENP	Upper case English
KOR	Korean
DES	Swiss German
CHT	Traditional Chinese
CHS	Simplified Chinese
XXX	Other

2. Change the message text in FMN3M*yyy* in your library.
3. Modify the FMN3UMDM member in FMN.SFMNSAM1 to meet your site's requirements, using the same language code as above. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN3UMDM.
5. Define and install program FMN3M*yyy* to your CICS region, where *yyy* is the same language code as above. You can do this by updating and running the FM/CICS installation job FMN3INST. See "Modifying and submitting FMN3INST" on page 242 for information about FMN3INST.

To use the messages you have translated, see "Using the translated messages and panels" on page 272.

Translating the panel text

All FM/CICS panels are provided in English. They are also provided in Japanese if you have installed the FM/CICS Japanese component, and in Korean if you have installed the FM/CICS Korean component. You can translate some or all of these panels into another language. (If no translated version of a particular panel is available, FM/CICS uses the English version.)

All FM/CICS panels are stored in FMN.SFMNPENU. You translate a panel as follows:

1. Find the panel members in FMN.SFMNPENU that you want to translate. The panel members specific to FM/CICS are all named FMN3*zzzz*.
2. Create a library with the same characteristics as FMN.SFMNPENU, with the name FMN.SFMNP*yyy*, where *yyy* is the same language code you specified when you modified FMN3MENU. If you have already created a library with this name for other translated File Manager panels, use that library. Copy the required panel members from FMN.SFMNPENU to this library.
3. Change the required panel text in the members in your library. A panel may reference a **help** panel by means of a `.HELP` statement. If any panel you are changing contains any of these `.HELP` statements, also copy and change these referenced members in your library.

To use the panels you have translated, see "Using the translated messages and panels" on page 272.

Using the translated messages and panels

The language used in messages and panels displayed through the FM/CICS interface is determined by the setting of the LANGUAGE option defined in FMN3POPT. To use your translated messages and panels you must include your libraries in the message and panel concatenations in front of the File Manager English libraries, in FMN3CICB.

For example, to use your translated messages, add FMN.SFMNM*yyy* to FMN3CICB, to the IPVPLIB DD statement, in front of FMN.SFMNMENU. To use your translated panels, add FMN.SFMNP*yyy* to FMN3CICB, to the IPVPLIB DD statement, in front of FMN.SFMNPENU.

See “Customizing the batch procedure” on page 241 for information about FMN3CICB.

Customizing for the Japanese national language

The other customization task you might need to do for the Japanese national language is to modify the supplied Japanese translation tables. If you want to do this, do so as part of the customization of the File Manager base function. See “Modifying the Japanese translation tables” on page 65.

Changing the Japanese logon message text

If you have installed the FM/CICS Japanese component, all FM/CICS Japanese logon messages are stored in the FMN3MJPN source member. Normally you should not need to modify this module. However, if you do want to modify it, you can do so by means of the usermod, FMN3UMDN.

To do this:

1. Copy the member FMN3MJPN from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN3MJPN in your library.
3. Modify the FMN3UMDN member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN3UMDN.

Customizing for the Korean national language

The other customization task you might need to do for the Korean national language is to modify the supplied Korean translation tables. If you want to do this, do so as part of the customization of the File Manager base function. See “Modifying the Korean translation tables” on page 66.

Changing the Korean logon message text

If you have installed the FM/CICS Korean component, all FM/CICS Korean logon messages are stored in the FMN3MKOR source member. Normally you should not need to modify this module. However, if you do want to modify it, you can do so by means of the usermod, FMN3UMDQ.

To do this:

1. Copy the member FMN3MKOR from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN3MKOR in your library.
3. Modify the FMN3UMDQ member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN3UMDQ.

Changing the Korean logon message text

Chapter 35. Verifying the customization of FM/CICS

After you have completed all the necessary steps involved in the initial installation and customization of FM/CICS, you can perform the following steps to verify your customization, before completing the installation process. You might need to refer to the *File Manager User's Guide and Reference for CICS*.

You verify the installation and customization of FM/CICS by logging on to a CICS system where you plan to run FM/CICS. Before you can do this, you must have customized FM/CICS, your CICS system, and, if necessary, TCP/IP.

1. Log on to the CICS system where you have implemented FM/CICS, and enter the FM/CICS transaction, FM.
2. Enter your userid and press Enter.
3. The FM/CICS logon panel is displayed:

```
------------------------ File Manager for CICS Logon -------------------------

     Enter Logon parameters

     Userid . . . . . . . . FMUSER

     Password . . . . . . .             New Password . . .

     Node . . . . . . . . . FMNODE      (Machine the job is to be run on)

     Procedure. . . . . . . FMC62FM4    (Procedure to run File Manager)

     Profile data set . . . FMUSER.IPV.IPVPROF

     Prefix . . . . . . . . FMUSER      (Default prefix for data sets)

     Jobcard
     //FM&TERM.B JOB (,,,,),&USER,
     //     MSGCLASS=A,MSGLEVEL=(1,1),CLASS=A


     F1=Help   F3=Logoff   F4=Reset   F5=Wait   F6=Default   F12=Cancel
     Enter=Submit
```

If you specified START=TASK in the parameters in FMN3POPT, or you used (START=TASK) to initiate your FM/CICS session, and a valid port number has been provided, then you will see a logon panel similar to this.

© Copyright IBM Corp. 2000, 2011

```
------------------------ File Manager for CICS Logon -------------------------

     Enter Logon parameters

     Userid . . . . . . . . FMUSER

     Password . . . . . . .            New Password . . .

     Profile data set . . . FMUSER.IPV.IPVPROF

     Prefix . . . . . . . . FMUSER      (Default prefix for data sets)

     Port . . . . . . . . . 2054

     Host name



     F1=Help   F3=Logoff   F4=Reset   F6=Default   F12=Cancel
```

4. Enter your password and press Enter.
5. Unless using the START=TASK option, the batch job to start the FM/CICS session will be submitted, and then the FM/CICS Primary Option Menu should be displayed, as shown in Figure 25 on page 277.
   - If after submitting the FM/CICS batch job, message FMNCA016 is displayed (Job *jobname* not responding), the FM/CICS job may have failed or may be queued. Examine the job log for further details and possible error messages.

     If the job log cannot be found (the FM/CICS job was never started) verify that the FMNRDR DD statement has been properly defined in the CICS startup JCL and that your system has sufficient initiators. See "Updating the CICS startup procedures" on page 237 for further information.

     Some of the most common reasons the batch job may have failed or been queued are:
     - Insufficient authority to access one of the libraries in the FM/CICS procedure. Check the system log for security messages.
     - TCP/IP is not active on the CICS region. Check for TCP/IP error messages in the job log and verify that TCP/IP is active on the CICS region.
     - The CICS Sockets interface has not been started
     - Duplicate job name. Verify if another user may have specified the same job name in the FM/CICS **Jobcard** input field, or a previous FM/CICS job is still active.
     - JCL errors. Check the job log for further information.

```
 Process   Options   Help
_____________________________________________________________________________
FM/CICS                      Primary Option Menu

0  Settings      Set processing options                User ID . : FMUSER
1  View          View data                             CICS User : FMC5DFLT
2  Edit          Edit data                             CICS Appl : C62D2FM5
3  Utilities     Perform utility functions             Date. . . : 2008/09/02
4  Templates     Template and copybook utilities       Time. . . : 15:33
FM FM            File Manager z/OS
FI FM/IMS        File Manager for IMS z/OS
FD FM/DB2        File Manager for DB2 z/OS
X  Exit          Terminate FM/CICS

Processing Options:
 CICS Resource
 1  1. File
    2. Temporary Storage
    3. Transient Data



Command ===> ___________________________________________________________________
F1=Help     F3=Exit     F4=CRetriev F7=Backward F8=Forward  F10=Actions
F12=Cancel
```

Figure 25.

Note: The options **FM** for File Manager z/OS, and **FI** for File Manager for IMS z/OS, allow the user to invoke File Manager and FM/IMS from this primary option panel. These options will not appear if you did not customize File Manager base function security to allow this access. For more information, see "Accessing other File Manager functions from the primary option menu" on page 239.

6. Enter `VER` on the command line to display the release level and PTF level of FM/CICS and IBM Problem Determination Tools Common Components. A panel is displayed similar to this:

```
IBM File Manager for z/OS Version 11 Release 1
CICS Component
(not APF authorized)

Service Levels of installed components

            Base       IMS        DB2        CICS       IPV
English     -NONE-     -NONE-     -NONE-     -NONE-     -NONE-
```

Notes:

a. This will always show FM/CICS as `not APF authorized`, even if you have made File Manager APF-authorized, as FM/CICS cannot run APF-authorized.

b. When you first install FM/CICS, `-NONE-` will be shown under each component that you have installed. Subsequently, when you have applied service to FM/CICS, a PTF number will be shown, indicating the PTF level of each component you have installed. If you have not installed a component, that component will not be shown at all. If you have installed the Japanese or Korean language components, another line will be displayed indicating the service level of that component.

7. Press the Exit key (PF3) to exit from the VER display.

8. Enter VERCICS on the command line to display the level and version of the running File Manager batch job and FM/CICS on all connected CICS regions. A panel is displayed similar to this:

```
 Process   Options   Help
_____________________________________________________________________________
FM/CICS                      Primary Option Menu
                                 CICS Levels

 IBM File Manager for z/OS Version 11 Release 1 CICS Component

 Current PTF: -NONE-      Level: 1

    Connected FM/CICS Levels
    Sysid    Applid      Version    PTF        Level
             C62D2FM5    V11R1      -NONE-         1

 Command ===> _____________________________________________Scroll PAGE
 F1=Help     F3=Exit     F4=CRetriev F7=Backward F8=Forward  F10=Actions
 F12=Cancel
```

Notes:

a. The version, PTF level, and buffer layout level for the running File Manager batch job, and FM/CICS on all interconnected CICS regions (if the FMN3LVL program is available on those regions), is displayed.

b. The level of the File Manager CICS component and all FM/CICS programs should be the same, otherwise some functions will fail.

c. If more than one interconnected CICS region is displayed, each region should be at the same level as the running FM/CICS batch job. If the level or version is different, these values will be highlighted in red, indicating a possible error.

d. A difference in version or level is likely to be due to maintenance not being applied to all instances of FM/CICS, or the FM/CICS programs not being refreshed in the CICS region after maintenance was applied.

e. If FMN3LVL is not available on a connected system, the text LOAD ERROR *condition* will be displayed. Ensure that FMN3LVL can be loaded on the connected CICS region and retry the VERCICS command.

f. If an entry for an interconnected CICS region where FM/CICS has been installed is not displayed in the list, this indicates one of the following:

   - An active connection to the region was not acquired when the VERCICS command was issued.
   - The FM/CICS program FMN3CICS could not be linked to and loaded on the connected CICS region.
   - A TD queue has been specified for the CONN option in FMN3POPT and the APPLID of the connected CICS region has not been added to this queue.

   Verify that an active connection to the remote region has been acquired and FMN3CICS is available, then retry the VERCICS command.

9. Press the Exit key (PF3) to exit from the VERCICS display.
10. Press the Exit key (PF3) twice more to exit from FM/CICS.

Complete the installation

You can now complete the installation of FM/CICS by performing the ACCEPT processing. The steps involved are described in the *File Manager Program Directory*.

Verifying the customization of FM/CICS

Chapter 36. Customizing IBM Problem Determination Tools Common Components for use with FM/CICS

IBM Problem Determination Tools Common Components is required for the operation of FM/CICS. There are some customization tasks you can perform for IBM Problem Determination Tools Common Components. You do this after you have installed IBM Problem Determination Tools Common Components. Information about installing is included in the *File Manager Program Directory*.

The customization tasks you can perform are:

- Translate the IBM Problem Determination Tools Common Components message text into your national language.
- Translate the IBM Problem Determination Tools Common Components panel text into your national language.
- Change the Japanese message text.
- Change the Korean message text.

Library names after you finish installing

This chapter assumes you have installed IBM Problem Determination Tools Common Components into the default libraries. The default high level qualifier is IPV. Therefore after you have installed IBM Problem Determination Tools Common Components the names of the target and distribution libraries will be those listed in Table 62. If you have installed the IBM Problem Determination Tools Common Components Japanese component, the Japanese target and distribution libraries will be those listed in Table 63 on page 282. If you have installed the IBM Problem Determination Tools Common Components Korean component, the Korean target and distribution libraries will be those listed in Table 64 on page 282.

Table 62. IBM Problem Determination Tools Common Components target and distribution libraries

Target library name	Distribution library name	Usage
IPV.SIPVMOD1	IPV.AIPVMOD1	IBM Problem Determination Tools Common Components load modules
IPV.SIPVSAM1	IPV.AIPVSAM1	Sample jobs
IPV.SIPVPENU	IPV.AIPVPENU	IBM Problem Determination Tools Common Components English panels
IPV.SIPVMENU	IPV.AIPVMENU	IBM Problem Determination Tools Common Components English messages
IPV.SIPVTENU	IPV.AIPVTENU	IBM Problem Determination Tools Common Components English tables

© Copyright IBM Corp. 2000, 2011

Table 63. IBM Problem Determination Tools Common Components Japanese target and distribution libraries

Target library name	Distribution library name	Usage
IPV.SIPVPJPN	IPV.AIPVPJPN	IBM Problem Determination Tools Common Components Japanese panels
IPV.SIPVMJPN	IPV.AIPVMJPN	IBM Problem Determination Tools Common Components Japanese messages
IPV.SIPVTJPN	IPV.AIPVTJPN	IBM Problem Determination Tools Common Components Japanese tables

Table 64. IBM Problem Determination Tools Common Components Korean target and distribution libraries

Target library name	Distribution library name	Usage
IPV.SIPVPKOR	IPV.AIPVPKOR	IBM Problem Determination Tools Common Components Korean panels
IPV.SIPVMKOR	IPV.AIPVMKOR	IBM Problem Determination Tools Common Components Korean messages
IPV.SIPVTKOR	IPV.AIPVTKOR	IBM Problem Determination Tools Common Components Korean tables

Translating the message text

All IBM Problem Determination Tools Common Components messages are provided in English. They are also provided in Japanese if you have installed the Japanese component of IBM Problem Determination Tools Common Components, and in Korean if you have installed the Korean component of IBM Problem Determination Tools Common Components. You can translate these messages into another language. (If no translated version of a particular message is available, IBM Problem Determination Tools Common Components uses the English version.)

All IBM Problem Determination Tools Common Components messages are stored in IPV.SIPVMENU. You translate a message as follows:

1. Find the message members in IPV.SIPVMENU that you want to translate.
2. Create a library with the same characteristics as IPV.SIPVMENU, with the name IPV.SIPVM*yyy*, where *yyy* is one of the following language codes:
   FRA French
   DEU German
   ITA Italian
   JPN Japanese
   PTG Portuguese
   ESP Spanish
   DAN Danish
   ENP Upper case English
   KOR Korean
   DES Swiss German
   CHT Traditional Chinese
   CHS Simplified Chinese

XXX Other

Copy the required message members from IPV.SIPVMENU to this library.

3. Change the required message text in the members in your library.

To use the messages you have translated, see "Using the translated messages and panels."

Translating the panel text

All IBM Problem Determination Tools Common Components panels are provided in English. They are also provided in Japanese if you have installed the Japanese component of IBM Problem Determination Tools Common Components, and in Korean if you have installed the Korean component of IBM Problem Determination Tools Common Components. You can translate these panels into another language. (If no translated version of a particular panel is available, IBM Problem Determination Tools Common Components uses the English version.)

All IBM Problem Determination Tools Common Components panels are stored in IPV.SIPVPENU. You translate a panel as follows:

1. Find the panel members in IPV.SIPVPENU that you want to translate.
2. Create a library with the same characteristics as IPV.SIPVPENU, with the name IPV.SIPVP*yyy*, where *yyy* is the same language code you specified if you translated any messages. Copy the required panel members from IPV.SIPVPENU to this library.
3. Change the required panel text in the members in your library. A panel may reference a **help** panel by means of a `.HELP` statement. If any panel you are changing contains any of these `.HELP` statements, also copy and change these referenced members in your library.

To use the panels you have translated, see "Using the translated messages and panels."

Using the translated messages and panels

To use your translated messages, add your messages library, IPV.SIPVM*yyy*, to the IPVPLIB concatenation in front of IPV.SIPVMENU, in FMN3CICB.

To use your translated panels, add your panel library, IPV.SIPVP*yyy*, to the IPVPLIB concatenation in front of IPV.SIPVPENU, in FMN3CICB.

See "Customizing the batch procedure" on page 241 for information about FMN3CICB.

Changing the Japanese message text

If you have installed the Japanese component of IBM Problem Determination Tools Common Components, all IBM Problem Determination Tools Common Components Japanese messages are stored in IPV.SIPVMJPN. Normally you should not need to modify any Japanese message, however, if you do want to modify them, you can do so in IPV.SIPVMJPN.

Changing the Korean message text

If you have installed the Korean component of IBM Problem Determination Tools Common Components, all IBM Problem Determination Tools Common Components Korean messages are stored in IPV.SIPVMKOR. Normally you should not need to modify any Korean message, however, if you do want to modify them, you can do so in IPV.SIPVMKOR.

Chapter 37. IBM File Manager plug-in for Eclipse

IBM File Manager Plug-in for Eclipse, when used with an Eclipse- based platform (like CICS Explorer or Rational Developer for System z), provides easy access through a graphical user interface (GUI) on the workstation to the power of File Manager for z/OS. The GUI provides a workstation alternative to the z/OS ISPF interface. With this interface, existing application developers and system programmers may experience productivity gains over the traditional "green screen" interface. The interface can also lower the learning curve for new z/OS developers, making them proficient quicker by providing a GUI interface to the z/OS tool that is potentially easier and more familiar for new developers to use.

The File Manager plug-in, when integrated with File Manager for z/OS and the Eclipse-based platform, provides a set of tools for manipulating and browsing data stored in various z/OS resources including data sets, UNIX System Services files, and IBM WebSphere MQ queues. The File Manager plug-in supports these usage scenarios:

- **Manipulation and browsing of data using the File Manager editor.** The editor provides three different views of data:
  - Character display (text-based representation of data).
  - Formatted/table display using a template.
  - Single display using a template (focused on the display of fields defined in a single record).
- **Creation, modification, and updating of File Manager templates:**
  - Create templates from COBOL, PL/I, and HLASM copybooks.
  - Create dynamic templates on the fly (if you do not have a copybook to describe your data).
- **Use of wizards to access utility functions to:**
  - Allocate, delete, and rename data sets.
  - Copy, compare, and create data in resources, including z/OS data sets, WebSphere Message Queues, and USS files.
  - View load modules.

Installation requirements

This section describes the host and client requirements for IBM File Manager plug-in for Eclipse.

Host requirements

IBM File Manager plug-in for Eclipse requires File Manager V11.1 with the PTF for APAR PM34376, or later. The FM server needs to be configured for the client to communicate with the host system. See Appendix E, “Customizing the FM Server,” on page 435.

Client requirements

IBM File Manager plug-in for Eclipse is designed to work in an Eclipse environment such as IBM CICS Explorer and IBM Rational Developer for System z. File Manager plug-in has been tested to work in these environments:

- IBM CICS Explorer V1.0.1

© Copyright IBM Corp. 2000, 2011

- IBM CICS Explorer V1.1.0
- IBM Rational Developer for System z V7.6
- IBM Rational Developer for System z V8.0

The system requirements of your workstation depend on the target environment where you are installing the File Manager plug-in. Refer to the respective system requirements page of your target environment for more information.

Host installation and configuration

For File Manager clients to communicate with your z/OS system, you must first configure the File Manager server. For a full description of how to do this, see Appendix E, "Customizing the FM Server," on page 435. To establish connection, File Manager plug-in users must know the IP address of your z/OS system and the port number where the File Manager server is listening.

For Rational Developer for System z users, the RSE server must be configured so that it knows the port number of the File Manager server running on the system. To do this, define a variable called `FM_SERVER_PORT` in your RSE server environment by adding this statement in the rsed.envvars configuration file:

```
FM_SERVER_PORT=xxxx
```

where *xxxx* is an integer port number of the FM server.

For information about how to change your RSE environment, see *IBM Rational Developer for System z Host Configuration Guide*.

Note: Chapter 6 in the *IBM Rational Developer for System z Host Configuration Guide* describes configuration requirements using FMIEXT.properties. These steps are no longer required by File Manager plug-in for Eclipse.

Client installation

The installation of IBM File Manager plug-in for Eclipse depends on the target environment where you are installing the plug-in. There are two installation methods:

- Installation using the *Eclipse P2 installation mechanism* for CICS Explorer users.
- *IBM Installation Manager* for Rational Developer for System z users.

Eclipse P2 installation for CICS Explorer users

To install IBM File Manager plug-in for Eclipse using the Eclipse P2 installation mechanism for CICS Explorer, follow these steps:

1. Download the zip file containing IBM File Manager plug-in for Eclipse. You can find the latest version from the IBM z/OS Problem Determination Tools site. The zipped file is called FM_plugin_V11*xxx*.zip.
2. Start your CICS Explorer.
3. Click **Help > Software updates...**.
4. Under the **Available Software** tab, click **Add Site...**.
5. In the Add Site dialog, click the **Archive...** button and select the zipped file.
6. Select the **IBM File Manager plug-in for Eclipse** feature from the list. Optionally you may need to select and install EMF run-time for IBM File Manager plug-in for Eclipse if your CICS Explorer environment does not have a version of EMF run-time environment installed.

7. Click the **Install...** button to install IBM File Manager plug-in for Eclipse. After successful installation, CICS Explorer prompts you to restart CICS Explorer.
8. Select **Yes** to re-start CICS Explorer and complete the installation.

When you have completed the installation steps, File Manager plug-in is ready for use. You can now open the Systems Information view. To do this:

1. Select **Window > Show View > Others...**
2. Select the **Systems Information** view under the File Manager category.

Installation Manager installation for Rational Developer for System z users

To install IBM File Manager plug-in for Eclipse using the IBM Installation Manager, follow these steps:

1. If you have an older version of File Manager plug-in which was installed as part of the installation process for Rational Developer for System z (the feature referred to as FMI), you must first uninstall the old FMI feature using the Installation Manager.
2. Download the zip file containing IBM File Manager plug-in for Eclipse for Rational Developer for System z. You can find the latest version from the IBM z/OS Problem Determination Tools site. The zipped file is called FM_RDZplug_V11*xxx*.zip.
3. Unzip the downloaded file into a new folder. You should find these items:
   - A readme.html file.
   - A folder containing license information.
   - A zipped file which is required by the IBM Installation Manager to install the File Manager plug-in.
4. Start the IBM Installation Manager and click **File > Preferences ...**.
5. Click on the **Add Repository...** button in the Repositories section.
6. Select the **Browse...** option and specify the zipped file containing the File Manager plug-in. You must specify the FM_Offering_V*xxxxx*.zip file which is found in the downloaded zipped file (FM_RDzPlugin_V*xxxxx*.zip).
7. Press OK.
8. Press OK again to close the preferences page.
9. Select the **Install** option in the Installation Manager. You should see "IBM® File Manager® plug-in for Eclipse" in the list of installation packages. Select this package and press **Next**.
10. You should see the license agreement for the usage of IBM File Manager plug-in for Eclipse. You must accept the terms to install the software. Press **Next**.
11. Under the Location page, you should see a list of Package Groups that you have installed on your computer (for example, Rational Developer for System z). You must select a target group to install the File Manager plug-in. Select a group and press **Next**.
12. In the Features page, you should see "IBM® File Manager® plug-in for Eclipse" as selected. Press **Next**.
13. Press **Install** in the Summary page. You should see the message "The packages are installed" displayed in the Installation Manager. If any error occurs during the install, record the error details and send them to IBM support.
14. Press **Finish** to complete the installation.

After following these installation steps, File Manager plug-in is ready for use. The File Manager related functions appear in the context menu in the Remote Systems view in Rational Developer for System z (see Figure 26).

For File Manager plug-in functions to work correctly, the host configuration steps must be completed first. See “Host installation and configuration” on page 286.

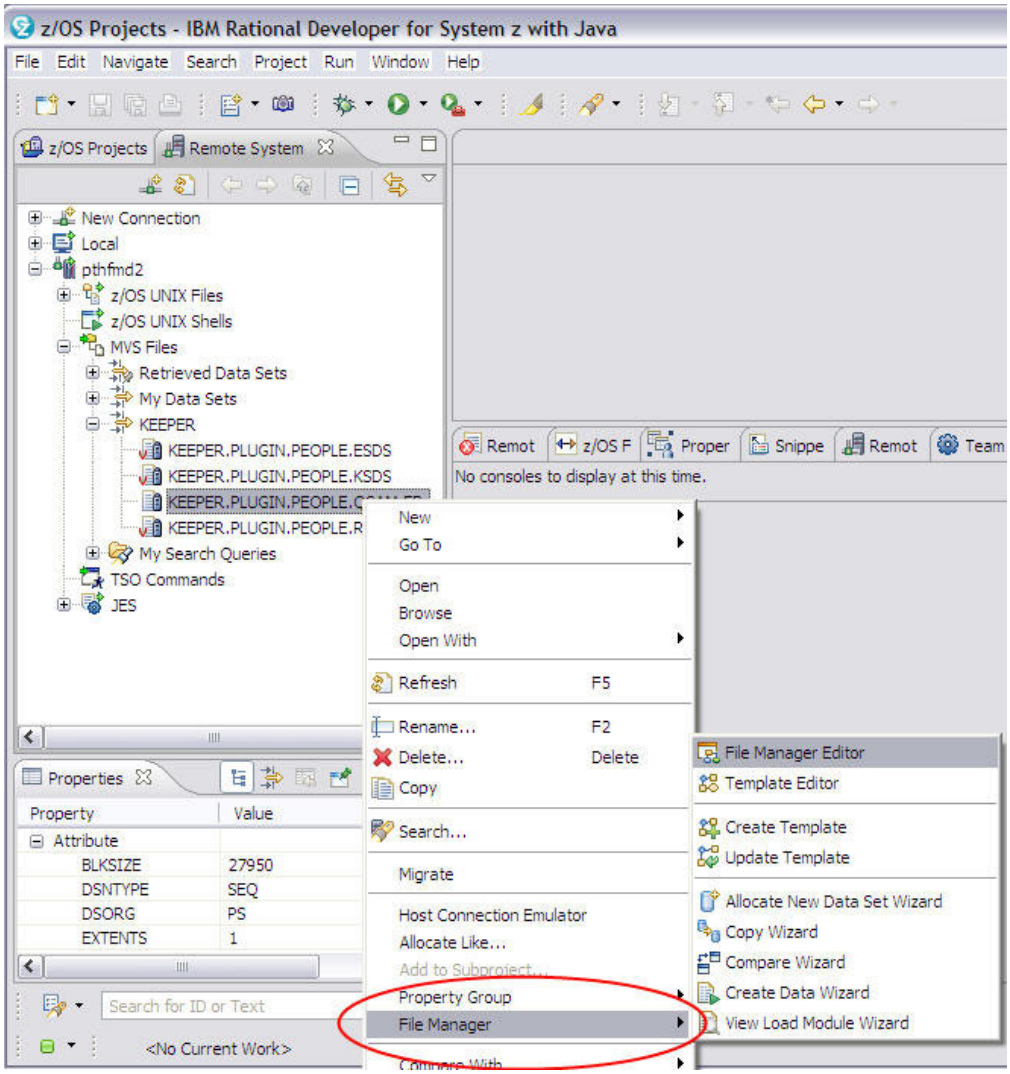


Figure 26. Screenshot of IBM File Manager plug-in for Eclipse

Problem Diagnostics

A detailed explanation of problem diagnostics is provided in the IBM File Manager plug-in for Eclipse online help which is installed into your Eclipse environment as part of the installation process described above. Select **Help > Help Contents**. You can find "IBM File Manager plug-in for Eclipse Help" in the Table of Contents section of the online manual. You can find the Problem Diagnostics section under the General Overview section.

Appendix A. File Manager options

This section describes the File Manager options. The syntax described here applies to the FMN0POPT macro, which is installed by default in FMN.SFMNSAM1. You can modify these File Manager options to suit your requirements, for File Manager base function, FM/DB2, FM/IMS, and FM/CICS.

However, if you want to change an option setting for File Manager base function, and also for FM/DB2, FM/IMS, or FM/CICS, you must change the option in each of FMN0POPT, FMN2POPT, FMN1POPT, FMN3POPT, or FMN4POPT and apply the appropriate usermods. The options only take effect in the component where they are applied.

For information about changing File Manager base function options, see "Changing the default options" on page 23.

For information about changing FM/DB2 options, see "Changing the default options" on page 126.

For information about changing FM/IMS options, see "Customizing the FM/IMS installation options module" on page 177.

For information about changing FM/CICS options, see "Changing the default options" on page 246.

For File Manager Plugin for Eclipse users, the information about base function options applies, however the options module is contained separately in FMN4POPT.

ABENDCC

ABENDCC

Specifies how File Manager processing resulting in a non-zero return code (including customized return codes) will be completed in batch.

NONE

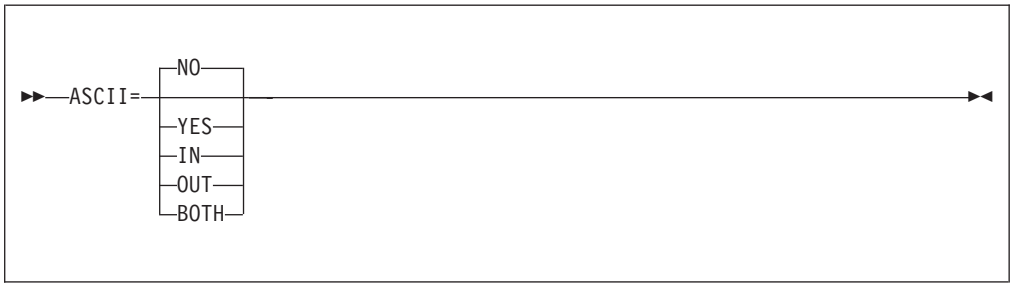
The job step will terminate with a non-zero return code. The default is NONE.

nnnnn If the return code is greater than or equal to *nnnnn*, File Manager will abend with U999, Reason Code=888 (hex: 378). *nnnnn* can be any integer between 1 and 32767. *nnnnn = 0* is the same as NONE. *nnnnn* can be selected as appropriate, for specific return codes (including customized return codes) issued by File Manager functions.

© Copyright IBM Corp. 2000, 2011

Note: Specification of any non-zero integer will prevent File Manager from intercepting abnormal terminations (system abends).

ASCII



ASCII Specifies the requirements for translation of tape data between ASCII and EBCDIC.

NO No translation is performed.

IN or **YES**
Translates tape input from ASCII format to EBCDIC format.

OUT Translates tape output from EBCDIC format to ASCII format before writing it to the output tape.

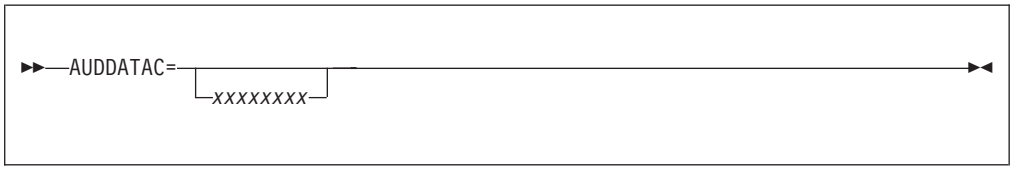
BOTH Translates tape input from ASCII format to EBCDIC format, and translates tape output from EBCDIC format to ASCII format.

Use this option to convert ASCII tapes from one ASCII character set to another.

Input translation is done using File Manager's ASCII/EBCDIC translate table; output translation is done using File Manager's EBCDIC/ASCII translate table.

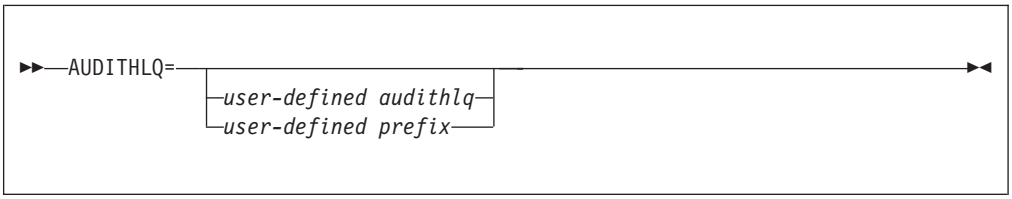
Note: Changing the ASCII option has no effect for FM/DB2, FM/IMS, or FM/CICS.

AUDDATAC



AUDDATAC
Specifies the SMS data class that File Manager uses when allocating the File Manager Audit Log. The maximum length you can specify for AUDDATAC is eight (8) bytes. AUDDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied.

AUDITHLQ



AUDITHLQ
Specifies an optional high-level qualifier, or data set prefix, for the File Manager audit log data sets.

The format of the generated data set names is shown in Table 65.

Table 65. The format of generated data set names

FM component	AUDITHLQ value	Audit log data set name format
Base, FM/CICS	blank (none)	`userid.FMNLOG.Dyymmdd.Thhmmss`
Base, FM/CICS	*audithlq*	`audithlq.userid.FMNLOG.Dyymmdd.Thhmmss`
Base, FM/CICS	*prefix*	`qual1.<qual2.><qual3.>Dyymmdd.Thhmmss`
FM/DB2	blank (none)	`userid.FMN2AUD.<ssid>.Dyymmdd.Thhmmss`
FM/DB2	*audithlq*	`audithlq.FMN2AUD.<ssid>.Dyymmdd.Thhmmss`
FM/DB2	*prefix*	`qual1.<qual2.><qual3.>Dyymmdd.Thhmmss`
FM/IMS	blank (none)	`userid.IMSAUDIT.Dyymmdd.Thhmmss`
FM/IMS	*audithlq*	`audithlq.IMSAUDIT.Dyymmdd.Thhmmss`
FM/IMS	*prefix*	`qual1.<qual2.><qual3.>Dyymmdd.Thhmmss`

where:

audithlq Is any 1-8 character constant that is valid in the context of a data set name

userid Is the userid creating the data set

Dyymmdd Is the date of the activity

Thhmmss Is the time of the activity

prefix `qual1.<qual2.><qual3.>`, as defined below

The following values can be specified for each level (qual1, qual2, or qual3) of the data set name prefix.

XXX Any 1-8 character constant that is valid in the context of a data set name.

&&PREFIX Indicates that the user's TSO prefix should be used. This will be null if TSO NOPREFIX is in effect.

&&USER Indicates that the user's logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&UID Indicates that the user's TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user's logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&FUNCOD Indicates that the File Manager internal function code should be used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.

&&SSID Indicates that the currently connected subsystem name (DB2 or IMS only) should be used. If this symbolic parameter is specified in FMN0POPT or FMN3POPT, the value is ignored, the substituted value is null.

Notes:

- When a symbolic parameter is specified (any value preceded by &&), a trailing period is required, except for qual3.
- You cannot specify a combination of characters and symbolic parameters at the same level of the data set name prefix.
- You cannot use more than one symbolic parameter at the same level of the data set name prefix.
- You cannot specify a prefix value with more than 3 levels.
- You can specify a trailing period to distinguish between a high level qualifier value and a prefix value comprising a single level.

Examples:

AUDITHLQ= Default

AUDITHLQ=FMNAUD, HLQ value is FMNAUD

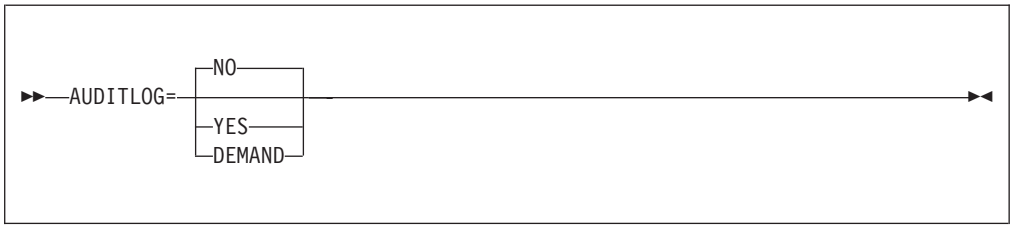
AUDITHLQ=FMNAUD., Prefix value is FMNAUD

AUDITHLQ=&&USER..FMN1AUD
Two-level prefix

AUDITHLQ=FMNAUD.&&FUNCOD..&&USER
Three-level prefix

AUDITHLQ=FMNAUD.&&USER..&&SSID
Three-level prefix, SSID non-blank (DB2 and IMS only)

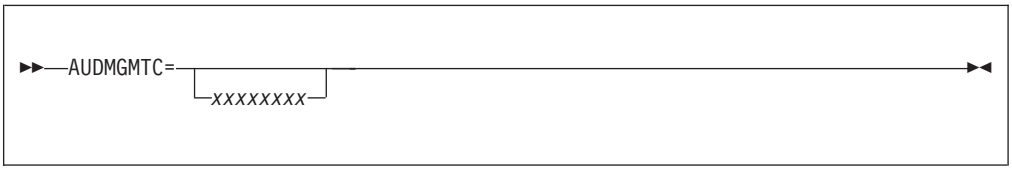
AUDITLOG



AUDITLOG
Specifies whether or not audit trail logging for the File Manager base function or for FM/CICS is mandatory.
NO Audit trail logging is not mandatory.
YES Audit trail logging is mandatory.
DEMAND
Audit trail logging is mandatory and a job described by skeleton FMN0FTAD (FMN3FTAD for FM/CICS) will be submitted to report on the changes made by a user when exiting the edit function.

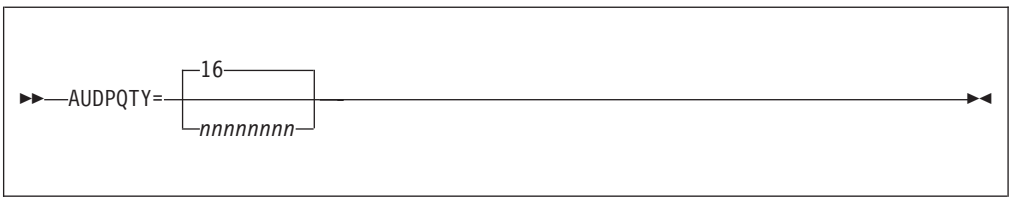
Note: Changing the AUDITLOG option has no effect if the relevant FMN*n*PARM PARMLIB member has FMAUDIT **SAF_CTRL=YES**.

AUDMGMTC



AUDMGMTC
Specifies the default SMS management class that File Manager uses when allocating the File manager Audit Log. The maximum length you can specify for AUDMGMTC is eight (8) bytes. AUDMGMTC is optional. If omitted the SMS management class specified in PMGMT is used, if supplied.

AUDPQTY

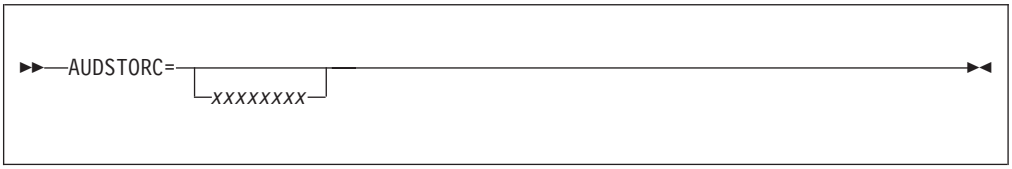


AUDPQTY
Specifies the amount of DASD space to be used for primary space allocation of the File Manager Audit Log. The range depends on the space unit specified and the DASD device type.

AUDSQTY

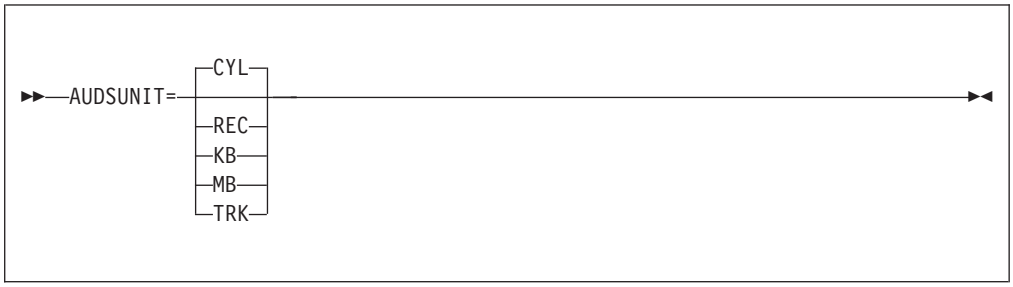
AUDSQTY
Specifies the amount of DASD space to be used for secondary space allocation of the File Manager Audit Log. The range depends on the space unit specified and the DASD device type.

AUDSTORC



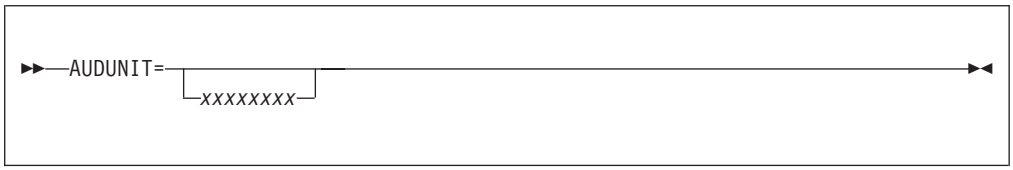
AUDSTORC
Specifies the default SMS storage class that File Manager uses when allocating the File Manager Audit Log. The maximum length you can specify for AUDSTORC is eight (8) bytes. AUDSTORC is optional. If omitted the SMS storage class specified in PSTORC is used, if supplied.

AUDSUNIT



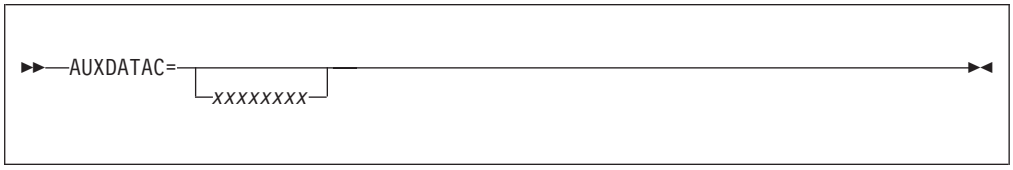
AUDSUNIT
Specifies the unit of primary and secondary space to be allocated to the File Manager Audit Log.
REC Record of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

AUDUNIT



AUDUNIT
Specifies the default permanent unit that File Manager uses when allocating the File Manager Audit Log. The maximum length you can specify for AUDUNIT is eight (8) bytes. AUDUNIT is optional. If omitted the SMS storage class specified in PUNIT is used.

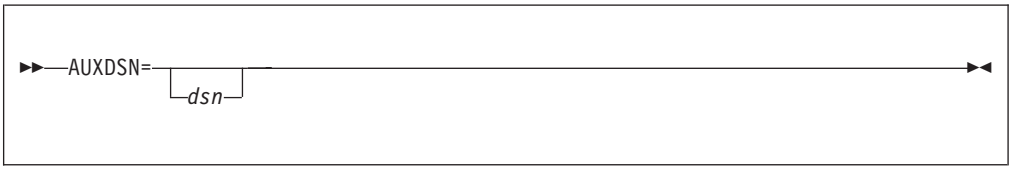
AUXDATAC



AUXDATAC
Specifies the default SMS data class that File Manager uses when allocating the auxiliary data set for a full function edit session. AUXDATAC is

optional. The maximum length you can specify for AUXDATAC is eight (8) bytes. If not supplied, no SMS data class will be specified when allocating the auxiliary data set.

AUXDSN

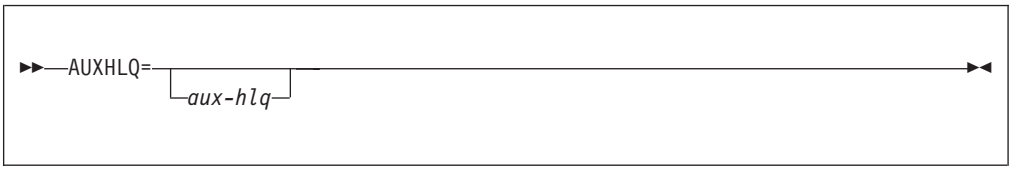


AUXDSN
Specifies an installation-specific data set name to be used as the auxiliary file for a full function edit session.

You can specify any valid name for this data set. If you specify &&USER. as a high-level qualifier for AUXDSN, the data set name has a high-level qualifier of the user's userid. If you specify &&PREFIX. as the high-level qualifier for AUXDSN, the data set name has a high-level qualifier of the user's TSO PREFIX.

The auxiliary file must be a REUSEable RRDS file, whose record length definitions are compatible with the file being edited. If this field is left blank, an RRDS file is defined for the edit session and subsequently deleted.

AUXHLQ



AUXHLQ
Specifies an installation-specific high-level qualifier to be used in the name of the auxiliary data set for a full function edit session.

You can specify any valid qualifier, including multilevel if required, up to a total length of 24 characters. If you specify &&USER. for any part of the qualifier in your FMN0POPT macro, this is replaced with the user's userid. If you specify &&PREFIX. for any part of the qualifier, this is replaced with the user's TSO prefix.

&&USER. and &&PREFIX. can also be concatenated with a character string. A multilevel qualifier with symbols follows the same rules as in the specification of a data set name with symbolic parameters in JCL procedures. Thus, a single period used after a symbol combines the symbol with the text following it (See Example 1 below); a period separating parts of a multilevel qualifier must be doubled (see Example 2 below).

Example 1

If the userid is XXXX and the TSO prefix is YYYY, then specifying

```
&&USER.1.TEMP.&&PREFIX
```

results in a high-level-qualifier of
```
XXX1.TEMP.YYYY
```

Example 2

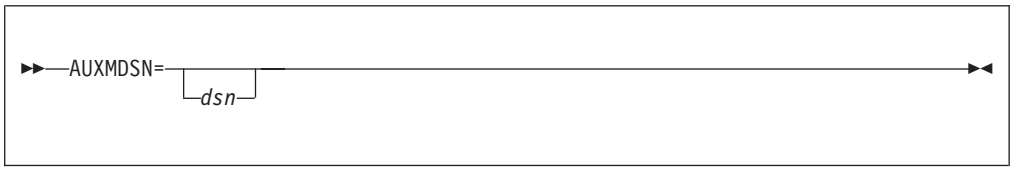
If the userid is ZZZZ then specifying
```
FMN.&&USER..ABCD
```

results in a high-level-qualifier of
```
FMN.ZZZZ.ABCD
```

Notes:

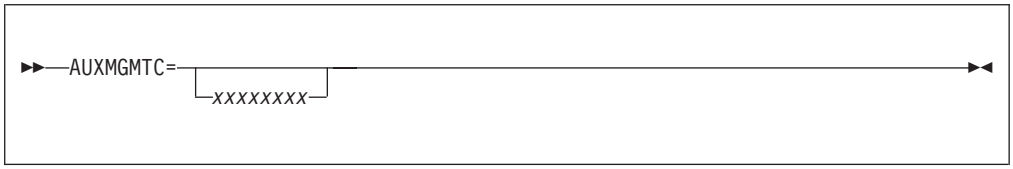
1. The AUXHLQ parameter can be up to 24 characters long. However, temporary data sets can have different lengths, and the system part of the data set name may be longer than 20 characters. In this case, minor levels of AUXHLQ are ignored, to the extent that the final data set name is less than or equal to 44 characters.
2. Concatenation of &&USER.string, or &&PREFIX.string, could result in a part of the high-level qualifier being longer than eight characters. In this case, the string is shortened to the extent that the final part of the qualifier is eight characters long.

AUXMDSN



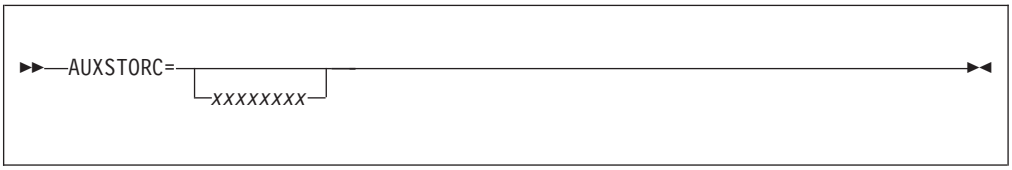
AUXMDSN
Specifies the data set name to be used as the model file when defining an the auxiliary file for a full function edit session. A model file should be used to determine the volume placement of the auxiliary data set in a non-SMS environment. The model file must be a VSAM cluster.

AUXMGMTC



AUXMGMTC
Specifies the default SMS management class that File Manager uses when allocating the auxiliary data set for a full function edit session. AUXMGMTC is optional. The maximum length you can specify for AUXMGMTC is eight (8) bytes. If not supplied, no SMS management class will be specified when allocating the auxiliary data set.

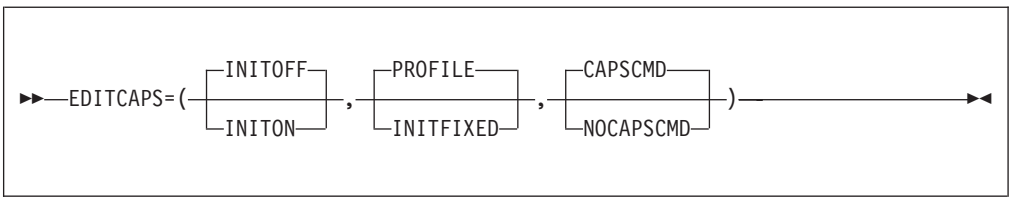
AUXSTORC



AUXSTORC

Specifies the default SMS storage class that File Manager uses when allocating the auxiliary data set for a full function edit session. AUXSTORC is optional. The maximum length you can specify for AUXSTORC is eight (8) bytes. If not supplied, no SMS storage class will be supplied when allocating the auxiliary data set.

EDITCAPS



EDITCAPS

Controls the setting of the CAPS option in edit. A number of options are provided, which allow you to:

- Set the initial value (either CAPS ON or CAPS OFF) for new File Manager users.
- Force all users of the File Manager editor to commence an edit session with either CAPS ON or CAPS OFF; alternatively to allow the user to control the initial CAPS setting using a new editor option, that is saved in the user's ISPF profile.
- Disable the CAPS command within a File Manager edit session.

  You can use certain settings of the EDITCAPS parameters to create an environment where every File Manager edit session starts with CAPS ON, and this cannot be altered using the CAPS OFF command.

There are three sub-parameters to EDITCAPS. You do not have to specify them all in FMN0POPT, but if you do so, they must be in the order shown in the syntax diagram. If you specify more than one, they must be separated by commas. If you specify any sub-parameters, you must also provide the parentheses.

The default is `(INITOFF,PROFILE,CAPSCMD)`.

These sub-parameters are described below.

INITOFF, INITON

Determines the initial CAPS setting in the user's profile, when the profile is first created.

For users with an existing File Manager profile, the effect depends on the setting of the second parameter (described below). If INITFIXED is specified for the second parameter, then every File

Manager edit session begins with either CAPS OFF if INITOFF is specified, or CAPS ON if INITON is specified.

PROFILE, INITFIXED
Determines if the user can alter the initial setting of CAPS in File Manager edit. INITFIXED prevents the user from altering the initial setting of CAPS. PROFILE allows the user to change this initial setting and it is preserved between File Manager sessions in the user's profile.

CAPSCMD, NOCAPSCMD
Determines if the user is permitted to use the CAPS command within a File Manager edit session. NOCAPSCMD does not affect the CASE command, which remains available for use.

When CAPSCMD is specified, the user can turn CAPS on or off in a File Manager edit session, as required. When NOCAPSCMD is specified, the CAPS command is disabled, and is fixed at whatever setting was selected when the File Manager edit session commenced.

COBDBCS

COBDBCS
Specifies if the **DBCS** COBOL compiler option is to be specified when compiling COBOL copybooks.
YES The **DBCS** COBOL compiler option is specified
NO The **DBCS** COBOL compiler option is not specified.

COBDPC

COBDPC
Specifies if the **Decimal-point is comma** SPECIAL-NAMES paragraph is included when compiling COBOL copybooks.
YES The paragraph is included.
NO The paragraph is not included.

COBEXTND

COBEXTND

Specifies if the **Arith(extend)** COBOL compiler option is to be specified when compiling COBOL copybooks.

YES The **Arith(extend)** COBOL compiler option is specified.

NO The **Arith(extend)** COBOL compiler option is not specified.

COBMAXRTN

COBMAXRTN

Specifies the maximum warning or error code received from the COBOL compiler, when compiling a copybook. Any code higher than specified stops the current File Manager function. You can specify a value from 0 to 99. The default value is 4.

COBMCASE

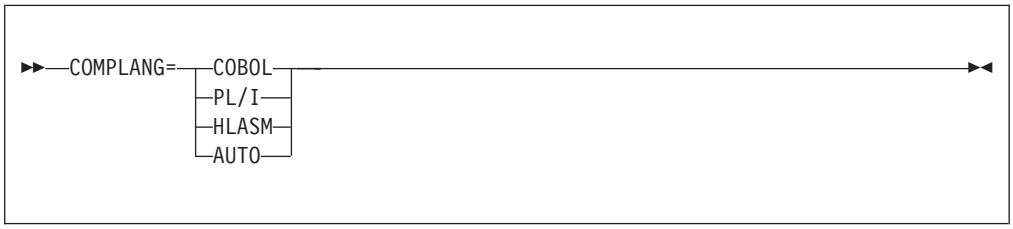
COBMCASE

Specifies if the original case of the field name as coded in the COBOL copybooks is to be retained. This feature is only available if you are running with the File Manager COBOL compiler or a minimum compiler level of Enterprise COBOL V4R1.

YES The original case of the field name as coded is to be retained.

NO The field name is translated to uppercase.

COMPLANG



COMPLANG
Specifies the language which is to be used for compiling a source member to create a template.

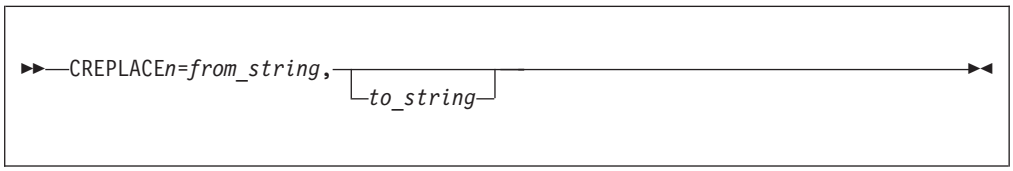
COBOL
Use the COBOL compiler

PL/I Use the PL/I compiler

HLASM
Use the HLASM compiler

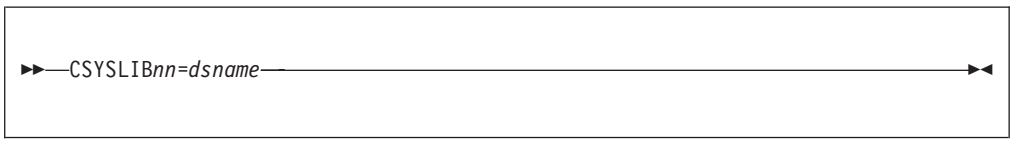
AUTO
File Manager analyzes the source to determine whether the language of the source is COBOL or PL/I and creates a template based on the result. It does not recognize HLASM source.

CREPLACE*n*



CREPLACE*n*
Where *n* is a number in the range 1 to 5, specifies the from pseudo-text character string that you want to be replaced by another character string in your COBOL copybooks via a COBOL REPLACE statement. The *from_string* must be 1 to 30 characters in length. The *to_string* is optional. If supplied it must be 1 to 30 characters in length. If it is not supplied a comma must still be used to delimit the *from* string.

CSYSLIB*nn*



CSYSLIB*nn*
Where *nn* is a number in the range 1 to 10, specifies the names of SYSLIB data sets which are searched in the order specified for COPY or INCLUDE members for the COBOL compilation.

CYLHD

CYLHD
Defines the way you specify the location of a disk data set in disk functions.

ABSOLUTE
File Manager interprets cylinder-head and track-number values as actual physical addresses.

RELATIVE
File Manager interprets cylinder-head and track-number values as relative to the start of the data set.

Note: Changing the CYLHD option has no effect for FM/DB2, FM/IMS, or FM/CICS.

DATAHDR

DATAHDR
Specifies whether record header information (that is, record number and length) is added to each record printed.

YES Record header information is included in the output.

NO Record header information is not included in the output.

Notes:

1. This option affects only DSP and FCH processing.
2. This option affects only output in character format. When printing in hexadecimal format, the output always includes a data header.

DSPINC

DSPINC
Specifies the amount (in KB) by which a clipboard will expand when it becomes full. Default is 2048 (2MB). Maximum is 4194303 (4GB).

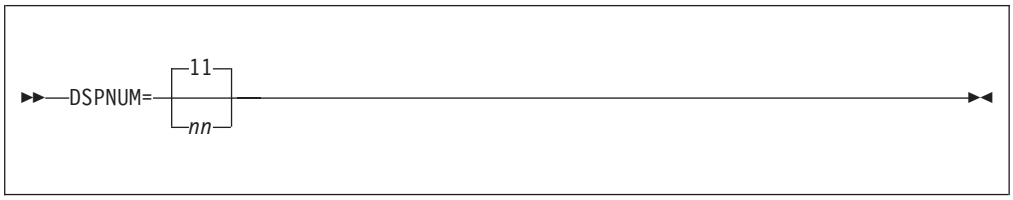
DSPMAX

DSPMAX
Specifies the maximum size in KB, of any one clipboard. Default is 2097152 (2GB). Maximum is 4194303 (4GB). Note that dataspace limits may be otherwise determined by the installation IEFUSI system exit. Refer to *z/OS MVS Installation Exits*, SA22-7593, for more information about system exits.

DSPMIN

DSPMIN
Specifies the starting size in KB of clipboards created by FM. Default is 1024 (1MB). Maximum is 4194303 (4GB).

DSPNUM



DSPNUM
Specifies the maximum number of clipboards that FM can create. The default, and maximum, is 11.

DUMP

DUMP
Specifies the format of hexadecimal print output (for example, when you use Tape Print with **Print format** set to HEX).

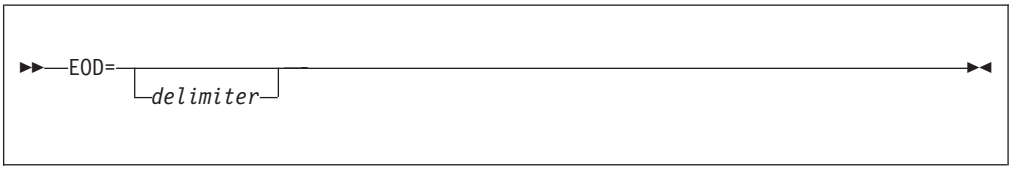
UPDOWN
The two digits making up the hexadecimal representation of each EBCDIC character are displayed vertically directly under that character. This is the File Manager dump format.

ACROSS
The hexadecimal digits are displayed as 2 groups of 4 fullwords resulting in 32 hexadecimal digits followed by the EBCDIC character display to the right of the hexadecimal display. This corresponds to the system dump format.

EDMAXVIRT

EDMAXVIRT
Specifies the limit in KB, that an editor session should use. The default is 0, meaning it is limited only by available region. Note that the editor generally allocates virtual storage in 512KB blocks, so the actual usage may be rounded up to the next 512KB block. Specifying a value larger than the available virtual storage has no net effect on the editors virtual storage usage. Maximum is 4194303 (4GB).

EOD



EOD Specifies the end-of-data delimiter. Must be between one and eight characters. For character data, enclose the string in quotation marks if it contains blanks, commas, or lowercase letters. For hexadecimal data, enter an X followed by the string enclosed in quotation marks (for example, X'04').

FMEDITOR

FMEDITOR

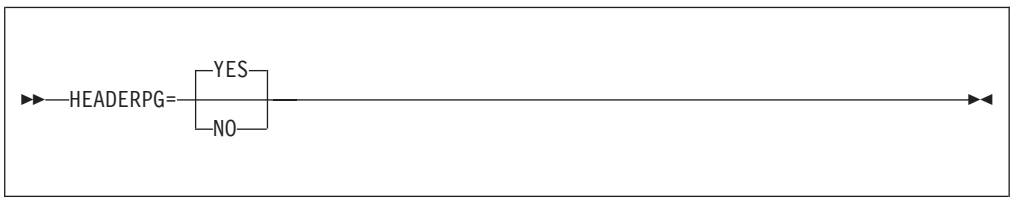
Specifies which editor is to be used when viewing or browsing a member from a member selection panel or when viewing generated output.

YES The File Manager editor is used.

NO The ISPF editor is used.

Note: In a CICS environment, only the File Manager editor is available.

HEADERPG



HEADERPG

Specifies whether a header page (a title page with File Manager on it) is included in print output. A header page does not appear on output routed to the terminal, even if HEADERPG=YES is set.

YES A header page is included in the print output.

If you specify YES then the output will also show the File Manager installed components, version and PTF level. This is the same information as is produced by the $$FILEM VER control statement.

NO A header page is not included in the print output.

Note: Changing the HEADERPG option has no effect for FM/DB2, FM/IMS, or FM/CICS.

HLDBCS

HLDBCS

Specifies if the **DBCS** HLASM compiler option is to be specified when compiling assembler copybooks.

YES The **DBCS** HLASM compiler option is specified
NO The **DBCS** HLASM compiler option is not specified.

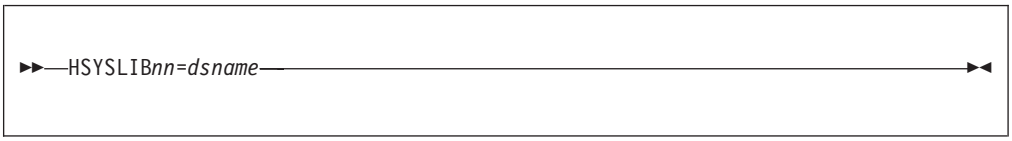
HLMAXRTN

HLMAXRTN
Specifies the maximum warning or error code received from the HLASM compiler, when compiling a copybook. Any code higher than specified stops the current File Manager function. You can specify a value from 0 to 99. The default value is 4.

HLNOALIGN

HLNOALIGN
Specifies if the **NOALIGN** HLASM compiler option is be specified when compiling assembler copybooks.
YES The **NOALIGN** HLASM compiler option is specified.
NO The **NOALIGN** HLASM compiler option is not specified.

HSYSLIB*nn*



HSYSLIB*nn*
Where *nn* is a number in the range 1 to 10, specifies the names of SYSLIB data sets which are searched in the order specified for COPY members for the HLASM compilation.

ISPFPACK

ISPFPACK

Specifies whether File Manager is to check the data set or member to determine if it has been written with the ISPF PACK option.

YES File Manager is to check the data set or member to determine if it has been written with the ISPF PACK option. If so, File Manager then unpacks the data to allow it to be edited or viewed in the normal fashion. If the data set is too large to be contained in memory, File Manager is not able to edit or view the data set in unpacked form, but instead provides the data set in its packed form.

NO File Manager does not check the data and the editor operates on the record data regardless of the data being in ISPF pack form or not.

LANGUAGE

LANGUAGE

Specifies the language for File Manager batch message text. The possible values depend on which translated message texts you installed. For information on installing translated messages for the various File Manager components, see the relevant chapter on customizing for national languages. See also the following notes.

ENGLISH

Messages will appear in English.

language

Messages will appear in the language of your choice.

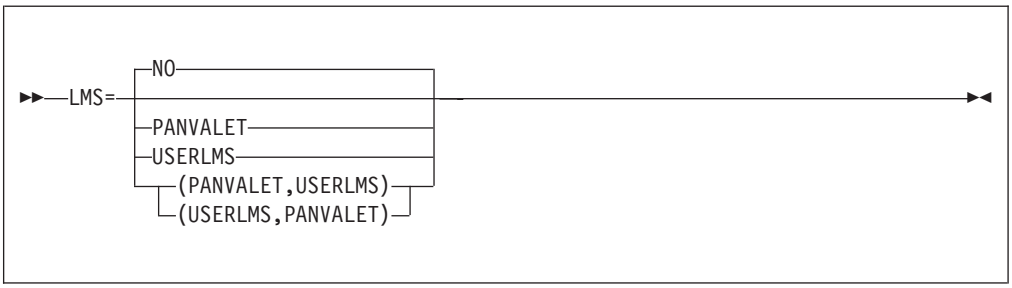
Notes:

1. See Table 15 on page 64 to determine what value to use for *language* for your requirements.
2. Specifying a language here only affects batch processing. The language used by File Manager under ISPF depends on the language setting for your ISPF session.

   However, FM/DB2 also uses the value of *language* to determine the data module to use when displaying DB2 catalog table information. See "OP34MOD" on page 354 for more information.

3. If you specify a language other than English, you must also customize the translation tables, the message modules, and (in some cases) the ISPF messages for your chosen language. See Chapter 7, “Customizing File Manager for national languages,” on page 61, or Chapter 17, “Customizing FM/DB2 for national languages,” on page 151, or Chapter 25, “Customizing FM/IMS for national languages,” on page 213, or Chapter 34, “Customizing FM/CICS for national languages,” on page 269, for information about customizing for national languages.
4. Specifying LANGUAGE=*language*, causes the relevant language translate tables and message tables to be loaded for batch processing.
5. You must specify LANGUAGE=JAPANESE (or another DBCS-capable language) to print DBCS characters in batch.
6. You must specify LANGUAGE=JAPANESE (or another DBCS-capable language) for batch processing, if you want to use the FMT function, and field type of DB, to describe COBOL or PL/I Graphic fields.

LMS



LMS Specifies whether or not File Manager will use COBOL copybooks, PL/I include books, or HLASM copybooks stored in CA-Panvalet libraries or other Library Management System libraries.

NO File Manager will not use any library management system libraries.

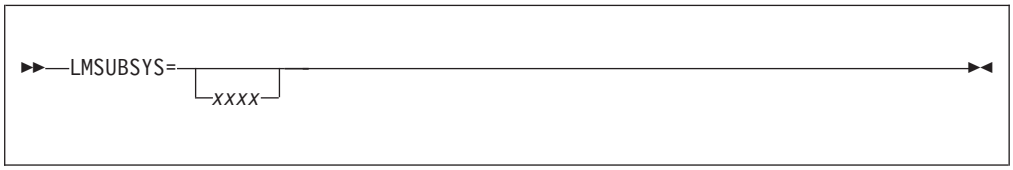
PANVALET
File Manager will use copybooks or include books stored in CA-Panvalet libraries.

USERLMS
File Manager will use copybooks or include books stored in a library management system, other than CA-Panvalet, that does **not** provide a SUBSYS interface.

(PANVALET,USERLMS) or (USERLMS,PANVALET)
File Manager can use copybooks or include books stored in CA-Panvalet together with any other library management system that does **not** provide a SUBSYS interface. (PANVALET,USERLMS) and (USERLMS,PANVALET) are equivalent.

Note: You cannot specify LMS=PANVALET together with LMSUBSYS=xxxx, since you cannot access source from CA-Panvalet together with source from an LMS that provides a SUBSYS interface.

LMSUBSYS



LMSUBSYS
Specifies that File Manager will, when accessing COBOL copybooks, PL/I include books, or HLASM copybooks, attempt the access via an I/O subsystem using the DFSMSdfp SUBSYS=xxxx allocation parameter.

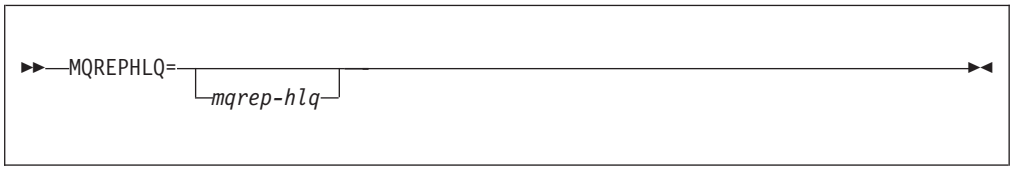
This access is only available to copybooks and include books that exist in data sets controlled by the LMS. Data set organizations of BDAM, VSAM, and PS (sequential) are supported.

This facility is used to access copybooks or include books in OEM library management system data sets.

Specify *xxxx* as provided by your library management system vendor.

Note: You cannot specify LMSUBSYS=xxxx if you have specified LMS=PANVALET, LMS=(PANVALET,USERLMS), or LMS=(USERLMS,PANVALET), since you cannot specify a combination of LMS and LMSUBSYS that equates to using CA-Panvalet with another LMS that provides a SUBSYS interface.

MQREPHLQ



MQREPHLQP
Specifies an installation-specific high-level qualifier to be used in the name pattern of the dynamic reply queue for Websphere MQ related functions. The specific queue name used is returned by the Queue Manager being used at the time.

mqrep-hlq
You can specify any valid qualifier, including multilevel if required, up to a total length of 38 characters. If you specify &&USER. for any part of the qualifier in your FMN0POPT macro, this is replaced with your userid. If you specify &&PREFIX. for any part of the qualifier, this is replaced with your TSO prefix. Note that the current userid is always appended to this high-level qualifier when requesting a dynamic queue.

Also note that the characters '#' (X'7C'), '@' (X'7B') and '$' (X'5B'), while allowed in userids, are not permitted in Websphere MQ. Although specifying them in this parameter is accepted, they are changed at runtime to the allowable MQ character '_'.

&&USER. and &&PREFIX. can also be concatenated with a character string. A multilevel qualifier with symbols follows the same rules as in the specification of a data set name with symbolic parameters in JCL procedures. Thus, a single period used after a symbol combines the symbol with the text following it (see Example 2 below). A period separating parts of a multilevel qualifier must be doubled (see Example 3 below).

Example 1

If the userid is XXXX, specifying the value FMNTMQL. results in a dynamic name request for the pattern:

FMNTMQL.XXXX.*

Example 2

If the userid is XX#XX and the TSO prefix is YYYY, then specifying &&USER.1.TEMP.&&PREFIX results in a dynamic name request for the pattern:

XX_XX1.TEMP.YYYY.XX_XX.*

Example 3

If the userid is ZZZZ, then specifying FMN.&&USER..ABCD. results in a dynamic name request for the pattern:

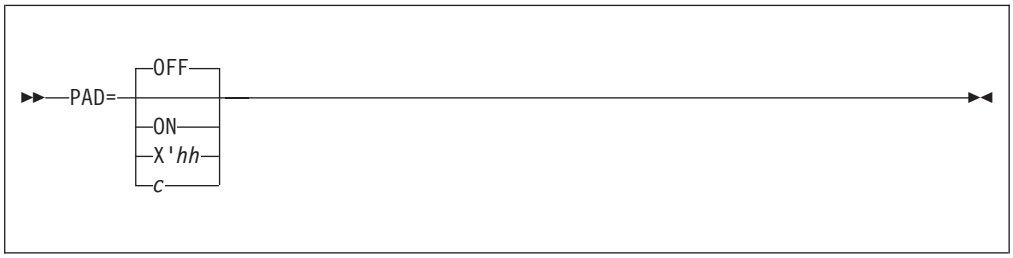
FMN.ZZZZ.ABCD.ZZZZ.*

Note: Concatenation of &&USER.string, or &&PREFIX.string, could result in a part of the high-level qualifier being longer than eight characters. In this case, the string is shortened to the extent that the final part of the qualifier is eight characters long.

MSGUPPER

MSGUPPER
Specifies that all message text is to be translated to uppercase.
NO No messages are translated to uppercase.
YES All messages are translated to uppercase.

PAD



PAD Specifies whether records are padded or truncated when being copied.

OFF Records are not padded or truncated.

ON Records are padded with blanks, or truncated.

c or **X'***hh***'**
The same as PAD=ON except that the specified alphanumeric or hexadecimal character is used as the pad character instead of a blank.

PAGESIZE

PAGESIZE
Specifies the number of lines printed on each page of print output. You can specify a value from 1 to 999. The default is 60.

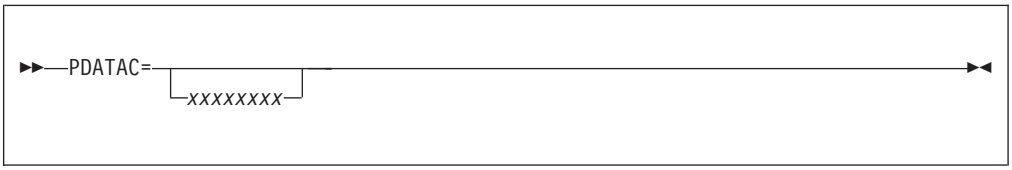
PAGESKIP

PAGESKIP
Specifies whether print output from each function begins on a new page.

NO Print output from each function does not begin on a new page.

YES Print output from each function begins on a new page.

PDATAC



PDATAC

Specifies the default temporary SMS data class that File Manager will use when allocating work data sets for internal use. PDATAC is optional. The maximum length you can specify for PDATAC is eight (8) bytes.

PLI31DIGIT

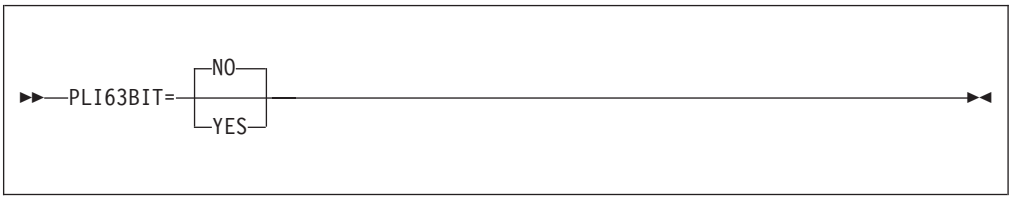
PLI31DIGIT

Specifies if the **LIMITS(FIXEDDEC(31))** PL/I compiler option is to be specified when compiling PL/I copybooks.

YES The **LIMITS(FIXEDDEC(31))** PL/I compiler option is specified

NO The **LIMITS(FIXEDDEC(31))** PL/I compiler option is not specified.

PLI63BIT



PLI63BIT

Specifies if the **LIMITS(FIXEDBIN(63))** PL/I compiler option is to be specified when compiling PL/I copybooks.

YES The **LIMITS(FIXEDBIN(63))** PL/I compiler option is specified

NO The **LIMITS(FIXEDBIN(63))** PL/I compiler option is not specified.

PLIGRAPHIC

PLIGRAPHIC
Specifies if the **GRAPHIC** PL/I compiler option is to be specified when compiling PL/I copybooks.
YES The **GRAPHIC** PL/I compiler option is specified.
NO The **GRAPHIC** PL/I compiler option is not specified.

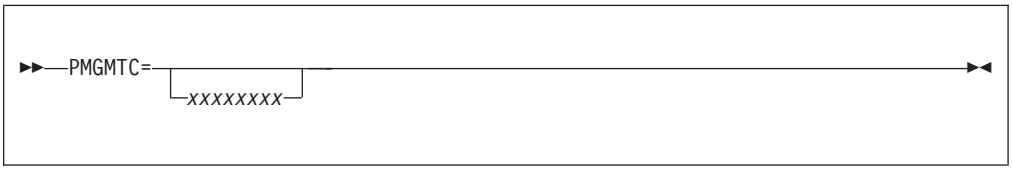
PLIMAXRTN

PLIMAXRTN
Specifies the maximum warning or error code received from the PL/I compiler, when compiling a copybook. Any code higher than specified stops the current File Manager function. You can specify a value from 0 to 99. The default value is 4.

PLIUNALIGN

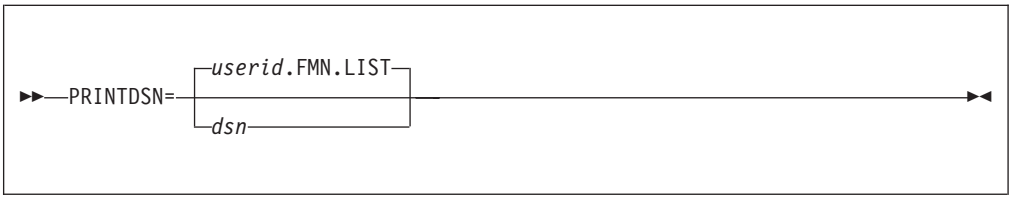
PLIUNALIGN
Specifies if the statement **DEFAULT RANGE(*) UNALIGNED;** is to be added when compiling PL/I copybooks.
YES The statement is to be added.
NO The statement is not to be added.

PMGMTC



PMGMTC
Specifies the default permanent SMS management class that File Manager will use when allocating permanent data sets, (for example, a print file). PMGMTC is optional. The maximum length you can specify for PMGMTC is eight (8) bytes.

PRINTDSN



PRINTDSN
Specifies the name of the temporary print data set where File Manager print output is directed when PRINTDSN is specified for the **PRINTOUT** option.

You can specify any valid name for this data set. If you specify &&USER. as a high-level qualifier for PRINTDSN, the data set name will have a high-level qualifier of the user's userid. If you specify &&PREFIX. as the high-level qualifier for PRINTDSN, the data set name will have a high-level qualifier of the user's TSO PREFIX.

To view the contents of the print data set, use the PB command (Print Browse), or option 3.9. For information on Print Browse, see "Printing from File Manager" in the *File Manager User's Guide and Reference*.

PRINTLEN

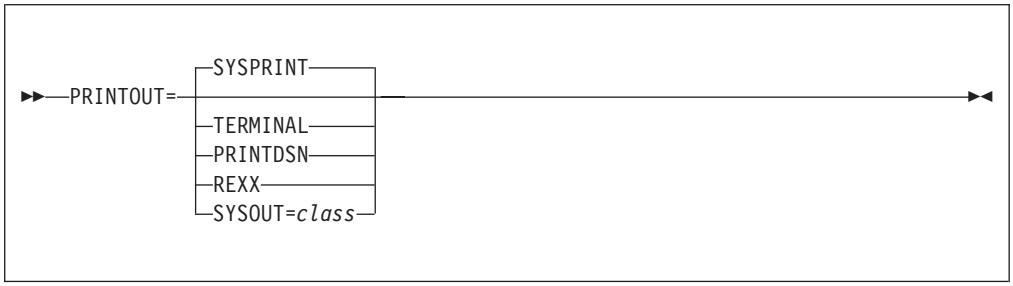
PRINTLEN
Specifies the line length of printed output.

132 The line length of printed output is 132 characters, suitable for a line printer.

80 The line length of printed output is 80 characters, suitable for a terminal.

PRINTLEN is not applicable when the data is printed in TABL format. When data is printed in TABL format, the line length is determined from the number and size of the fields printed. If this length is greater than the record length specified for the data set, the print line is truncated. The maximum print line length is 32760, unless limited to 132 characters by setting WIDEPRT=NO. For information about WIDEPRT, see "WIDEPRT" on page 329.

PRINTOUT



PRINTOUT
Specifies where print output is sent, except for batch execution.

SYSPRINT
Send print output to the current SYSPRINT allocation.

PRINTDSN
Send print output to the temporary data set specified by the PRINTDSN option (see below).

TERMINAL
Send print output to the terminal.

REXX Send print output to the REXX stem variable FILEM.

SYSOUT=*class*
Send print output to the temporary data set specified by the PRINTDSN option (see below) and direct output generated by the PRINT Primary command to the JES spool queue class *class*. (It is recommended that you do not use this parameter. Instead, use the PRINTDSN parameter together with the PRTCLASS.)

PRTCLASS

PRTCLASS
Specifies the class of the JES spool queue to be used when the PRINT primary command is issued.

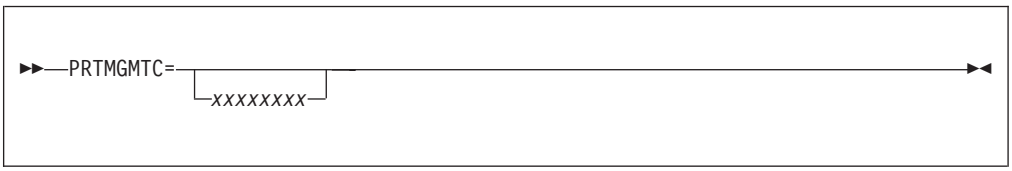
PRTDISP

PRTDISP
Specifies the disposition of a print data set.

MOD Print output is appended to the print data set.

OLD A print data set is cleared before each print operation, and print output is written from the beginning of the data set.

PRTMGMTC



PRTMGMTC
Specifies the default SMS management class that File Manager uses when allocating the File Manager print data set. The maximum length you can specify for PRTMGMTC is eight (8) bytes. PRTMGMTC is optional. If omitted the SMS management class specified in PMGMT is used, if supplied.

PRTPQTY

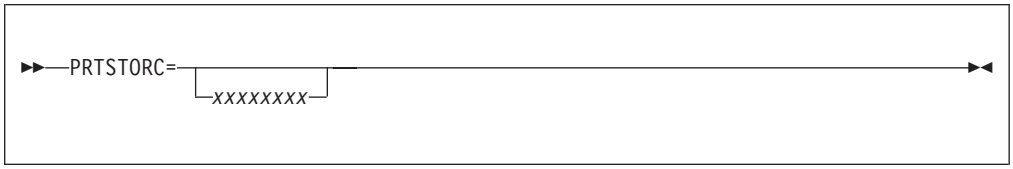
PRTPQTY
Specifies the amount of DASD space to be used for primary space allocation of the File Manager print data set. The range depends on the space unit specified and the DASD device type.

PRTSQTY

PRTSQTY

Specifies the amount of DASD space to be used for secondary space allocation of the File Manager print data set. The range depends on the space unit specified and the DASD device type.

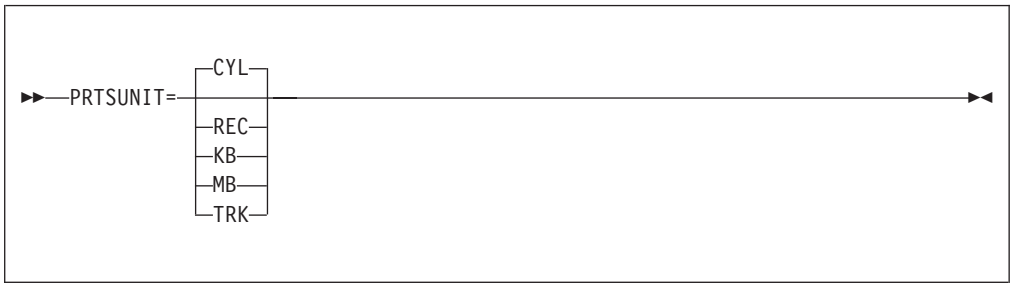
PRTSTORC



PRTSTORC

Specifies the default SMS storage class that File Manager uses when allocating the File Manager print data set. The maximum length you can specify for PRTSTORC is eight (8) bytes. PRTSTORC is optional. If omitted the SMS storage class specified in PSTORC is used, if supplied.

PRTSUNIT



PRTSUNIT

Specifies the unit of primary and secondary space to be allocated to the File Manager print data set.

REC Record of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

PRTTRANS

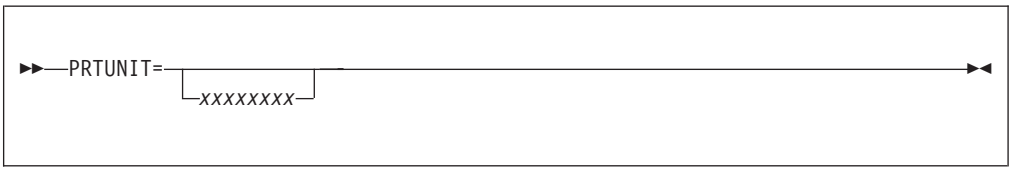
PRTTRANS
Specifies how File Manager should translate non-printable characters.

ON Non-printable characters are translated to blanks using a translate table. This may make printing faster.

OFF No translation is performed. Use PRTTRANS=OFF to support special print chains.

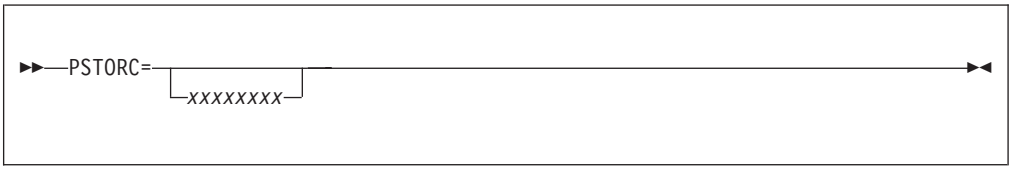
For information on altering the print translate table, see "Changing the print and display translation tables" on page 28 and "Changing the print and display translation tables for languages other than English" on page 61.

PRTUNIT



PRTUNIT
Specifies the default permanent unit that File Manager uses when allocating the File Manager print data set. The maximum length you can specify for PRTUNIT is eight (8) bytes. PRTUNIT is optional. If omitted the SMS storage class specified in PUNIT is used.

PSTORC

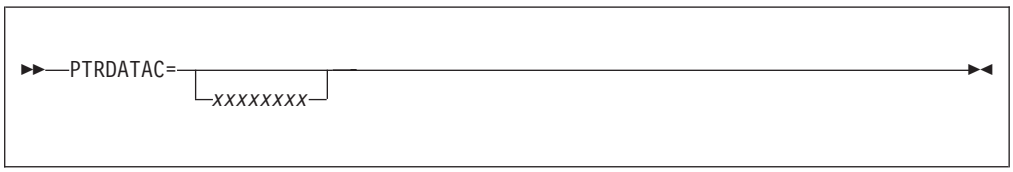


PSTORC
Specifies the default permanent SMS storage class that File Manager will use when allocating permanent data sets, (for example, a print file). PSTORC is optional. The maximum length you can specify for PSTORC is eight (8) bytes.

PSYSLIB*nn*

PSYSLIB*nn*
Where *nn* is a number in the range 1 to 10, specifies the names of SYSLIB data sets which are searched in the order specified for %INCLUDE members for the PL/I compilation.

PTRDATAC



PTRDATAC
Specifies the SMS data class that File Manager uses when allocating the File Manager print data set. The maximum length you can specify for PRTDATAC is eight (8) bytes. PTRDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied.

PUNIT

PUNIT
Specifies the default permanent unit that File Manager will use when allocating permanent data sets, (for example, a print file). The default is SYSALLDA. The maximum length you can specify for PUNIT is eight (8) bytes.

RECLIMIT

RECLIMIT
For data printed in CHAR or LHEX print format, limits print output for each record (or OAM object).

(1,*) The entire record (or block) is printed.

(*n*,*m*) Print output is limited to the data from columns (bytes) *n* through *m*. An asterisk (*) specified for *m* indicates the end of the record.

Note: Changing the RECLIMIT option has no effect for FM/DB2, FM/IMS, or FM/CICS.

SEC

SEC Specifies whether or not update functions are to be protected in File Manager base function or in FM/CICS.
NO Update functions are not protected.
YES Update functions can be protected.

Note: To protect update functions in File Manager base function set SEC=YES in FMN0POPT. To protect update functions in FM/DB2 set SEC=YES in FMN2POPT. To protect update functions in FM/CICS set SEC=YES in FMN3POPT. Setting this option in FMN1POPT has no effect in FM/IMS.

For more information about protecting update functions, see "Unprotected functions and profile names for protected functions" on page 44.

SFMNMOD1

SFMNMOD1
The data set name of the File Manager load library which is used in the FMNFTEXC and FMN1FTEX job control skeletons.

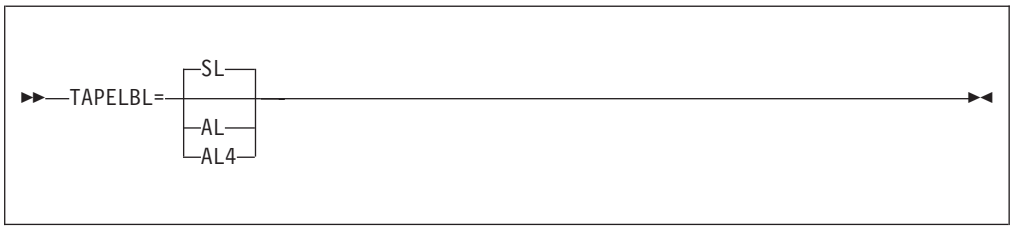
SMFNO

SMFNO
Specifies the SMF record type for audit logging. You can specify a value from 128 to 255, or 0. The default is 0.

Notes:

1. Specifying SMFNO=0 indicates that audit logging will not create SMF records.
2. If you specify a value for SMFNO other than zero, and the FMNSMF module is not APF-authorized, you may receive an error message, even if you did not intend any audit logging to take place.
3. File Manager base only uses the value specified for this parameter, when the FMAUDIT parameter in the FMN0PARM member in the SYS1.PARMLIB (or its concatenations) specifies SAF_CTRL=NO.
4. FM/IMS only uses the value specified for this parameter, when the FMAUDIT parameter in the FMN1PARM member in the SYS1.PARMLIB (or its concatenations) specifies SAF_CTRL=NO.
5. FM/DB2 only uses the value specified for this parameter, when the FMAUDIT parameter in the FMN2PARM member in the SYS1.PARMLIB (or its concatenations) specifies SAF_CTRL=NO.
6. FM/CICS only uses the value specified for this parameter, when the FMAUDIT parameter in the FMN3PARM member in the SYS1.PARMLIB (or its concatenations) specifies SAF_CTRL=NO.
7. When the FMAUDIT parameter specifies SAF_CTRL=YES, audit logging is controlled by SAF and File Manager ignores this parameter.

TAPELBL



TAPELBL
Specifies the type of tape labels that are created by the Initialize Tape function.

SL EBCDIC labels are created.

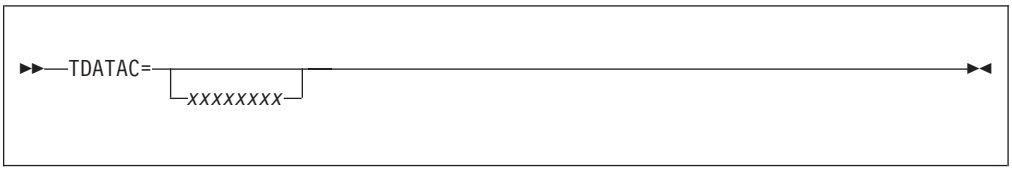
AL ANSI Version 3 labels are created. (“Version 3” refers to ANSI X3.27–1978, ISO 1001–1979, and FIPS 79 standards.)

AL4 ANSI Version 4 labels are created. (“Version 4” refers to ANSI X3.27–1987 level 4 and ISO 1001–1986(E) standards.)

This parameter only affects the Initialize Tape function. For information on Initialize Tape, see "INT (Initialize Tape)" in the *File Manager User's Guide and Reference*.

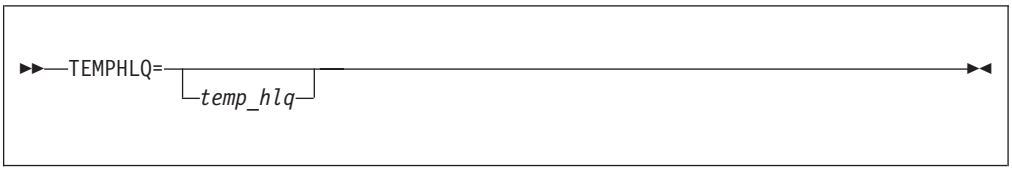
Note: Changing the TAPELBL option has no effect for FM/DB2, FM/IMS or FM/CICS.

TDATAC



TDATAC
Specifies the default temporary SMS data class that File Manager will use when allocating work data sets for internal use. TDATAC is optional. The maximum length you can specify for TDATAC is eight (8) bytes.

TEMPHLQ



TEMPHLQ
Specifies an installation-specific high-level qualifier for the temporary data sets created during a File Manager session.

Some File Manager functions that create temporary data sets, for which the TEMPHLQ value is used if specified, are:
- FM/IMS processing the XKEY command
- FM/IMS creating the IMS LOG data set
- FM/IMS executing IXB (IMS Extract data in batch) function
- FM/IMS processing and building PSBs
- FM/DB2 editing and showing SQL statements

You can specify any valid qualifier, including multilevel if required, up to a total length of 24 characters. If you specify &&USER. for any part of the qualifier in your FMN0POPT macro, this is replaced with the user's userid. If you specify &&PREFIX. for any part of the qualifier, this is replaced with the user's TSO prefix.

&&USER. and &&PREFIX. can also be concatenated with a character string. A multilevel qualifier with symbols follows the same rules as in the specification of a data set name with symbolic parameters in JCL procedures. Thus, a single period used after a symbol combines the symbol with the text following it (See Example 1 below); a period separating parts of a multilevel qualifier must be doubled (see Example 2 below).

Example 1

If the userid is XXXX and the TSO prefix is YYYY, then specifying
`&&USER.1.TEMP.&&PREFIX`

will result in a high-level-qualifier of
`XXX1.TEMP.YYYY`

Example 2

If the userid is ZZZZ then specifying
`FMN.&&USER..ABCD`

will result in a high-level-qualifier of
`FMN.ZZZZ.ABCD`

Notes:

1. The TEMPHLQ parameter can be up to 24 characters long. However, temporary data sets can have different lengths, and the system part of the data set name may be longer than 20 characters. In this case, minor levels of TEMPHLQ will be ignored, to the extent that the final data set name is less than or equal to 44 characters.
2. Concatenation of &&USER.*string*, or &&PREFIX.*string*, could result in a part of the high-level qualifier being longer than eight characters. In this case, the string will be shortened to the extent that the final part of the qualifier is eight characters long.

TERMTYPE

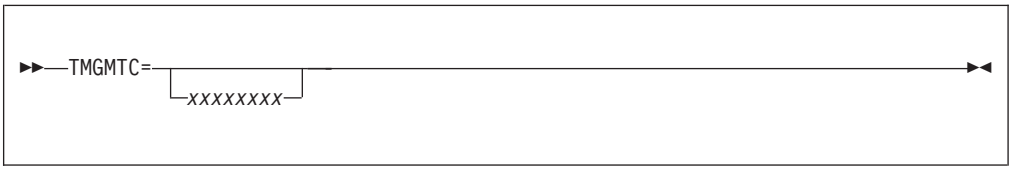
TERMTYPE
Specifies the terminal type.

3270 Specify 3270 for standard 3270 terminals.

3270KN
Specify 3270KN to support terminals that use Japanese Katakana characters or another DBCS language.

If 3270KN is specified, File Manager translates message text to uppercase if you have LANGUAGE=ENGLISH, and translates panel text to uppercase regardless of the language.

TMGMTC

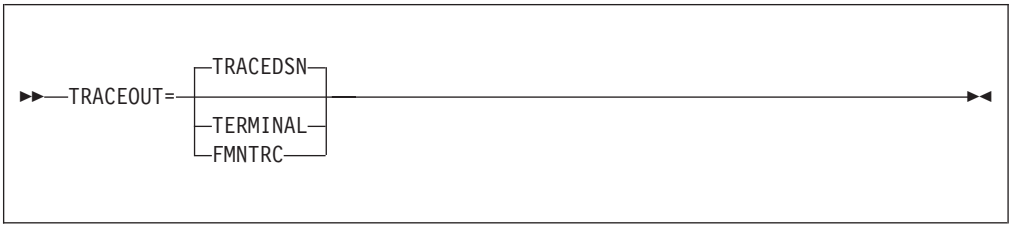


TMGMTC
Specifies the default temporary SMS management class that File Manager will use when allocating work data sets for internal use. TMGMTC is optional. The maximum length you can specify for TMGMTC is eight (8) bytes.

TRACECLS

TRACECLS
Specifies the class of the JES spool queue to be used when the PRINT primary command is issued when browsing the temporary trace data set.

TRACEOUT



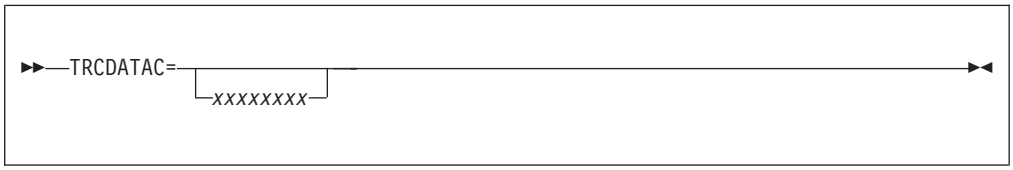
TRACEOUT
Specifies where printed File Manager debug trace output should be sent.

FMNTRC
Debug trace output is sent to the current FMNTRC allocation.

TERMINAL
Debug trace output is sent to the terminal.

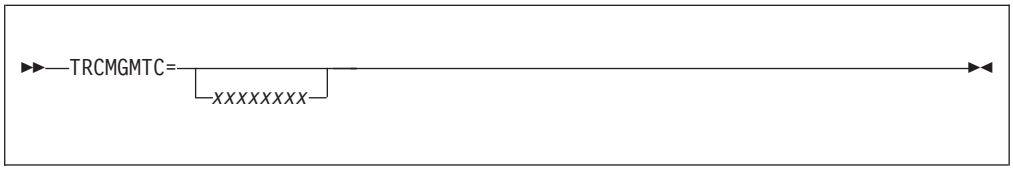
TRACEDIN
Debug trace output is sent to the temporary trace data set specified by the TRACEDSN option (see above).

TRCDATAC



TRCDATAC
Specifies the SMS data class that File Manager uses when allocating the File Manager Trace data set. The maximum length you can specify for TRCDATAC is eight (8) bytes. TRCDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied.

TRCMGMTC



TRCMGMTC
Specifies the default SMS management class that File Manager uses when allocating the File Manager Trace data set. The maximum length you can specify for TRCMGMTC is eight (8) bytes. TRCMGMTC is optional. If omitted the SMS management class specified in PMGMT is used, if supplied.

TRCPQTY

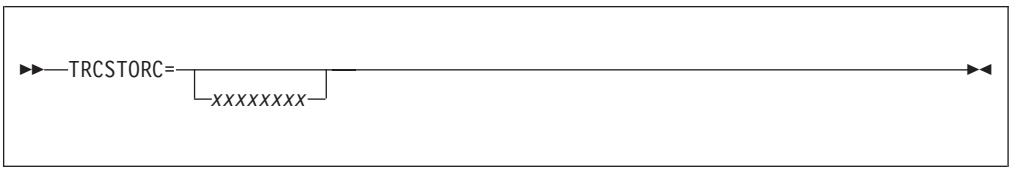
TRCPQTY
Specifies the amount of DASD space to be used for primary space allocation of the File Manager Trace data set. The range depends on the space unit specified and the DASD device type.

TRCSQTY

TRCSQTY

Specifies the amount of DASD space to be used for secondary space allocation of the File Manager Trace data set. The range depends on the space unit specified and the DASD device type.

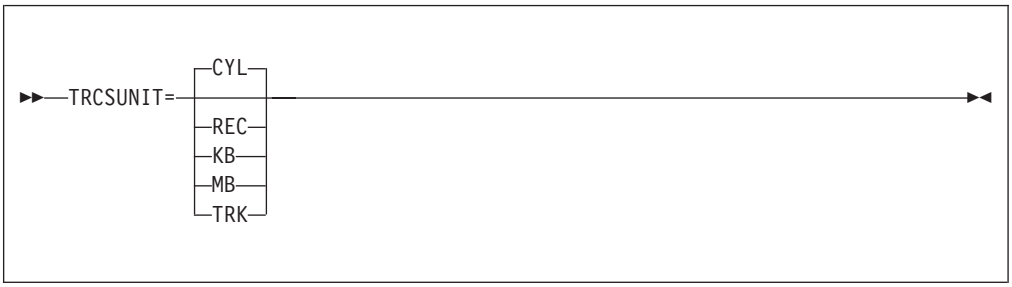
TRCSTORC



TRCSTORC

Specifies the default SMS storage class that File Manager uses when allocating the File Manager Trace data set. The maximum length you can specify for TRCSTORC is eight (8) bytes. TRCSTORC is optional. If omitted the SMS storage class specified in PSTORC is used, if supplied.

TRCSUNIT

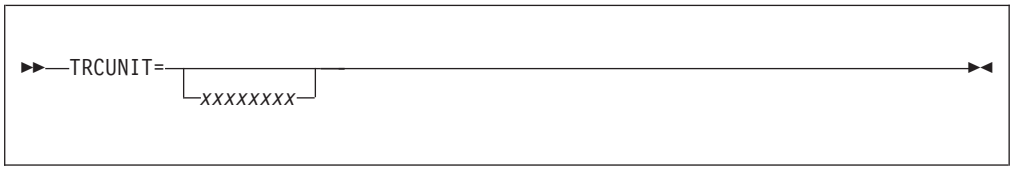


TRCSUNIT

Specifies the unit of primary and secondary space to be allocated to the File Manager Trace data set.

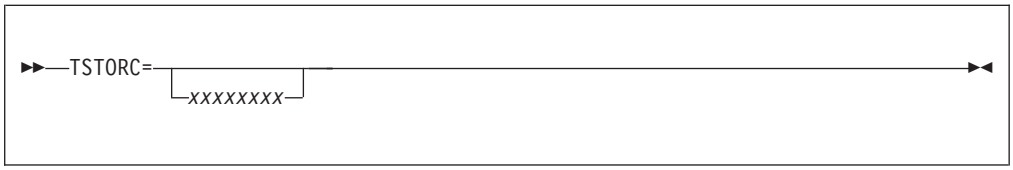
REC Record of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

TRCUNIT



TRCUNIT
Specifies the default permanent unit that File Manager uses when allocating the File Manager Trace data set. The maximum length you can specify for TRCUNIT is eight (8) bytes. TRCUNIT is optional. If omitted the SMS storage class specified in PUNIT is used.

TSTORC

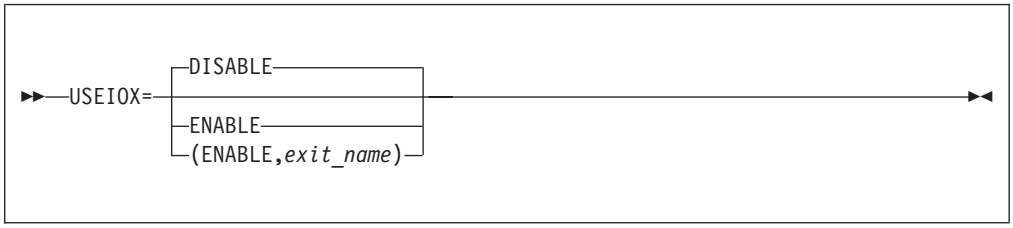


TSTORC
Specifies the default temporary SMS storage class that File Manager will use when allocating work data sets for internal use. TSTORC is optional. The maximum length you can specify for TSTORC is eight (8) bytes.

TUNIT

TUNIT
Specifies the default temporary unit that File Manager will use when allocating work data sets for internal use. The default is SYSALLDA. The maximum length you can specify for TUNIT is eight (8) bytes.

USEIOX



USEIOX
Specifies whether or not File Manager will load an I/O exit For information on providing a user I/O exit, see Chapter 9, “Customizing File Manager to use an I/O exit,” on page 73.

DISABLE
The exit function is disabled. File Manager will not use an I/O exit when processing data sets, and the user will not be able to specify an exit name on any panel. The default is DISABLE.

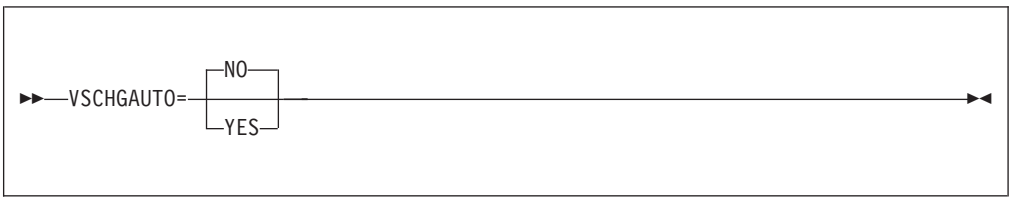
ENABLE
The exit function is enabled. The User I/O Exit selection field is displayed on the relevant panels, allowing the user to provide the name of a user-written exit, if required. Alternatively, the name of an exit can be specified in the system settings.

(ENABLE,*exit_name***)**
The exit function is enabled and a default exit, *exit_name*, will be displayed in the User I/O Exit selection field, unless overwritten by the user. If a value is specified for *exit_name*, then *exit_name* should exist as a module in a load library available to File Manager.

VSAUTO

VSAUTO
Specifies the frequency of periodic SAVEs when operating upon a VSAM shared file. If 0 is specified, File Manager will not automatically issue a SAVE command. If a value is specified, File Manager automatically issues a SAVE command for the file being operated upon after the given number of updates have occurred. You can specify a value from 0 to 99999999. The default value is 0.

VSCHGAUTO



VSCHGAUTO
Specifies if during a CHANGE command with the ALL operand, when a nonzero figure has been used for the VSCHGFRQ option, this determines if an automatic retry of the change is to take place when a record in a VSAM shared file has been updated by another user before the Change process has saved the changed record.

YES File Manager automatically refreshes records that it had attempted to save and re-applies the change update to them. This is the most effective way of using the Change ALL command upon a shared and heavily updated file.

NO File Manager displays a Record Integrity Warning panel for each instance of a record becoming out of date. The user then has the choice to reapply the update, skip the update or abort the Change process at that point.

VSCHGFRQ

VSCHGFRQ

Specifies the frequency of periodic SAVEs of the VSAM shared file during a CHANGE command with the ALL operand. If 0 is specified, File Manager does not automatically issue a SAVE command. If a value is specified, File Manager automatically issues a SAVE command for the file during a CHANGE command with the ALL operand after the given number of updates have occurred. You can specify a value from 0 to 99999999. The default value is 0.

VSSAVE

VSSAVE

Specifies if when operating upon a VSAM shared file, during a CHANGE command with the ALL operand, a SAVE is issued before commencing the CHANGE.

NO File Manager does not issue a SAVE before commenting the CHANGE.

YES File Manager issues a SAVE command which ensures that any updates made to the file so far are saved. During the CHANGE ALL process, when using VSCHGAUTO (and a VSCHGFRQ value), the change process may need to refresh records as it goes (due to concurrent updates by other users). That refresh may discard other updates that had been made in this edit session to records on the file. Leaving the setting at YES, to issue a SAVE before the CHANGE commences, is therefore recommended as it will ensure the state of the records that have been updated.

WIDEPRT

WIDEPRT

Specifies the record length and blocksize of new print output data sets.

YES For to-be-allocated (new) print output data sets (online) or for SYSPRINT allocation (in batch), File Manager will use the maximum record length/blocksize of 32756/32760 and record format of VBA. This is the default and is the same as previous behaviour.

NO For to-be-allocated (new) print output data sets (online) or for SYSPRINT allocation (in batch), File Manager will use the record length/blocksize of 133/137 and record format of VBA.

If the user has pre-allocated the print output data set, as specified in the PRINTDSN parameter, then this data set's record length and record format are not changed. Also note that once the print output data set has been allocated and opened by File Manager, then such data set's attributes may not be changed again, for example, by using a SET command.

If WIDEPRT is set to NO, all printed output exceeding the current record length of the output data set will be truncated without any warning.

Appendix B. FM/DB2 options

This section describes the FM/DB2 options. The syntax described here applies to FMN2SSDM and FMN2POPI. You modify these FM/DB2 options to suit your requirements. See “Changing the default options” on page 126 for information on how to do this.

FMN2SSDM

In particular, before you can use FM/DB2, you must specify the DB2 subsystems, or DB2 data sharing groups, which FM/DB2 will use. To do this you must provide FMN2SSDM macro statements in FMN2POPT. For information about specifying DB2 subsystems and DB2 data sharing groups, and FMN2SSDM, see “Identifying all the DB2 systems that FM/DB2 will access and defining these in the FMN2POPT module (required)” on page 121.

You must code at least one FMN2SSDM macro entry in FMN2POPT. When there is only one entry it should specify SSID=DEFAULT (see the description of the SSID option). When there are multiple entries, the final FMN2SSDM macro entry must specify SSID=DEFAULT. If this entry is omitted an assembler error will occur.

Note: All library names specified in FMN2SSDM must be enclosed in quotation marks, and each set of library names must be further enclosed in parentheses. A maximum of sixteen (16) library names can be specified for one set of library names.

ATTACH

ATTACH

Specifies which attachment facility is to be used to connect to and use the DB2 subsystem defined by SSID, to process SQL statements, commands or instrument facility interface (IFI) calls.

The properties and requirements of the various DB2 attachment facilities are documented in the *DB2 Application Programming and SQL Guide*.

CAF The call attachment facility will be used to connect to and use the DB2 subsystem defined by SSID.

CAF is the default and recommended value. The CAF attachment works concurrently with other DB2 applications that use the DSNELI (DSN) attachment, the DSNALI (CAF) attachment, but not the DSNRLI (RRSAF) attachment.

© Copyright IBM Corp. 2000, 2011

RRSAF

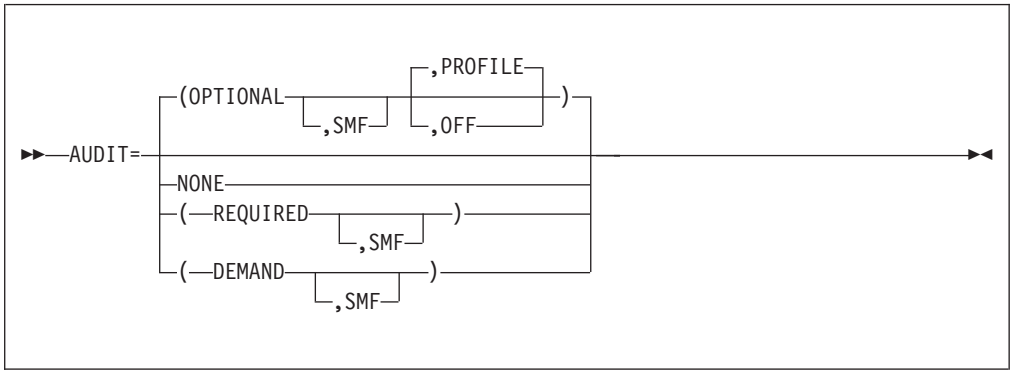
The DB2 Resource Recovery Services attachment facility (RRSAF) will be used to connect to and use the DB2 subsystem defined by SSID.

RRSAF uses z/OS Transaction Management and Recoverable Resource Manager Services (z/OS RRS) and this service must be active.

RRSAF should only be used in situations where there will be no conflicts with other DB2 applications that use other attachment facilities, especially the DSNELI attachment. For example, it is not possible to run DB2's SPUFI and FM/DB2 on separate ISPF split screens when FM/DB2 is using the RRSAF attachment (because of the conflict with the DSN attachment used by DB2's SPUFI).

The default is CAF.

AUDIT



AUDIT

Specifies the audit logging requirements for the DB2 subsystem defined by SSID. Note that this option is ignored when SAF-rule controlled auditing is active.

NONE

Disregards the setting of `Create an audit trail` in the editor options and no audit trail is produced.

OPTIONAL

If `Create an audit trail` in the editor options is selected, an audit trail is produced. If `Create an audit trail` is not selected, no audit trail is produced. The default is **OPTIONAL**.

(OPTIONAL,SMF)

If the suboperand SMF is specified, then if an audit trail is to be produced, it will be written to SMF. You can also optionally specify the suboperands **PROFILE** or **OFF**. In this case, the default is **PROFILE**.

When **PROFILE** is specified, FM/DB2 stores the current value of the `Create an audit trail` setting from the global options in the ISPF profile at the end of every FM/DB2 session.

When **OFF** is specified, FM/DB2 ignores any value for the `Create an audit trail` setting stored in the user's ISPF profile. At the start

of every FM/DB2 session the `Create an Audit trail` option is reset to **OFF**. Changes made to the option using either the global options, or local edit options last for the life of the current FM/DB2 session only. This option may be useful when audit data is written to data sets, rather than SMF, and taking an audit trail is an exceptional, rather than normal activity, as it will minimize the number of audit data sets produced.

REQUIRED
An audit trail is produced, regardless of the setting of `Create an audit trail` in the editor options. If **AUDIT=(REQUIRED,SMF)** is specified, then the audit trail is written to SMF.

DEMAND
An audit trail is produced, regardless of the setting of `Create an audit trail` in the editor options. An audit report job will be submitted at the conclusion of an edit function. If **AUDIT=(DEMAND,SMF)** is specified, then the audit trail is also written to SMF.

Notes:

1. If you specify SMF you must also specify a value for the SMF option in FMN2POPT.
2. The AUDIT option is active only when data is modified via the editor (view, browse or edit) or copy utility. The setting of this option will be ignored for any other File Manager function that allows data to be modified, and for data modified by the ISPF editor, even though this is from within File Manager.

   Auditing can be selectively disabled when using FM/DB2 view or browse by specifying AUDITBROWSE=N in the FMN2SSDM macro definition for a DB2 system. Specifying this option does not affect the audit records produced when using FM/DB2 edit. See “AUDITBROWSE” for information about AUDITBROWSE.

AUDITBROWSE

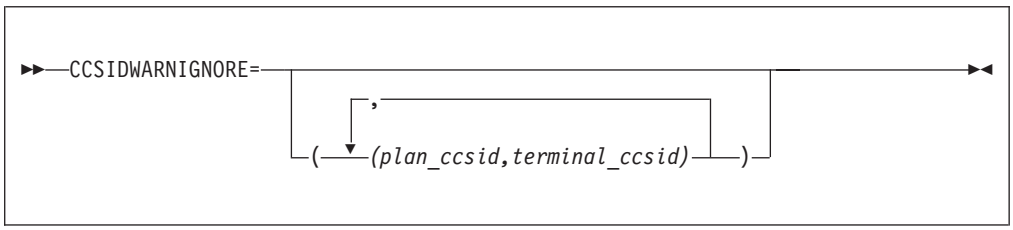
AUDITBROWSE
Specifies whether or not audit log records are to be written for FM/DB2 view or browse sessions. Note that this option is ignored when SAF-rule controlled auditing is active.

Y Specifies that audit log records are to be written for FM/DB2 view or browse sessions. The default is AUDITBROWSE=Y.

N Audit log records are not written for FM/DB2 view or browse sessions.

Specify AUDITBROWSE=N if audit records are not required for users accessing DB2 data using FM/DB2 view or browse.

CCSIDWARNIGNORE



CCSIDWARNIGNORE
Controls whether the CCSID Warning message will be displayed when FM/DB2 connects to a DB2 subsystem, if the CCSID of the terminal differs from the CCSID of the plan. FM/DB2 connects to a DB2 subsystem when the application is first entered or when the DB2 SSID is changed.

This option is used in conjunction with the CCSIDWARN option in the FMN2POPI macro.

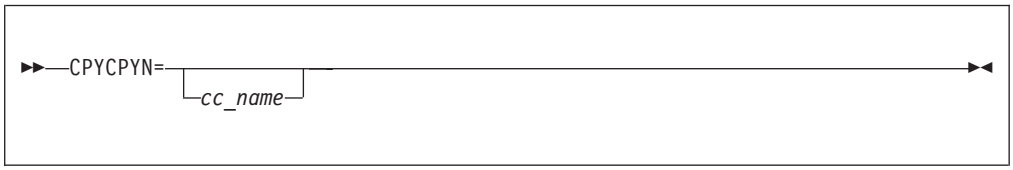
An option is provided to allow you to stop the displaying of the CCSID Warning message when there is a mismatch between the terminal CCSID and the SSID plan CCSID.

Each parameter contains two CCSIDS. These are the CCSID of the plan and the CCSID of the terminal. These CCSIDs are separated by a comma and the whole parameter is enclosed in parentheses.

Up to five pairs of CCSIDs may be specified. If more than one pair is specified the parameters must be separated by a comma and the parameters must be enclosed in an outer pair of parentheses.

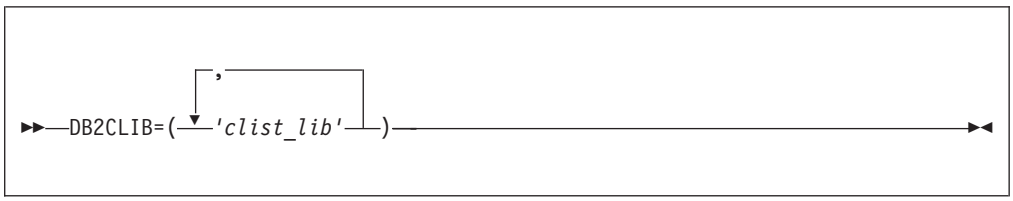
If you do not specify CCSIDWARNIGNORE in your FMN2SSDM macro, the default is no parameters.

CPYCPYN



CPYCPYN
Specifies the template name to be used in a COPYDDN clause which is added to the COPY utility statement. *cc_name* can be up to eight characters in length. The CPYCPYN keyword is optional.

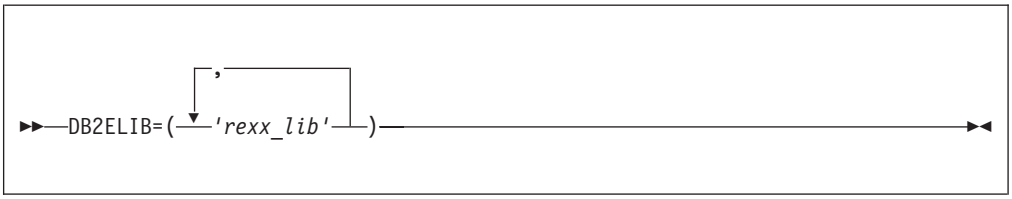
DB2CLIB



DB2CLIB
Specifies the DB2 CLIST libraries for the DB2 subsystem defined by SSID.

You only need to specify a value for DB2CLIB if the libraries have not been allocated prior to starting FM/DB2. You can do this in several ways, for example, in the TSO logon procedure as JCL statements, or in an initialization clist or exec run as part of the TSO logon procedure, or in a clist or exec executed prior to the initialization of FM/DB2.

DB2ELIB



DB2ELIB
Specifies the DB2 REXX libraries for the DB2 subsystem defined by SSID.

You only need to specify a value for DB2ELIB if the libraries have not been allocated prior to starting FM/DB2. You can do this in several ways, for example, in the TSO logon procedure as JCL statements, or in an initialization clist or exec run as part of the TSO logon procedure, or in a clist or exec executed prior to the initialization of FM/DB2.

DB2LLIB

DB2LLIB
Specifies the DB2 load libraries for the DB2 subsystem defined by SSID. All data sets defined to the DB2LLIB keyword must be APF-authorized.

You only need to specify a value for DB2LLIB if:

- The DB2 load libraries have not been added to the LINKLIST, and the libraries have not been allocated as part of the TSO logon procedure, or any initialization clist or exec run prior to starting FM/DB2.

  or:

- DB2 load libraries have been added to the LINKLIST, but the libraries in the LINKLIST are for a different version or release of DB2 to the DB2 system defined by this FMN2SSDM macro.

Notes:

1. All DB2 load libraries, however specified to FM/DB2, must be APF-authorized.
2. If you code DB2 load library names in the FMN2SSDM macro entries for a DB2 system, FM/DB2 will use an ISPF LIBDEF ISPLLIB service to allocate the DB2 load libraries. This requires the user to have either

READ or EXECUTE access to the DB2 load libraries. When EXECUTE access only is given, the environment must be a 'controlled environment' in RACF terms, and you may need to write additional security rules. For information about establishing and using a RACF 'controlled environment', see the *z/OS Security Server RACF Security Administrator's Guide*, or equivalent documentation for your security product, if you do not use RACF.

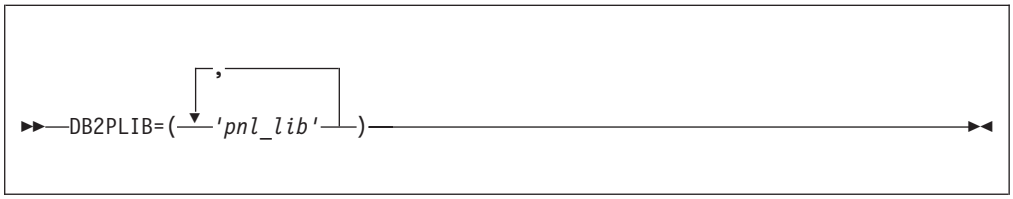
See "Alternatives for making FM/DB2 available" on page 102 for information about the LINKLIST and TSO logon procedures for FM/DB2.

DB2MLIB

DB2MLIB
Specifies the DB2 message libraries for the DB2 subsystem defined by SSID.

You only need to specify a value for DB2MLIB if the libraries have not been allocated prior to starting FM/DB2. You can do this in several ways, for example, in the TSO logon procedure as JCL statements, or in an initialization clist or exec run as part of the TSO logon procedure, or in a clist or exec executed prior to the initialization of FM/DB2.

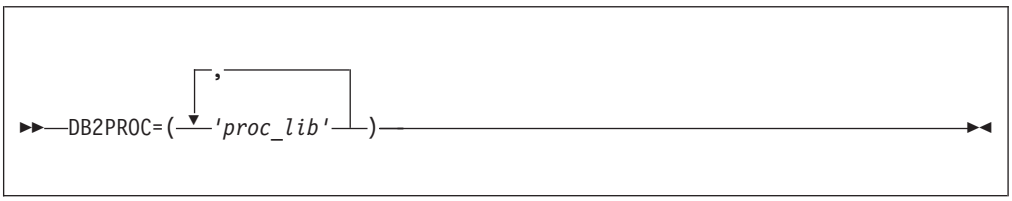
DB2PLIB



DB2PLIB
Specifies the DB2 panel libraries for the DB2 subsystem defined by SSID.

You only need to specify a value for DB2PLIB if the libraries have not been allocated prior to starting FM/DB2. You can do this in several ways, for example, in the TSO logon procedure as JCL statements, or in an initialization clist or exec run as part of the TSO logon procedure, or in a clist or exec executed prior to the initialization of FM/DB2.

DB2PROC



DB2PROC

Specifies the DB2 procedure libraries for the DB2 subsystem defined by SSID.

You only need to specify a value for DB2PROC for a DB2 subsystem if the standard DB2 procedures (for example, DSNUPROC) are not stored in a system procedure library. Specifying a DB2PROC statement will mean that any batch JCL generated by FM/DB2 will include a JCLLIB statement, when FM/DB2 is connected to that subsystem.

DB2RLIB

DB2RLIB

Specifies the DB2 run libraries for the DB2 subsystem defined by SSID.

You only need to specify a value for DB2RLIB if:

- The DB2 run libraries have not been added to the LINKLIST, and the libraries have not been allocated as part of the TSO logon procedure, or any initialization clist or exec run prior to starting FM/DB2.

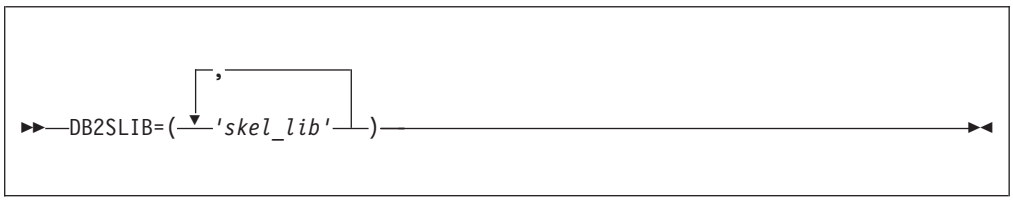
  or:

- DB2 run libraries have been added to the LINKLIST, but the libraries in the LINKLIST are for a different version or release of DB2 to the DB2 system defined in this FMN2SSDM macro.

Note: If you add your DB2 run libraries to the same TSO logon procedure, or initialization clist or exec, as your DB2 load libraries, then the DB2 run libraries must also be APF-authorized.

See "Alternatives for making FM/DB2 available" on page 102 for information about the LINKLIST and TSO logon procedures for FM/DB2.

DB2SLIB



DB2SLIB
Specifies the DB2 skeleton libraries for the DB2 subsystem defined by SSID.

You only need to specify a value for DB2SLIB if the libraries have not been allocated prior to starting FM/DB2. You can do this in several ways, for example, in the TSO logon procedure as JCL statements, or in an initialization clist or exec run as part of the TSO logon procedure, or in a clist or exec executed prior to the initialization of FM/DB2.

DB2TLIB

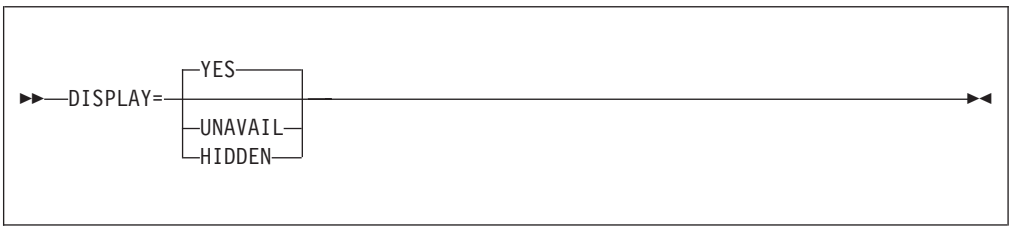
DB2TLIB
Specifies the DB2 table libraries for the DB2 subsystem defined by SSID.

You only need to specify a value for DB2TLIB if the libraries have not been allocated prior to starting FM/DB2. You can do this in several ways, for example, in the TSO logon procedure as JCL statements, or in an initialization clist or exec run as part of the TSO logon procedure, or in a clist or exec executed prior to the initialization of FM/DB2.

DESC

DESC Is the description of the subsystem that will appear in the FM/DB2 `DB2 Subsystem Selection` panel for this SSID. The maximum length of the description is 45 characters, enclosed in quotation marks.

DISPLAY



DISPLAY
Controls the display of the SSID on the subsystem selection panel.

YES The subsystem or group ID (SSID) will be displayed on the subsystem selection list. Inactive subsystems will not appear if **LIST=ACTIVE** is specified in the FMN2POPI macro. The default is YES.

UNAVAIL
Specify **DISPLAY=UNAVAIL** if FM/DB2 has not been installed on a subsystem or DB2 data sharing group, or you require that users will not be able to connect to the subsystem or group.

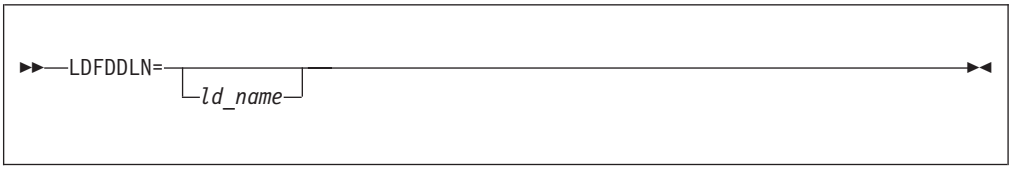
SSIDs specified UNAVAIL will appear on the subsystem selection list if **LIST=ALL** is specified in the FMN2POPI macro. The SSID appears with a status of UNAVAIL. and cannot be selected.

Unavailable SSIDs will not appear on the selection list if **LIST=ACTIVE** is specified in the FMN2POPI macro. However, FM/DB2 will attempt to connect to subsystems marked as UNAVAIL if **CONNECT=ANY** is specified in the FMN2POPI macro.

HIDDEN
Specify **DISPLAY=HIDDEN** when you require that an SSID should not appear on the subsystem selection list, and that users should not be able to connect to it.

Hidden subsystems will not appear on the subsystem selection list, regardless of the setting of the **LIST** option in the FMN2POPI macro. However, FM/DB2 will attempt to connect to "hidden" subsystems if **CONNECT=ANY** is specified in the FMN2POPI macro.

LDFDDLN



LDFDDLN
Specifies the DDNAME to be used in JCL to reference the site-specific listdef library. *ld_name* can be up to eight characters, and must conform to standard job control conventions. This library contains the listdef statements that establish the default site policy.

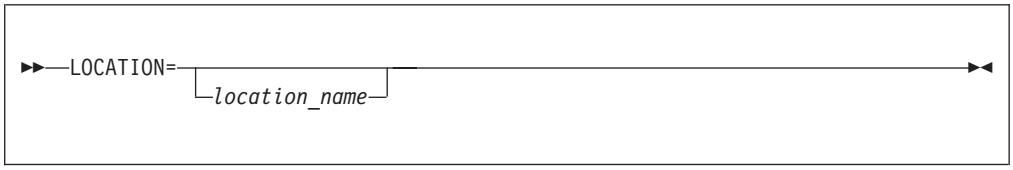
Specify a value for this keyword if you want to implement a site-specific policy for listdefs.

The LDFDDLN keyword is optional. If no value is specified for LDFDDLN then no installation value will be used as a default. The FM/DB2 default (SYSLISTD) will be used.

If you specify a value for LDFDDLN, also specify a value for at least SLDJCL1. Using the SLDJCL1 - SLDJCL4 keywords, you can build a DDNAME concatenation that comprises up to four libraries. See "SLDJCL1" on page 343, "SLDJCL2" on page 344, "SLDJCL3" on page 344, and "SLDJCL4" on page 344.

For an example, see "Examples of FMN2SSDM macros" on page 122.

LOCATION



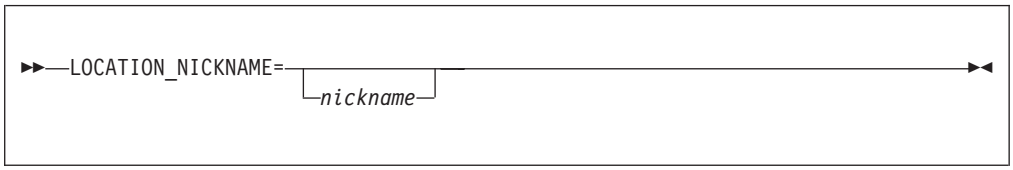
LOCATION
Specifies the DB2 location name for the DB2 system or group. This is the LOCATION value entered in the DDF statement (DB2 change log inventory utility), when DB2 was installed. Alternatively you can specify the value returned by the following SQL statement:

```
SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1.
```

The maximum length you can specify for *location_name* is 16 characters, and must conform to the naming conventions for a "location-name" as documented in the SQL Reference manual for your DB2 system.

Note: The LOCATION option is optional. FM/DB2 does not require the specification of the location name in the FMN2SSDM macro for successful access to remote DB2 systems. Specifying a DB2 location name enables the use of the DB2 subsystem or group identifier (for example, DSNA) as an alternative to the actual DB2 location name, for on-line functions only.

LOCATION_NICKNAME

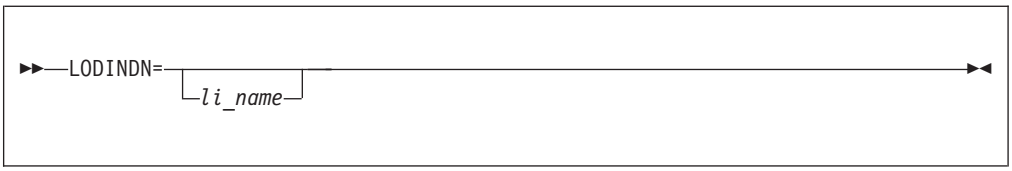


LOCATION_NICKNAME
Specifies a nickname for the DB2 system or group. This can be any value conforming to the rules for an SQL ordinary identifier. Examples include DEV, TEST, PROD, LA, NY.

The maximum length you can specify for *nickname* is 16 characters.

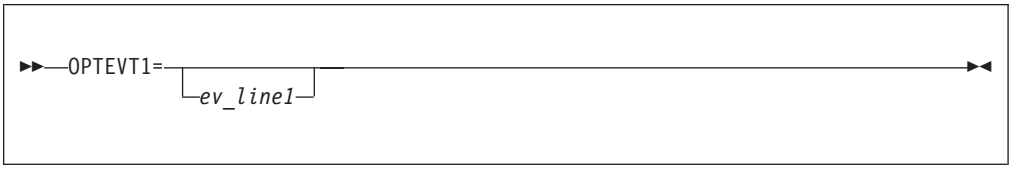
Note: The LOCATION_NICKNAME option is optional. If you specify a value for *nickname* for the DB2 system or group, you must also specify a value for the LOCATION option. Specifying a DB2 location nickname enables the use of that nickname (for example, DEV) as an alternative to the actual DB2 location name, for on-line functions only.

LODINDN



LODINDN
Specifies the template name to be used in an INDDN clause which is added to the LOAD utility statement. *li_name* can be up to eight characters in length. The LODINDN keyword is optional.

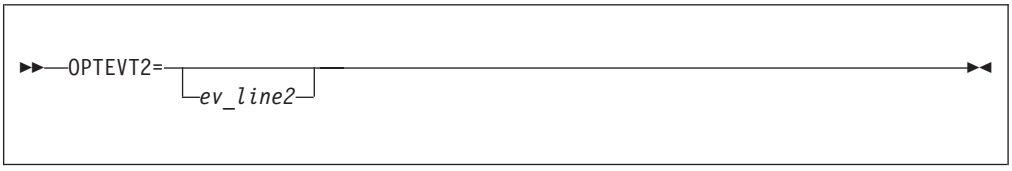
OPTEVT1



OPTEVT1
Specifies the first line to be placed in the EVENT clause of the OPTION utility statement. *ev_line1* can be up to 54 characters in length. If your clause includes one or more commas, enclose the clause within single quotation marks. The OPTEVT1 keyword is optional.

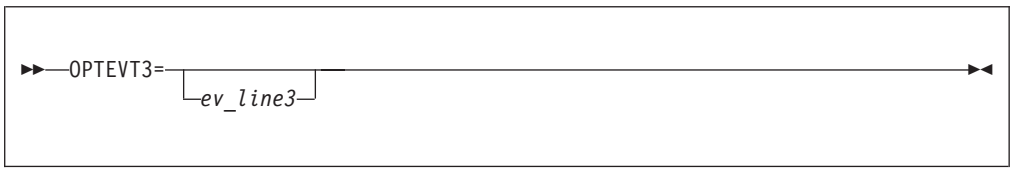
For an example, see "Examples of FMN2SSDM macros" on page 122.

OPTEVT2



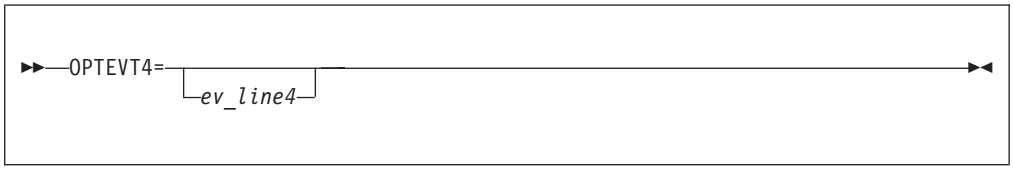
OPTEVT2
Specifies the second line to be placed in the EVENT clause of the OPTION utility statement. *ev_line2* can be up to 54 characters in length. If your clause includes one or more commas, enclose the clause within single quotation marks. The OPTEVT2 keyword is optional.

OPTEVT3



OPTEVT3
Specifies the third line to be placed in the EVENT clause of the OPTION utility statement. *ev_line3* can be up to 54 characters in length. If your clause includes one or more commas, enclose the clause within single quotation marks. The OPTEVT3 keyword is optional.

OPTEVT4



OPTEVT4
Specifies the fourth line to be placed in the EVENT clause of the OPTION utility statement. *ev_line4* can be up to 54 characters in length. If your clause includes one or more commas, enclose the clause within single quotation marks. The OPTEVT4 keyword is optional.

PLAN

PLAN Specifies the name of the FM/DB2 plan for the DB2 subsystem defined by SSID. The default plan name is FMN2PLAN.

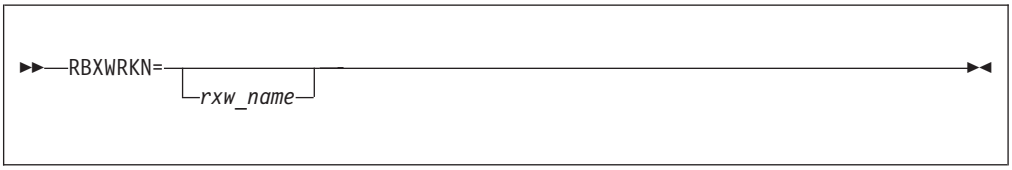
If you change **FMN2PLAN** you must make the same change in the sample job, **FMN2BIND**. See “Binding DB2 (required)” on page 119 for information about **FMN2BIND**.

PLAN2

PLAN2
Specifies the name of the DB2 reverse engineering plan for the DB2 subsystem defined by SSID. The default reverse engineering plan name is FMN2GEN.

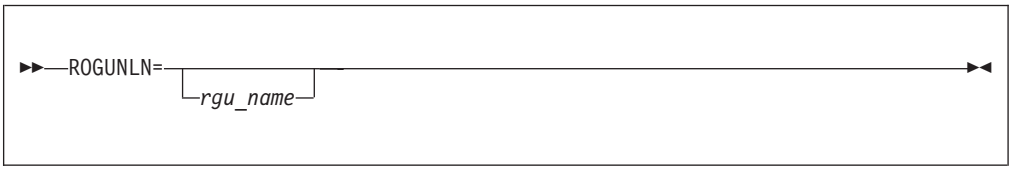
If you change **FMN2GEN**, you must make the same change in the sample job, **FMN2BIND**. See "Binding DB2 (required)" on page 119 for information about **FMN2BIND**.

RBXWRKN



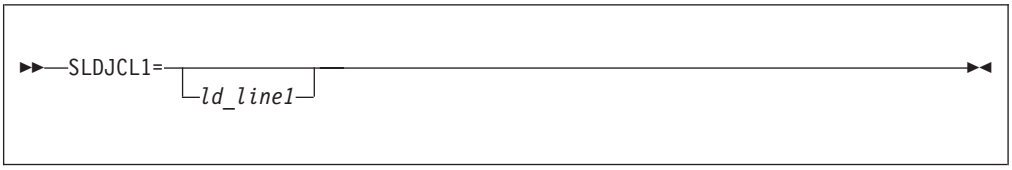
RBXWRKN
Specifies the template name to be used in an WORKDDN clause which is added to the REBUILD INDEX utility statement. *rxw_name* can be up to eight characters in length. The RBXWRKN keyword is optional.

ROGUNLN



ROGUNLN
Specifies the template name to be used in an UNLDDN clause which is added to the REORG TABLESPACE utility statement. *rgu_name* can be up to eight characters in length. The ROGUNLN keyword is optional.

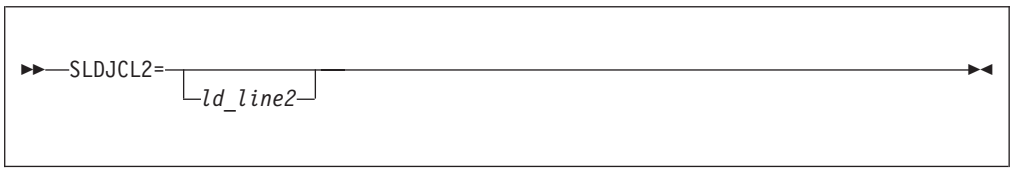
SLDJCL1



SLDJCL1
Specifies the first line of the generated DD statement for the site-specific listdef library concatenation, as specified on the LDFDDLN keyword. *ld_line1* can be up to 54 characters in length, and must conform to standard job control conventions. If your JCL statement includes one or more commas, then enclose the whole statement within single quotation marks. The data set so referenced must exist.

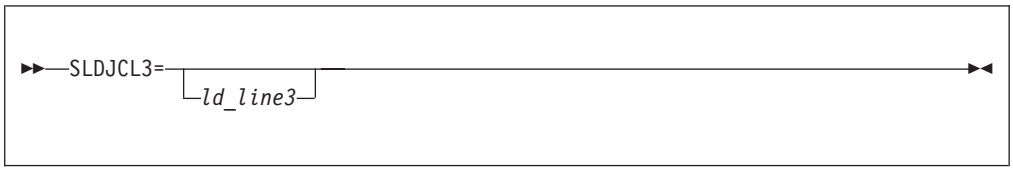
You can specify a value for SLDJCL1 even if you have not specified a value for LDFDDLN. however, if you have specified a value for LDFDDLN, then also specify a value for SLDJCL1.

SLDJCL2



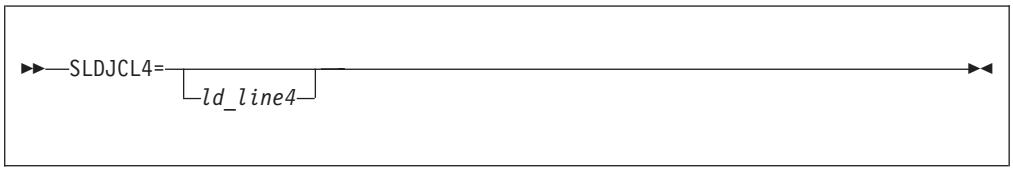
SLDJCL2
Specifies the second line of the generated DD statement for the site-specific listdef library concatenation, as specified on the LDFDDLN keyword. *ld_line2* can be up to 54 characters in length, and must conform to standard job control conventions. If your JCL statement includes one or more commas, then enclose the whole statement within single quotation marks. The data set so referenced must exist. The SLDJCL2 keyword is optional.

SLDJCL3



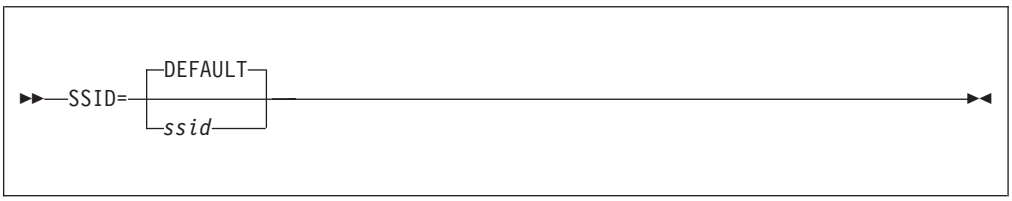
SLDJCL3
Specifies the third line of the generated DD statement for the site-specific listdef library concatenation, as specified on the LDFDDLN keyword. *ld_line3* can be up to 54 characters in length, and must conform to standard job control conventions. If your JCL statement includes one or more commas, then enclose the whole statement within single quotation marks. The data set so referenced must exist. The SLDJCL3 keyword is optional.

SLDJCL4



SLDJCL4
Specifies the fourth line of the generated DD statement for the site-specific listdef library concatenation, as specified on the LDFDDLN keyword. *ld_line4* can be up to 54 characters in length, and must conform to standard job control conventions. If your JCL statement includes one or more commas, then enclose the whole statement within single quotation marks. The data set so referenced must exist. The SLDJCL4 keyword is optional.

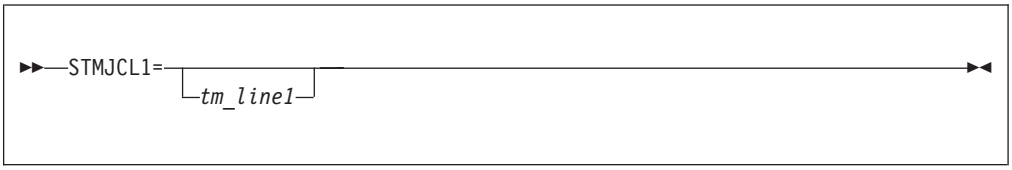
SSID



SSID Defines the (up to) 4-character DB2 subsystem or group ID for the DB2 system that FM/DB2 will connect to. The 4-character value will appear on the FM/DB2 primary option menu when FM/DB2 is connected to the DB2 system or group.

You must specify at least one FMN2SSDM macro definition, and the last (or only) FMN2SSDM macro definition must specify SSID=DEFAULT. If you have already specified a site-specific value for SSID in your last FMN2SSDM macro, you must supply further FMN2SSDM macro in which the only statement is SSID=DEFAULT. See example **DG02S**, page 123.

STMJCL1



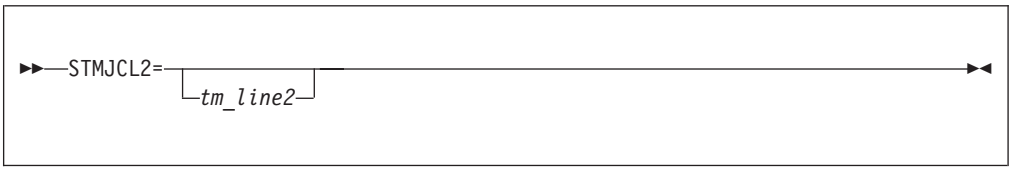
STMJCL1

Specifies the first line of the generated DD statement for the site-specific template library concatenation, as specified on the TMPDDLN keyword. *tm_line1* can be up to 54 characters in length, and must conform to standard job control conventions. If your JCL statement includes one or more commas, then enclose the whole statement within single quotation marks. The data set so referenced must exist.

You can specify a value for STMJCL1 even if you have not specified a value for TMPDDLN. However, if you have specified a value for TMPDDLN, then also specify a value for STMJCL1.

For an example, see "Examples of FMN2SSDM macros" on page 122.

STMJCL2

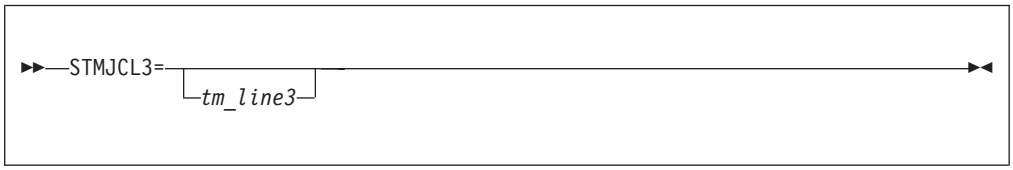


STMJCL2

Specifies the second line of the generated DD statement for the site-specific template library concatenation, as specified on the TMPDDLN keyword. *tm_line2* can be up to 54 characters in length, and must conform to standard job control conventions. If your JCL statement includes one or

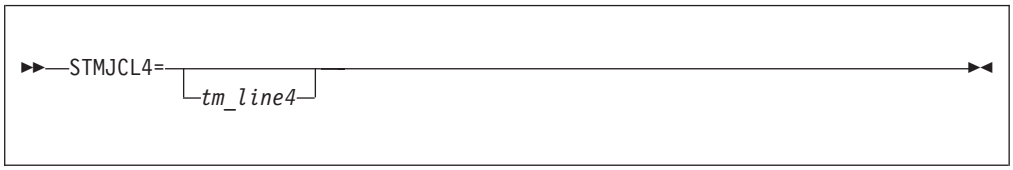
more commas, then enclose the whole statement within single quotation marks. The data set so referenced must exist. The STMJCL2 keyword is optional.

STMJCL3



STMJCL3
Specifies the third line of the generated DD statement for the site-specific template library concatenation, as specified on the TMPDDLN keyword. *tm_line3* can be up to 54 characters in length, and must conform to standard job control conventions. If your JCL statement includes one or more commas, then enclose the whole statement within single quotation marks. The data set so referenced must exist. The STMJCL3 keyword is optional.

STMJCL4



STMJCL4
Specifies the fourth line of the generated DD statement for the site-specific template library concatenation, as specified on the TMPDDLN keyword. *tm_line4* can be up to 54 characters in length, and must conform to standard job control conventions. If your JCL statement includes one or more commas, then enclose the whole statement within single quotation marks. The data set so referenced must exist. The STMJCL4 keyword is optional.

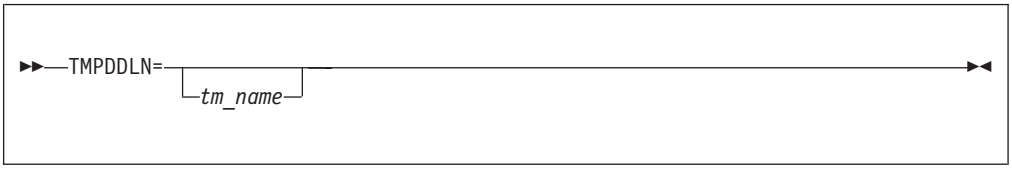
TABLE_LOCKING

TABLE_LOCKING
Specifies whether a table being edited can be explicitly locked. An explicit LOCK TABLE SQL statement is only issued if the "locking" option on the seventh editor options panel is set to either 2 (Share Mode) or 3 (Exclusive Mode).

YES An explicit LOCK TABLE SQL statement might be issued.

NO An explicit LOCK TABLE SQL statement is not issued, regardless of the setting of the "locking" editor option.

TMPDDLN



TMPDDLN

Specifies the DDNAME to be used in JCL to reference the site-specific template library. *tm_name* can be up to eight characters, and must conform to standard job control conventions. This library contains the template statements that establish the default site policy.

Specify a value for this keyword if you want to implement a site-specific policy for templates.

The TMPDDLN keyword is optional. If no value is specified for TMPDDLN then no installation value will be used as a default. The FM/DB2 default (SYSTEMPL) will be used.

If you specify a value for TMPDDLN, also specify a value for at least STMJCL1. Using the STMJCL1 - STMJCL4 keywords, you can build a DDNAME concatenation that comprises up to four libraries. See "STMJCL1" on page 345, "STMJCL2" on page 345,"STMJCL3" on page 346, and "STMJCL4" on page 346.

For an example, see "Examples of FMN2SSDM macros" on page 122.

TYPE

TYPE Specifies the type of DB2 subsystem defined by SSID.

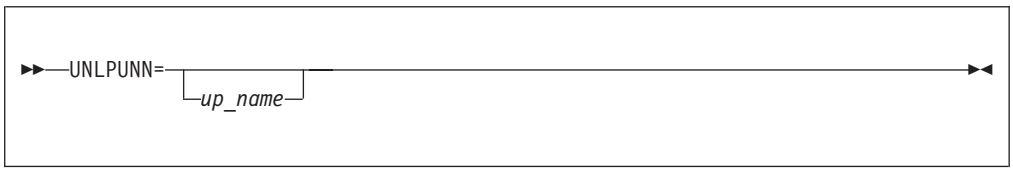
SUBSYS

Specifies that this SSID is the name of a DB2 subsystem defined in the IEFSSNxx member of SYS1.PARMLIB. The default is SUBSYS.

GROUP

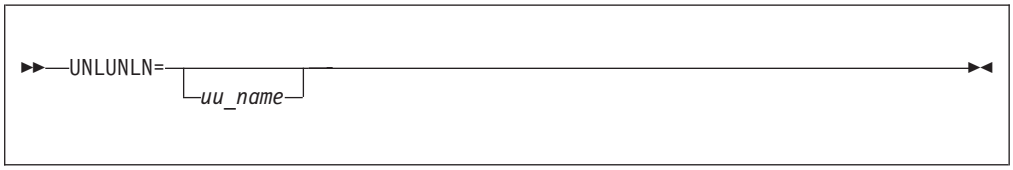
Specifies that this SSID is a DB2 group attachment identifier in a DB2 data sharing environment.

UNLPUNN



UNLPUNN
Specifies the template name to be used in an PUNCHDDN clause which is added to the UNLOAD utility statement. *up_name* can be up to eight characters in length. The UNLPUNN keyword is optional.

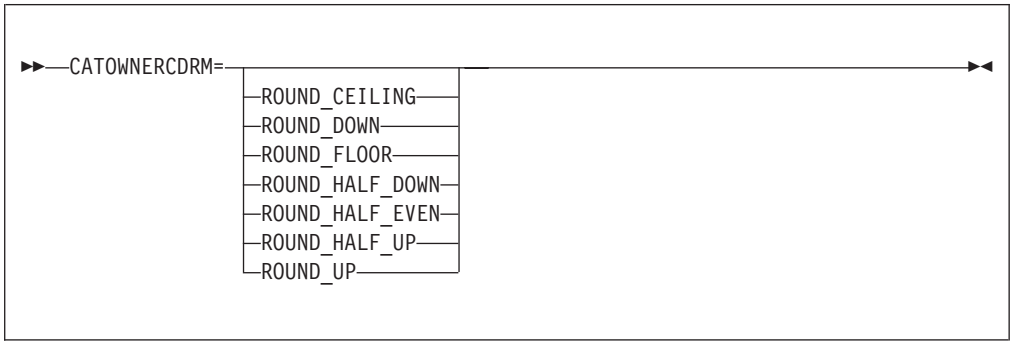
UNLUNLN



UNLUNLN
Specifies the template name to be used in an UNLDDN clause which is added to the UNLOAD utility statement. *uu_name* can be up to eight characters in length. The UNLUNLN keyword is optional.

FMN2POPI

CATOWNERCDRM



CATOWNERCDRM
Specifies the current DECFLOAT rounding mode to be used when accessing a DB2 version 9 (or later) system.

This option is only relevant when FM/DB2 is connected to a DB2 version 9 or later system. Any specified value is ignored when connected to a DB2 version 8 or earlier system.

Important Information

The current DECFLOAT rounding mode[1] (CDRM) becomes critical when processing ANY DB2 view in a DB2 version 9 or later system. This is because DB2 records the CDRM in effect when the view was created in the DB2 catalog tables. When the view is subsequently accessed, DB2 checks the CDRM when the view was created against the CDRM of the process attempting to access the view. If the two values do not match, DB2 disallows the access and issues SQLCODE-270. The CDRM for the process must be changed to be the same as the CDRM value when the view was originally created, in order for the view access to succeed.

This restriction applies to every view created in a DB2 version 9 system, regardless of whether or not there are DECFLOAT columns defined in the view.

If you intend to specify a CATOWNER parameter to implement access to the DB2 catalog tables (that is, FM/DB2 will access views of the DB2 catalog tables, rather than the catalog tables directly), you need to take account of the CDRM constraint as noted above. It is recommended that:

- You create all of the views of the DB2 catalog table at the same time, and set [2] the required CDRM before creating the views.
- The value used for the CDRM should be specified in the CATOWNERCDRM parameter.
- In the situation where FM/DB2 will access multiple DB2 version 9 (or later) systems, use the same CDRM value when creating the catalog table views for all target DB2 systems. Specify the selected value in the CATOWNERCDRM parameter.

If the CATOWNERCDRM parameter is omitted, or is incorrectly specified, FM/DB2 will auto-detect the CDRM mode used when attempting to access a view of the DB2 catalog table. This is an expensive procedure, requiring FM/DB2 to prepare up to 7 SQL statements each time a user accesses the DB2 system. It is therefore advisable for performance reasons to ensure that the CATOWNERCDRM value is correctly specified.

FM/DB2 will not function correctly if the required views of the DB2 catalog tables are created using more than one CDRM value.

ROUND_CEILING
: Round towards positive infinity. If all of the discarded digits are zero or if the sign is negative, the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient is incremented by 1 (round up).

ROUND_DOWN
: Round towards 0 (truncation). The discarded digits are ignored.

ROUND_FLOOR
: Round towards negative infinity. If all of the discarded digits are zero or if the sign is positive, the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient is incremented by 1 (round down).

ROUND_HALF_DOWN
Round to nearest value; if values are equidistant, rounds down. If the discarded digits represent greater than half (0.5) of the value of a number in the next left position, the result coefficient is incremented by 1 (round up). Otherwise, the discarded digits are ignored. This rounding mode is not recommended when creating a portable application because it is not supported by the IEEE draft standard for floating-point arithmetic.

ROUND_HALF_EVEN
Round to nearest value; if values are equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) of the value of a number in the next left position, the result coefficient is incremented by 1 (round up). If the discarded digits represent less than half of the value, the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise, the result coefficient is unaltered if its rightmost digit is even, or is incremented by 1 (round up) if its rightmost digit is odd (to make an even digit).

ROUND_HALF_UP
Round to nearest value; if values are equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a number in the next left position, the result coefficient is incremented by 1 (round up). Otherwise the discarded digits are ignored.

ROUND_UP
Round away from 0. If all of the discarded digits are zero, the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient is incremented by 1 (round up).

Notes:

1. The current DECFLOAT rounding mode (CDRM) refers to the behaviour of DB2 when processing a DECFLOAT number, where rounding of the value is required. See the DB2 manuals for a more detailed explanation.
2. The CDRM can be set using:
   ```
   SET CURRENT DECFLOAT ROUNDING MODE = value
   ```
   See the DB2 SQL Reference manual for a detailed explanation.

CATOWNER

CATOWNER
Specifies to FM/DB2 the owner of every DB2 catalog table or view accessed by FM/DB2.

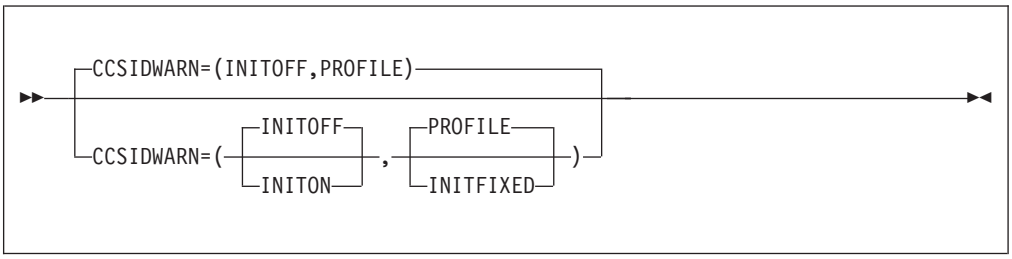
The default is SYSIBM. This means the DB2 catalog tables are accessed directly. You can specify an alternative value if you want FM/DB2 to access a copy of the DB2 catalog tables, or views defined on the DB2 catalog tables.

Important note: If you specify a *value* for CATOWNER, this value is used by FM/DB2 when accessing every DB2 subsystem defined in the

FMN2POPT module (each subsystem is defined by a separate invocation of the FMN2SSDM macro). Therefore, if you allow FM/DB2 to access the DB2 catalog directly on one DB2 subsystem, you must do so for every DB2 subsystem accessed by FM/DB2. Similarly, if you define views on the DB2 catalog tables for one DB2 subsystem and direct FM/DB2 to access those views, you must define views, with the same owner, for every DB2 subsystem accessed by FM/DB2.

For more information about FM/DB2 and the DB2 catalog tables, see "Granting access to the DB2 catalog (required)" on page 109.

CCSIDWARN



CCSIDWARN

Controls whether a warning message will be displayed when FM/DB2 connects to a DB2 subsystem, if the CCSID of the terminal differs from the CCSID of the plan. FM/DB2 will connect to a DB2 subsystem when the application id first entered or when the DB2 SSID is changed.

This option is used in conjunction with the CCSIDWARNIGNORE option in the FMN2SSDM macro.

Options are provided to allow you to:

- Set the initial value of the system CCSID Warning message option for new FM/DB2 users.
- Force the system CCSID Warning message option for all FM/DB2 users; alternatively allow FM/DB2 users to control the initial setting using the new system option, that is saved in the user's ISPF profile.

There are two sub-parameters to CCSIDWARN. You do not have to specify them all in FMN2POPI, but if you do so, they must be in the order shown in the syntax diagram. If you specify more than one, they must be separated by commas and enclosed in parentheses.

If you do not specify CCSIDWARN at all in your FMN2POPI macro, the default setting of (INITOFF,PROFILE) is used.

Note: This option only applies to DB2 Release 7 and later.

The sub-parameters are described below.

INITOFF, INITON

Determines the initial system CCSID Warning message option setting in the user's profile, when the profile is first created.

For users with an existing FM/DB2 profile, the effect depends on the setting of the second parameter (described below). If INITFIXED is specified for the second parameter, then every FM/DB2 edit session begins with either the system CCSID

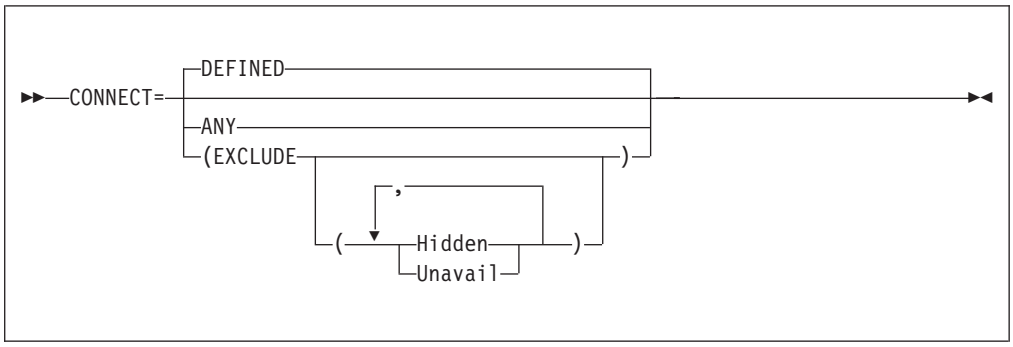
Warning message option not being set if INITOFF is specified, or the system CCSID Warning message option being set if INITON is specified.

PROFILE, INITFIXED
Determines if the user can alter the initial setting of system CCSID Warning message option in FM/DB2 edit.

INITFIXED prevents the user from altering the initial setting of the system CCSID Warning message option.

PROFILE allows the user to change this initial setting and it is preserved between FM/DB2 sessions in the user's profile.

CONNECT



CONNECT
Controls how FM/DB2 connects to DB2 subsystems or data sharing groups.

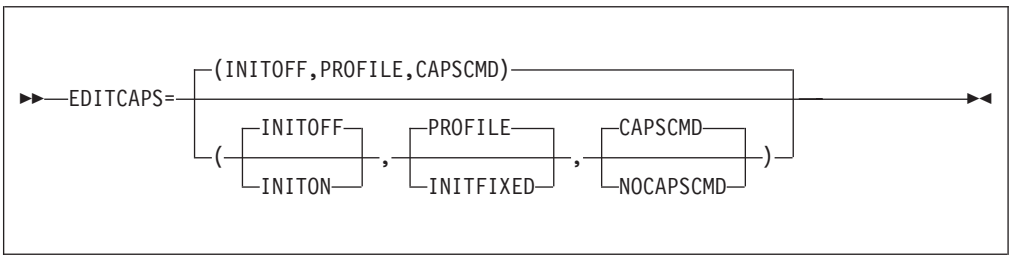
DEFINED
Connection will only be attempted to active DB2 subsystems or groups defined in the FMN2SSDM macro. The default is **DEFINED**.

ANY Will allow FM/DB2 to attempt to connect to any specified SSID, whether defined in the FMN2SSDM macro or not. FM/DB2 will only attempt connection to an SSID (either DB2 subsystem or data sharing group) that is active on the local z/OS system.

CONNECT=ANY over-rides DISPLAY=UNAVAIL and DISPLAY=HIDDEN specified in the FMN2SSDM macro, however the connection will only be attempted if the specified subsystem or group is active.

EXCLUDE
Excludes Hidden or Unavailable DB2 systems when attempting to connect to a DB2 system. If no parameters are specified for EXCLUDE, both Hidden and Unavailable systems are excluded. If both parameters are specified, separate them with a comma. The parameters may be abbreviated to H and U respectively.

EDITCAPS



EDITCAPS

Controls the setting of the CAPS option in edit. A number of options are provided, which allow you to:

- Set the initial value (either CAPS ON or CAPS OFF) for new FM/DB2 users.
- Force all users of the FM/DB2 editor to commence an edit session with either CAPS ON or CAPS OFF; alternatively to allow the user to control the initial CAPS setting using a new editor option, that is saved in the user's ISPF profile.
- Disable the CAPS command within an FM/DB2 edit session.

  You can use certain settings of the EDITCAPS parameters to create an environment where every FM/DB2 edit session starts with CAPS ON, and this cannot be altered using the CAPS OFF command.

There are three sub-parameters to EDITCAPS. You do not have to specify them all in FMN2POPI, but if you do so, they must be in the order shown in the syntax diagram. If you specify more than one, they must be separated by commas. If you specify any sub-parameters, you must also provide the parentheses.

If you do not specify EDITCAPS at all in your FMN2POPI macro, the default setting of `(INITOFF,PROFILE,CAPSCMD)` is used.

These sub-parameters are described below.

INITOFF, INITON

Determines the initial CAPS setting in the user's profile, when the profile is first created.

For users with an existing FM/DB2 profile, the effect depends on the setting of the second parameter (described below). If INITFIXED is specified for the second parameter, then every FM/DB2 edit session begins with either CAPS OFF if INITOFF is specified, or CAPS ON if INITON is specified.

PROFILE, INITFIXED

Determines if the user can alter the initial setting of CAPS in FM/DB2 edit. INITFIXED prevents the user from altering the initial setting of CAPS. PROFILE allows the user to change this initial setting and it is preserved between FM/DB2 sessions in the user's profile.

CAPSCMD, NOCAPSCMD

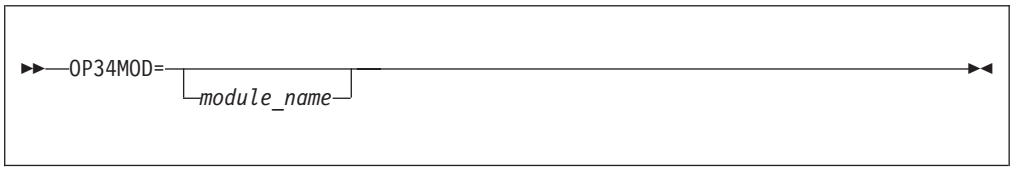
Determines if the user is permitted to use the CAPS command within an FM/DB2 edit session. NOCAPSCMD does not affect the CASE command, which remains available for use.

When CAPSCMD is specified, the user can turn CAPS on or off in an FM/DB2 edit session, as required. When NOCAPSCMD is specified, the CAPS command is disabled, and is fixed at whatever setting was selected when the FM/DB2 edit session commenced.

Notes:

1. EDITCAPS can also be specified in the FMN0POPI macro, which appears at the start of the FMN2POPT definition. Specifying EDITCAPS in the FMN0POPI macro has no effect on the FM/DB2 editor. To change the EDITCAPS behaviour in an FM/DB2 edit session, the EDITCAPS option must be specified in the FMN2POPI macro.
2. The EDITCAPS option available for the File Manager base function has no effect in FM/DB2.

OP34MOD



OP34MOD

Specifies the name of the data module FM/DB2 will use when displaying DB2 catalog table information, with FM/DB2 options 3.4, 3.5 and 4.5.

FM/DB2 uses a data module to display the DB2 catalog table information in a specific language. If no value is specified for OP34MOD, FM/DB2 determines which module to load from the value specified for the LANGUAGE option in FMN2POPT. You can specify a value for OP34MOD to override this automatic selection of the data module.

The name of the module is in the form FMN2D*yyy*.

The default for OP34MOD is OP34MOD=(blank). In this case FM/DB2 will load a module FMN2D*yyy* where *yyy* is the language code for the language specified on the LANGUAGE option in FMN2POPT. See table 8,"Keyword values for the LANGUAGE option", in chapter 6, for the code and keyword values for your language.

Notes:

1. If LANGUAGE=ENGLISH is specified in FMN2POPT and a value for OP34MOD is not specified, then the module used is FMN2DENU.
2. If a value is specified for LANGUAGE but no module FMN2D*yyy* exists with a value for *yyy* corresponding to the LANGUAGE option, then FMN2DENU is used.
3. Four data modules are shipped with FM/DB2, for English, upper case English, Japanese, and Korean. These modules are FMN2DENU, FMN2DENP, FMN2DJPN, and FMN2DKOR respectively.

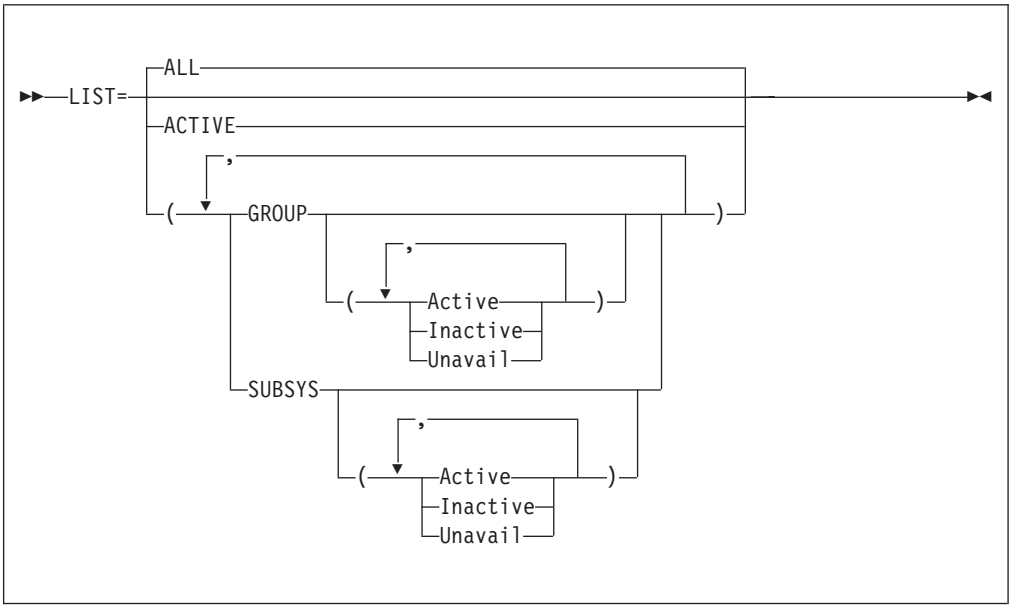
Example

A Japanese user is using ISPF ENGLISH, (and therefore has LANGUAGE=ENGLISH specified in FMN2POPT), but wants the DB2 catalog table information to be displayed in Japanese. Therefore they require FMN2DJPN, the Japanese version of the data module. In this case, specify

```
OP34MOD=FMN2DJPN
```

to ensure that the Japanese version of the data module is loaded.

LIST



LIST Controls which items appear in the subsystem selection list. FM/DB2 scans the sub-system vector table of the operating system when preparing the subsystem selection list. All DB2 systems found are considered **eligible** for inclusion on the subsystem selection list. If a DB2 system is not defined to z/OS it **will not appear** in the sub-system vector table and will therefore **not appear** in the FM/DB2 sub-system selection list, even if there is an FMN2SSDM macro entry for it.

ALL All eligible (see above) subsystems and group identifiers defined in the FMN2SSDM macro appear in the subsystem selection list, with the exception of any entries specified DISPLAY=HIDDEN in the FMN2SSDM macro. The default is **LIST=ALL**.

ACTIVE
Restricts the subsystem selection list to active DB2 subsystems and data sharing groups (on the local z/OS system).

GROUP
Lists DB2 group entries. If no parameters are specified on the GROUP option, all Active, Inactive, and Unavailable group entries are listed. To select groups in a specific status, provide one or more parameters for the GROUP option. You can specify any combination of Active, Inactive and Unavailable. If more than one parameter is specified, separate them with a comma. The parameters may be abbreviated to A, I, or U respectively.

If SUBSYS is specified, but GROUP is not, no DB2 data sharing group items will appear on the subsystem selection list.

SUBSYS
Lists DB2 subsystem entries. If no parameters are specified on the SUBSYS option, all Active, Inactive, and Unavailable subsystem

entries are listed. To select subsystems in a specific status, provide one or more parameters for the SUBSYS option. You can specify any combination of Active, Inactive and Unavailable. If more than one parameter is specified, separate them with a comma. The parameters may be abbreviated to A, I, and U respectively.

If GROUP is specified, but SUBSYS is not, no DB2 subsystems will appear on the subsystem selection list.

Some explanatory information

The following information is provided to assist you in understanding why items appear on the FM/DB2 subsystem selection list. It is not necessary to change the entries in member IEFSSNxx.

DB2 systems are defined to z/OS using entries in member IEFSSNxx in SYS1.PARMLIB. Here are three examples:

```
SUBSYS SUBNAME(DFB2) INITRTN(DSN3INI) INITPARM('DSN3EPX,-DFB2,S')       A
SUBSYS SUBNAME(DFA2) INITRTN(DSN3INI) INITPARM('DSN3EPX,-DFA2,S,DFA2')  B
SUBSYS SUBNAME(DFS2) INITRTN(DSN3INI) INITPARM('DSN3EPX,-DFS2,S,DFG2')  C
```

A defines a DB2 subsystem DFB2 that is not part of a DB2 data sharing group.

B defines a DB2 subsystem DFA2 that is also part of a DB2 data sharing group. The data sharing group is also called DFA2.

C defines a DB2 subsystem DFS2 that is also part of a DB2 data sharing group. The data sharing group name is DFG2.

A generates a single item on the FM/DB2 subsystem selection list. The corresponding FMN2SSDM macro entry should specify:

```
FMN2SSDM  SSID=DFB2,TYPE=SUBSYS, ...
```

B and **C** generate two items each on the FM/DB2 subsystem selection list. For **B** the corresponding FMN2SSDM macro entries are:

```
FMN2SSDM  SSID=DFA2,TYPE=SUBSYS, ...
FMN2SSDM  SSID=DFA2,TYPE=GROUP, ...
```

For **C** the required FMN2SSDM macro entries are:

```
FMN2SSDM  SSID=DFS2,TYPE=SUBSYS, ...
FMN2SSDM  SSID=DFG2,TYPE=GROUP, ...
```

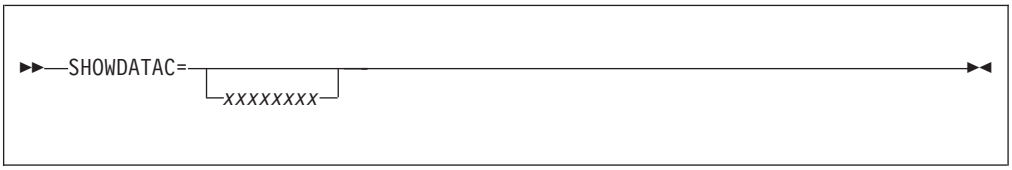
Note that when DB2 system DFA2 is active, DB2 group DFA2 is also active. Similarly when DB2 system DFS2 is active, DB2 group DFG2 is also active. This occurs regardless of whether DB2 data sharing groups DFA2 and DFG2 has been completed implemented or not. From the perspective of z/Os, definitions **B** and **C** in IEFSSNxx are all that is required to define the DB2 data sharing groups DFA2 and DFG2.

In the situation where a DB2 subsystem and DB2 group have the same name (for example, **B**), you can hide one of the entries using DISPLAY=HIDDEN on the corresponding FMN2SSDM macro definition. You can also use the SUBSYS or GROUP options to show only subsystems or groups, respectively, although the latter method applies to every subsystem or group.

For example, to show the group DFA2 but not the subsystem DFA2, the FMN2SSDM macro entries are:

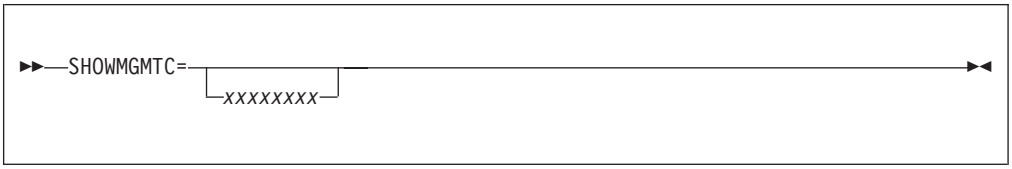
```
FMN2SSDM SSID=DFS2,TYPE=SUBSYS,DISPLAY=HIDDEN
FMN2SSDM SSID=DFG2,TYPE=GROUP, ...
```

SHOWDATAC



SHOWDATAC
Specifies the SMS data class that File Manager uses when allocating the data set created when the SQL command is issued. The maximum length you can specify for SHOWDATAC is eight (8) bytes. SHOWDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied.

SHOWMGMTC

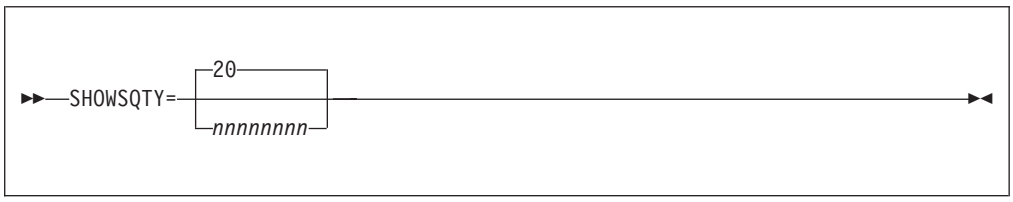


SHOWMGMTC
Specifies the default SMS management class that File Manager uses when allocating the data set created when the SQL command is issued. The maximum length you can specify for SHOWMGMTC is eight (8) bytes. SHOWMGMTC is optional. If omitted the SMS management class specified in PMGMT is used, if supplied.

SHOWPQTY

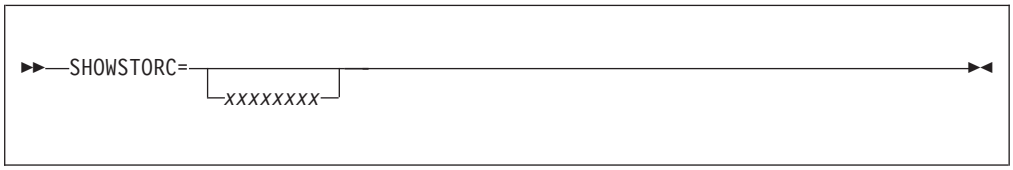
SHOWPQTY
Specifies the amount of DASD space to be used for primary space allocation of the data set created when the SQL command is issued. The range depends on the space unit specified and the DASD device type.

SHOWSQTY



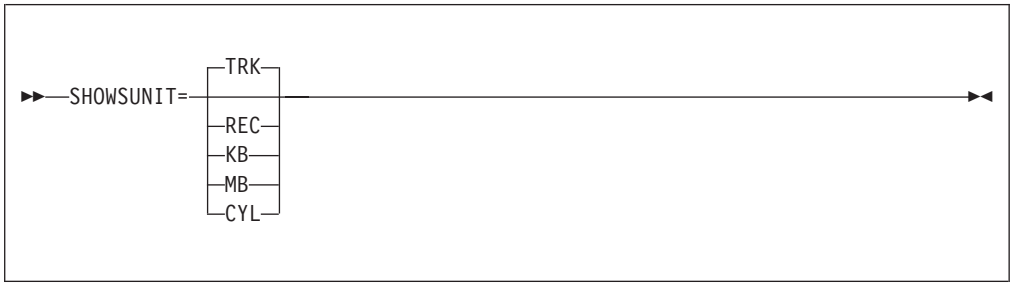
SHOWSQTY
Specifies the amount of DASD space to be used for secondary space allocation of the data set created when the SQL command is issued. The range depends on the space unit specified and the DASD device type.

SHOWSTORC



SHOWSTORC
Specifies the default SMS storage class that File Manager uses when allocating the data set created when the SQL command is issued. The maximum length you can specify for SHOWSTORC is eight (8) bytes. SHOWSTORC is optional. If omitted the SMS storage class specified in PSTORC is used, if supplied.

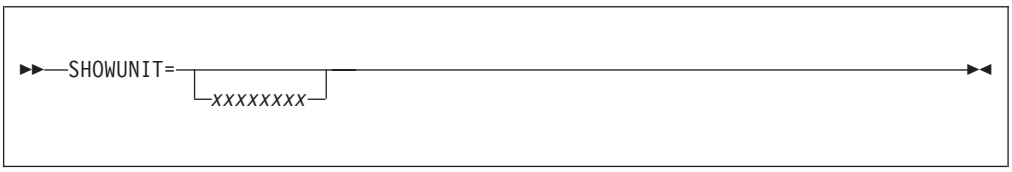
SHOWSUNIT



SHOWSUNIT
Specifies the unit of primary and secondary space to be allocated to the data set created when the SQL command is issued.

REC Record of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

SHOWUNIT



SHOWUNIT
Specifies the default permanent unit that File Manager uses when allocating the data set created when the SQL command is issued. The maximum length you can specify for SHOWUNIT is eight (8) bytes. SHOWUNIT is optional. If omitted the SMS storage class specified in PUNIT is used.

SSIDCMD1

SSIDCMD1
Specifies the FM/DB2 command to be used to dynamically change the currently connected DB2 subsystem. This keyword is optional.

SSID Default command to dynamically change the currently connected DB2 subsystem.

value1 Installation-defined command to dynamically change the currently connected DB2 subsystem. If you specify a value for *value1* it must be 4-8 characters in length, and start with an alphabetic or national character. The remaining characters can be alphabetic, national, or numeric.

SSIDCMD2

SSIDCMD2
Specifies an additional FM/DB2 command to be used to dynamically change the currently connected DB2 subsystem, if required. This keyword is optional.

DB2SYS
Default alternative command to dynamically change the currently connected DB2 subsystem.

value2 An alternative installation-defined command to dynamically

change the currently connected DB2 subsystem. If you specify a value for *value2* it must be 4-8 characters in length, and start with an alphabetic or national character. The remaining characters can be alphabetic, national, or numeric. SSIDCMD1 and SSIDCMD2 can have the same value.

Appendix C. FM/IMS options

This section describes the parameters of the FMN1POPD, FMN1POPI and FMN1AGNT macro statements used in the FM/IMS installation options module. For more information on how to use these macro statements, see "Customizing the FM/IMS installation options module" on page 177.

FMN1POPD and FMN1POPI macros

The FMN1POPD macro statement supports all the parameters that the FMN1POPI macro statement supports except for the following:

- SSID
- DESC

The FMN1POPI macro statement supports all the parameters that the FMN1POPD macro statement supports except for the following:

- COMPAT
- LKEYxxxx
- LOGxxxx
- PSBTYPE
- RKEYxxxx
- SFMNSLIB
- UPSBTYPE
- XKEYxxxx

For the parameters that both macro statements support:

- The syntax of the parameter on the FMN1POPD macro statement is the same as the syntax of the parameter on the FMN1POPI macro statement.
- Each parameter is optional on both macro statements.
- Only the defaults are different. The default for the FMN1POPI parameter is the value specified on the FMN1POPD macro statement, while the default for the FMN1POPD macro statement is a fixed value.

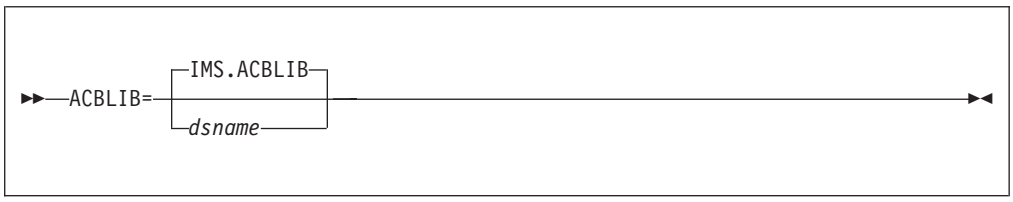
Many parameters support the specification of either a single value, or two values in parentheses separated by a comma.

For each of these parameters, when two values are specified in parentheses separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When a single value is specified, this value is used for both BMP and DLI mode.

The parameters of the FMN1POPD and FMN1POPI macro statements are described in the following sections. They are listed in alphabetical order.

© Copyright IBM Corp. 2000, 2011

ACBLIB



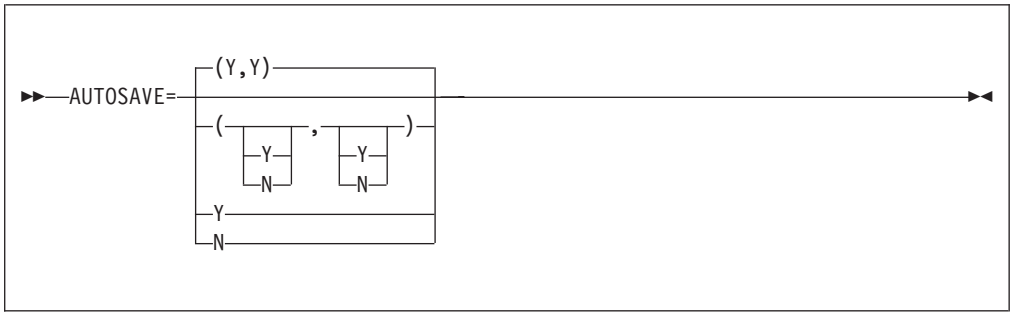
ACBLIB
Specifies the staging ACBLIB data set name. This parameter is used by the Initialize function when a Fast Path database is specified.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or IMS.ACBLIB.

If you do not want the user to override the value you specify for this parameter, set UACBLIB=N.

The ACBLIB parameter was new in FM/IMS version 8.

AUTOSAVE



AUTOSAVE
Specifies whether or not the automatic save function is set on during an Edit.

Y The automatic save function is set on during an Edit.

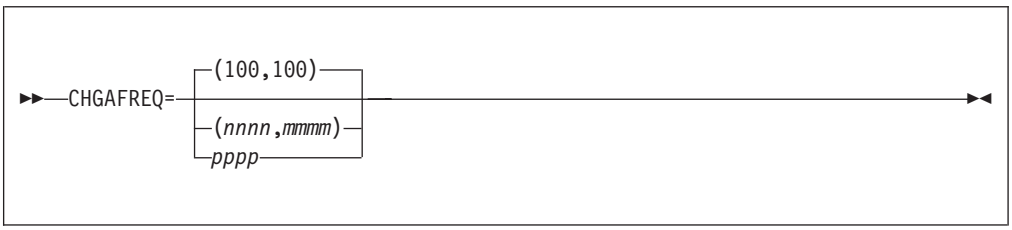
N The automatic save function is not set on during an Edit.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(Y,Y)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set UAUTOSAV=N.

CHGAFREQ



CHGAFREQ
Specifies the frequency of the automatic save function during Change All and Repeat All operations; that is, the Change All/Repeat All checkpoint frequency.

When AUTOSAVE is set on, FM/IMS:
- Increments a count by 1 each time a segment is updated during a Change All operation or inserted during a Repeat All operation.
- Issues a checkpoint when the count is equal to the Change All/Repeat All checkpoint frequency.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(100,100)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set UAUTOSAV=N.

COMPAT

COMPAT
Specifies whether or not FM/IMS accepts the absence of the IMSID parameter when a batch job runs in DLI mode.

N The IMSID parameter must be specified in the FMIMSIN input of batch functions that run in DLI mode.

Y The IMSID parameter in the FMIMSIN input of batch functions that run in DLI mode is optional.

Prior to version 8, FM/IMS did not generate an IMSID parameter when the batch function ran in DLI mode.

Therefore, if you are migrating to FM/IMS version 11 from a version of FM/IMS earlier than version 8, and you specify (or default to) COMPAT=N, some FM/IMS

batch job steps that ran with the version you are migrating from will require modification before they can be run with FM/IMS version 11.

In particular, you will need to add the IMSID parameter to FM/IMS job steps:
- That execute the IXB, ILB, IPR, IEB or IBB function and that specify REGNTYPE=DLI.
- That execute the DIB function.

If you specify COMPAT=Y, JCL generated by versions of FM/IMS earlier than version 8 should run without modification.

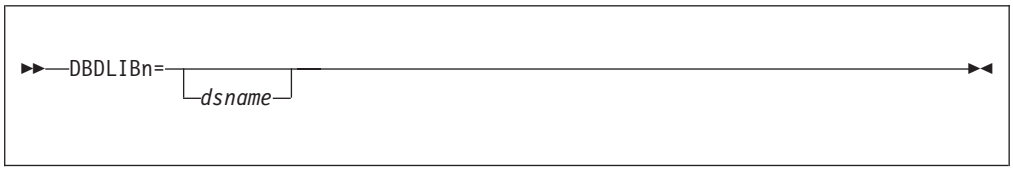
However, COMPAT=N is recommended.

This parameter can only be specified on the FMN1POPD macro statement.

This parameter is optional. The default is N. The user cannot override the value you specify for this parameter.

The COMPAT parameter was new in FM/IMS version 8.

DBDLIB*n*



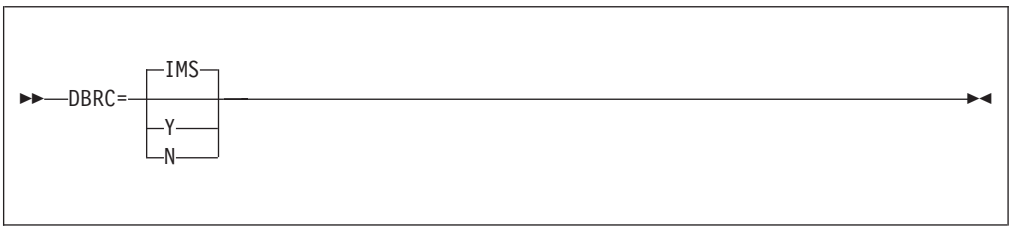
DBDLIB*n*
Where *n* is a number in the range 1 to 6, specifies the names of the data sets containing the DBDs for the databases that FM/IMS functions are to access.

These parameters are optional. But if you specify a DBDLIB parameter, you must also specify the DBDLIB parameters that precede it. So, for example, if you specify the DBDLIB3 parameter, you must also specify the DBDLIB1 parameter and the DBDLIB2 parameter.

If no DBDLIB parameters are specified on the FMN1POPI macro statement, FM/IMS uses either the DBDLIB values are specified on the FMN1POPD macro statement (if specified on that statement) or IMS.DBDLIB (otherwise).

If you do not want the user to override the values you specify for these parameters, set UDBDLIB=N.

DBRC



DBRC Specifies whether or not FM/IMS functions use Database Recovery Control (DBRC) when running in DLI mode.

IMS DBRC is used if the IMSCTRL macro statement specifies DBRC=YES or DBRC=FORCE. Otherwise, DBRC is not used.

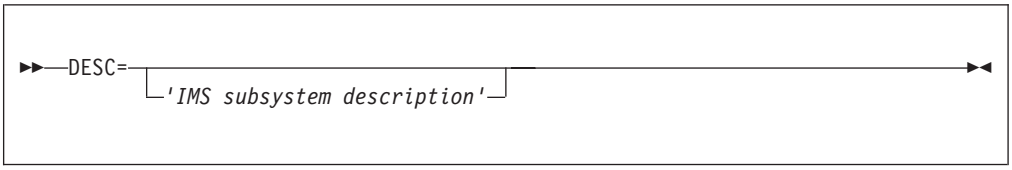
Y DBRC is used.

N DBRC is not used, unless the IMSCTRL macro specifies DBRC=FORCE.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or IMS (otherwise).

If you do not want the user to override the value you specify for this parameter, set UDBRC=N.

DESC



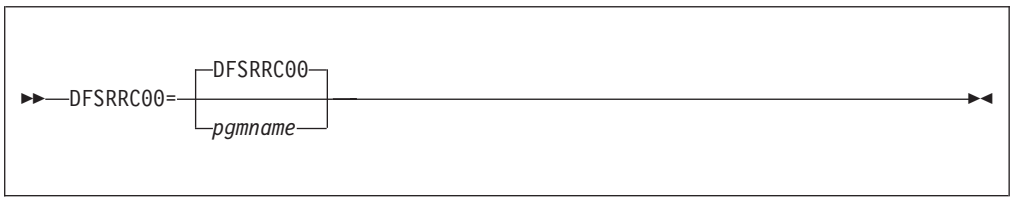
DESC Specifies a description of the IMS subsystem, to be displayed on the Subsystem Selection panel. The maximum length of the description is 30 characters. The description should be enclosed in quotation marks.

This parameter can only be specified on the FMN1POPI macro statement.

This parameter is optional. The default is no description. The user cannot override the value you specify for this parameter.

Note: In versions of FM/IMS prior to version 8, the DESC parameter could be up to 45 characters long.

DFSRRC00



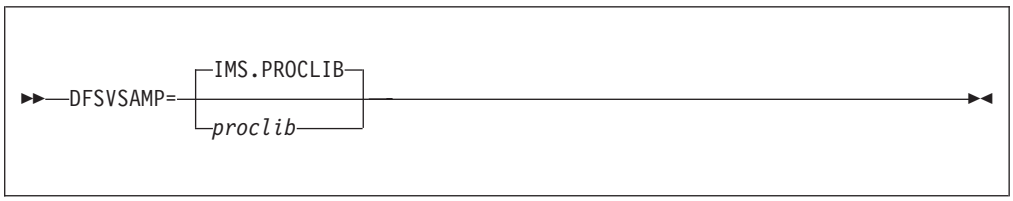
DFSRRC00
Specifies the name of the program that FM/IMS attaches to invoke IMS. This will usually be the IMS region controller program, DFSRRC00.

If a different program is to be used, it must be in one of the RESLIB data sets. For the RESLIB parameters see "RESLIB*n*" on page 386.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or DFSRRC00 (otherwise).

The user cannot override the value you specify for this parameter.

DFSVSAMP



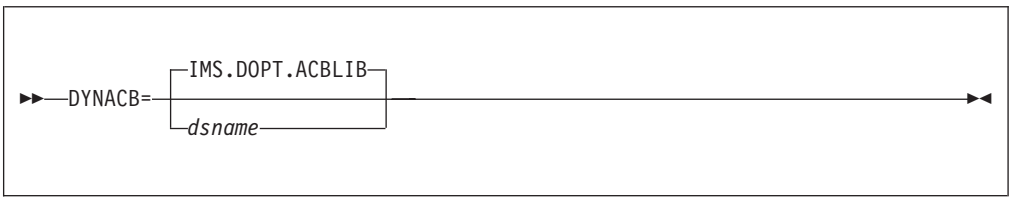
DFSVSAMP
Specifies the name of the data set that contains the parameters defining the IMS buffer sub pools for VSAM and OSAM data sets. When an FM/IMS function is run in DLI mode, the specified data set is allocated to the DFSVSAMP DD.

If the specified data set is a PDS or PDSE, specify the required member name of this data set in the VSMPMEM parameter.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or IMS.PROCLIB.

If you do not want the user to override the value you specify for this parameter, set UDFSVSMP=N.

DYNACB



DYNACB
Specifies the name of the DOPT ACBLIB data set into which FM/IMS dynamically builds PSBs. This parameter is used by functions that use dynamic PSBs to access databases in BMP mode.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or IMS.DOPT.ACBLIB.

The user cannot override the value you specify for this parameter.

DYNALLOC

DYNALLOC
Specifies whether or not FM/IMS enforces usage of the database data sets specified in the dynamic allocation modules, when accessing databases in DLI mode.

N The user is not forced to use the database data sets specified in the dynamic allocation modules when accessing databases in DLI mode.

Y The user is forced to use the database data sets specified in the dynamic allocation modules when accessing databases in DLI mode.

If you specify DYNALLOC=Y:
- The Database Data Set Display panel is displayed in the FM/IMS dialog when a DLI region type is selected on the Entry panel.
- The JCL generated by FM/IMS dialogs does not include DD statements for the database data sets.
- Batch functions running in DLI mode will fail if their JCL includes DD statements for the database data sets.
- If, for an Edit or Browse in DLI mode, the database data set DD names are allocated to your TSO session prior to the DL/I batch processing region being started, FM/IMS frees the data sets allocated to these DD names.
- The database data sets are dynamically allocated by IMS.

If you specify DYNALLOC=N:

- The Database Data Set Specification panel is displayed in the FM/IMS dialog when a DLI region type and a non-HALDB database is specified on the Entry panel.
- The JCL generated by FM/IMS dialogs includes DD statements for non-HALDB database data sets.
- In an Edit or Browse, FM/IMS allocates the database data sets specified on the Database Data Set Specification panel.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or N (otherwise).

The user cannot override the value you specify for this parameter.

To ensure that the database data sets specified in the dynamic allocation modules are used when the subsystem is selected, you should also set URESLIB=N and URECON=N.

The DYNALLOC parameter was new in FM/IMS version 8.

DYNPRFN

DYNPRFN
Specifies the number of dynamic PSBs required by concurrent FM/IMS BMP users.

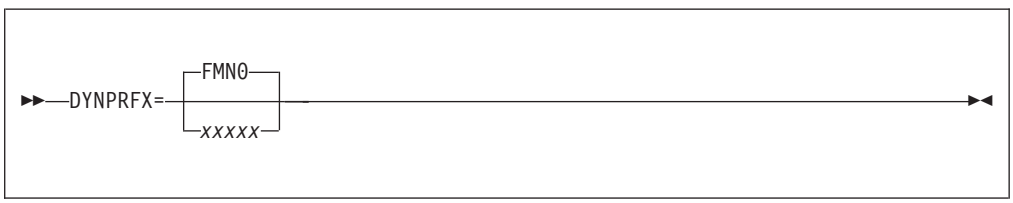
The number must be in the range 1 to 999.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or 50 (otherwise).

The user cannot override the value you specify for this parameter.

Note: If you want to run functions that use dynamic PSBs in BMP mode, you need to specify APPLCTN macro statements for each dynamic PSB name that FM/IMS is to use. (See "Customizing IMS to support the use of dynamic PSBs in BMP mode" on page 174.)

DYNPRFX



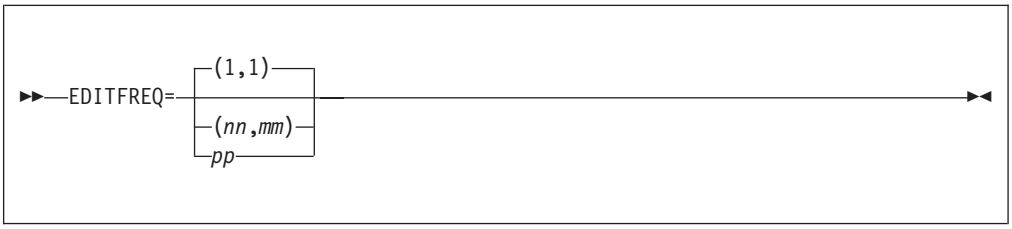
DYNPRFX
Specifies the first 1- to 5-characters of the dynamic PSB names used by FM/IMS functions when run in BMP mode.

For BMP mode, the dynamic PSB names are obtained by combining the value you specify in this parameter with a three digit number in the range 001 to the number specified in the DYNPRFN parameter. For example, if you specify DYNPRFX=FMN and DYNPRFN=3, then FM/IMS uses FMN001, FMN002 and FMN003 for the dynamic PSB names when functions are run in BMP mode.

Note: If you want to run functions that use dynamic PSBs in BMP mode, you need to specify APPLCTN macro statements for each dynamic PSB name that FM/IMS is to use. (See "Customizing IMS to support the use of dynamic PSBs in BMP mode" on page 174.)

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or FMN0 (otherwise). The user cannot override the value you specify for this parameter.

EDITFREQ



EDITFREQ
Specifies the frequency of the automatic save function when editing databases; that is, the Edit checkpoint frequency.

When AUTOSAVE is set on, FM/IMS:
- Increments a count by 1 each time an action key is pressed and data changes have been requested.
- Issues a checkpoint when the count is equal to the Edit checkpoint frequency.

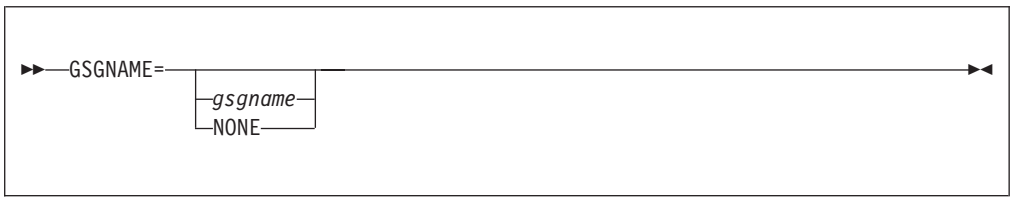
The valid range is 1 to 99.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(1,1)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set UAUTOSAV=N.

GSGNAME



GSGNAME

If the IMS subsystem is part of a Remote Site Recovery (RSR) complex and the GSGNAME has not been specified in the IMSCTRL macro statement, and you want any activity performed by FM/IMS functions running in DLI mode to be tracked by RSR, specify the Global Service Group (GSG) name for the RSR complex.

If the IMS subsystem is part of an RSR complex and the GSGNAME has been specified in the IMSCTRL macro statement and you do not want the activity of FM/IMS functions running in DLI mode to be tracked, specify NONE.

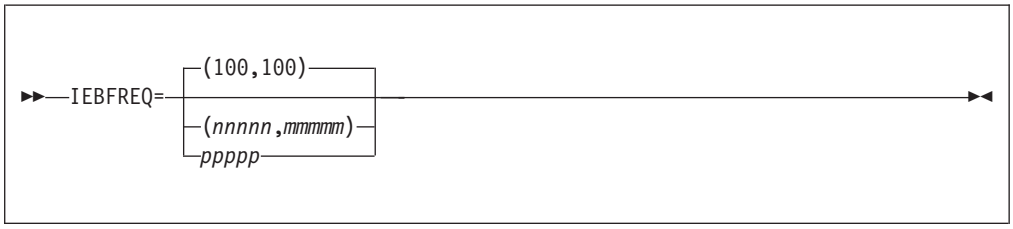
Otherwise, do not specify a value for GSGNAME.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or a null value (otherwise). When a null value is passed to the IMS region controller, IMS uses:

- The value specified for the GSGNAME parameter on the IMSCTRL macro statement (if specified on that statement), or NONE (otherwise).

If you do not want the user to override the value you specify for this parameter, set URSR=N.

IEBFREQ



IEBFREQ

Specifies the frequency of the automatic save function when running a Batch Edit job; that is, the Batch Edit checkpoint frequency.

FM/IMS:

- Increments a count by 1 each time a Batch Edit job inserts, deletes or replaces a segment.
- Issues a checkpoint when the count is equal to the Batch Edit checkpoint frequency.

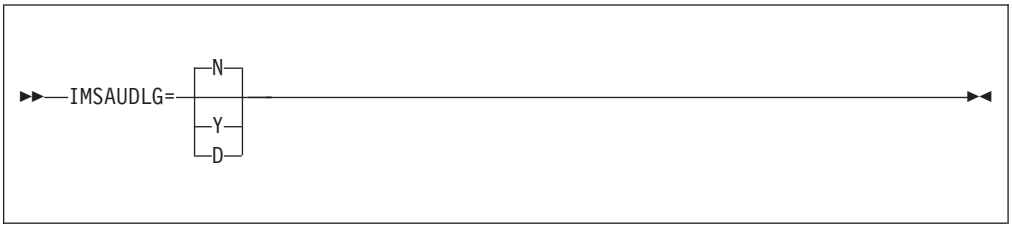
The valid range is 1 to 99999.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(100,100)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set UIEBFREQ=N.

IMSAUDLG



IMSAUDLG
Specifies whether or not audit logging is enforced during Edit.

N Audit logging is not enforced during Edit.

Y Audit logging is enforced during Edit.

D Audit logging is enforced during edit, and at the conclusion of the edit session an audit report job is submitted to report on the changes. This job can be customized by changing member FMN1FTAD from FMN.SFMNSLIB to specify the job card and the reporting options you require.

Note: FM/IMS only uses the value specified for this parameter, when the FMAUDIT parameter in the FMN1PARM member in the SYS1.PARMLIB (or its concatenations) specifies SAF_CTRL=NO. When the FMAUDIT parameter specifies SAF_CTRL=YES, FM/IMS audit logging is controlled by SAF and FM/IMS ignores this parameter.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or N (otherwise). This is true if FM/IMS does not call an Audit trail exit.

If FM/IMS calls an Audit trail exit, the value returned by the exit overrides the value specified on the FMN1POPI and FMN1POPD macro statements.

IMSBKO

IMSBKO
Specifies whether or not a dynamic backout is to be performed when an IMS pseudoabend occurs in an FM/IMS function running in DLI mode.

Y A dynamic backout is performed when an IMS pseudoabend occurs in an FM/IMS function running in DLI mode.

If you specify IMSBKO=Y, users will be able to use the UNDO or CANCEL command when editing a database in DLI mode.

N A dynamic backout is not performed when an IMS pseudoabend occurs in an FM/IMS function running in DLI mode.

This parameter only takes effect when the FM/IMS function is running with an IMS log data set.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

If you do not want the user to override the value you specify for this parameter, set UIMSBKO=N.

IMSNBA

IMSNBA
Specifies the number of Fast Path database buffers to be made available in the Common Service Area (CSA) when a Fast Path region is activated. The number specified is used by FM/IMS functions when they access Fast Path databases.

The number you specify must be in the range 1 to 9999.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or 10 (otherwise).

If you do not want the user to override the value you specify for this parameter, set UIMSNBA=N.

IMSOBA

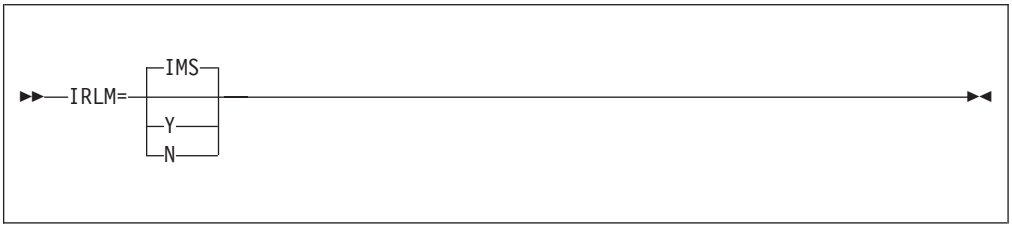
IMSOBA
Specifies the number of additional page-fixed buffers to be made available to a Fast Path region if the normal allotment (specified or defaulted in IMSNBA) is used. The number specified is used by FM/IMS functions when they access Fast Path databases.

The number you specify must be in the range 1 to 9999.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or 6 (otherwise).

If you do not want the user to override the value you specify for this parameter, set UIMSNBA=N.

IRLM



IRLM Specifies whether or not FM/IMS functions uses an Internal Resource Lock Manager (IRLM) when running in DLI mode.

IMS IRLM is used if either the IMSCTRL macro statement specifies IRLM=Y, or the IMSCTRL macro statement specifies an IRLMNM and does not specify IRLM=N.

IRLM is not used if either the IMSCTRL macro statement specifies IRLM=N or the IMSCTRL macro statement does not specify a value for IRLM and IRLMNM.

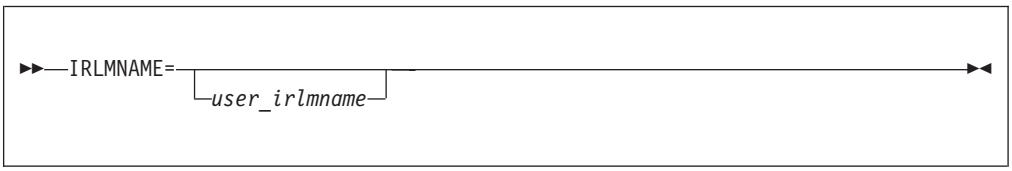
Y IRLM is used.

N IRLM is not used unless the IMSCTRL macro statement specifies IRLM=Y and an IRLMNM.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or IMS (otherwise).

If you do not want the user to override the value you specify for this parameter, set UIRLM=N.

IRLMNAME



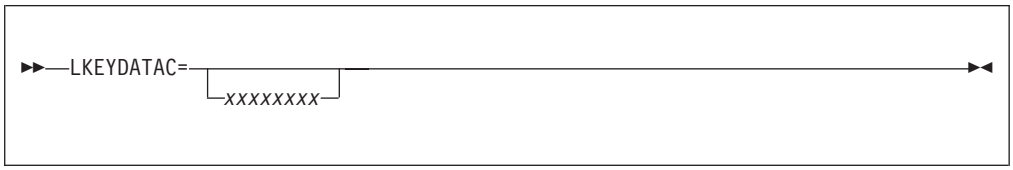
IRLMNAME
Specifies the 4-byte z/OS subsystem name assigned to the Internal Resource Lock Manager (IRLM). FM/IMS functions running in DLI mode pass the name to the IMS region controller when IRLM=Y or IRLM=IMS.

This parameter is optional. If it is not specified on the FMN1POPI macro statement and you specify IRLM=Y or IRLM=IMS, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or a null value (otherwise). When a null value is passed to the IMS region controller, IMS uses:

- The value specified for the IRLMNM parameter on the IMSCTRL macro statement (if specified on that statement), or IRLM (otherwise).

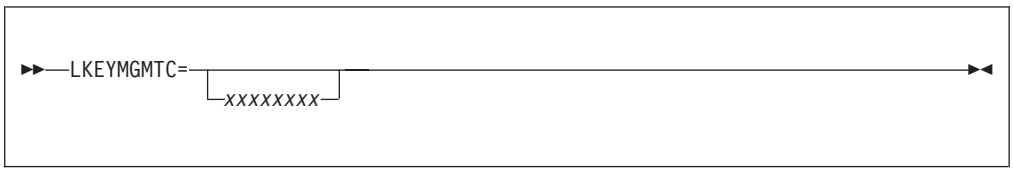
If you do not want the user to override the value you specify for this parameter, set UIRLM=N.

LKEYDATAC



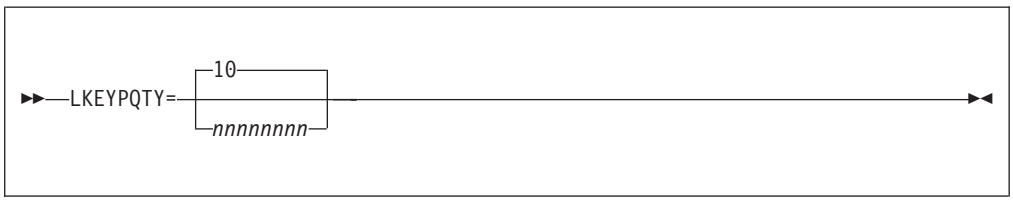
LKEYDATAC
Specifies the SMS data class that FM/IMS uses when allocating the Extract logical key VSAM work file. The maximum length you can specify for LKEYDATAC is eight (8) bytes. LKEYDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied.

LKEYMGMTC



LKEYMGMTC
Specifies the default SMS management class that FM/IMS uses when allocating the Extract logical key VSAM work file. The maximum length you can specify for LKEYMGMTC is eight (8) bytes. LKEYMGMTC is optional. If omitted the SMS management class specified in PMGMT is used, if supplied.

LKEYPQTY

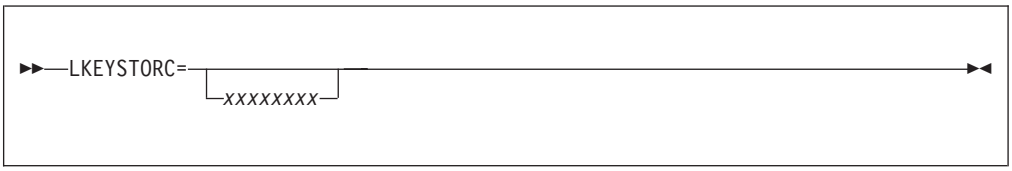


LKEYPQTY
Specifies the amount of DASD space to be used for primary space allocation of the Extract logical key VSAM work file. The range depends on the space unit specified and the DASD device type.

LKEYSQTY

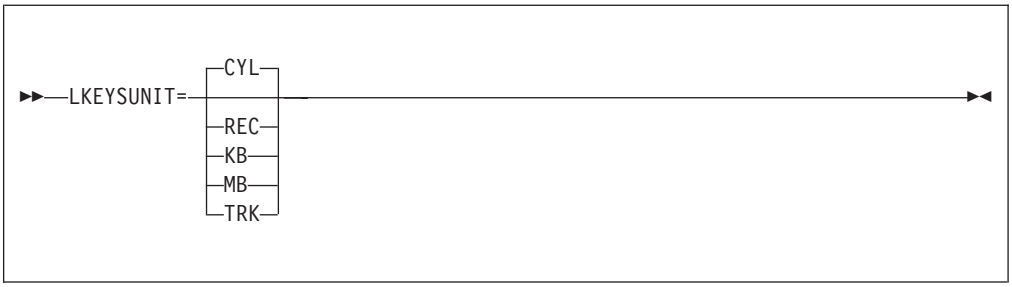
LKEYSQTY
Specifies the amount of DASD space to be used for secondary space allocation of the Extract logical key work file. The range depends on the space unit specified and the DASD device type.

LKEYSTORC



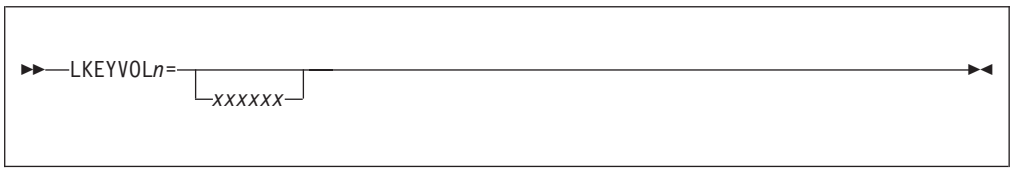
LKEYSTORC
Specifies the default SMS storage class that Fm/IMS uses when allocating the File Manager Extract logical key VSAM work file. The maximum length you can specify for LKEYSTORC is eight (8) bytes. LKEYSTORC is optional. If omitted the SMS storage class specified in PSTORC is used, if supplied.

LKEYSUNIT



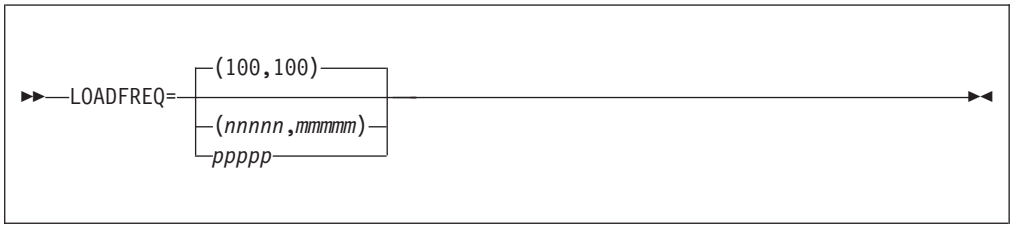
LKEYSUNIT
Specifies the unit of primary and secondary space to be allocated to the Extract logical key VSAM work file.
REC Record of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

LKEYVOL*n*



LKEYVOL*n*
Where *n* is a number in the range 1 to 3, specifies the default serial number of the volume which is to contain the Extract logical key VSAM work file. This field should only be used for sites without SMS controlled data sets as SMS overrides this value.

LOADFREQ



LOADFREQ
Specifies the frequency of the automatic save function when running a Load job; that is, the Load checkpoint frequency.

FM/IMS:
- Increments a count by 1 each time a Load job inserts or replaces a segment.
- Issues a checkpoint when the count is equal to the Load checkpoint frequency.

The valid range is 1 to 99999.

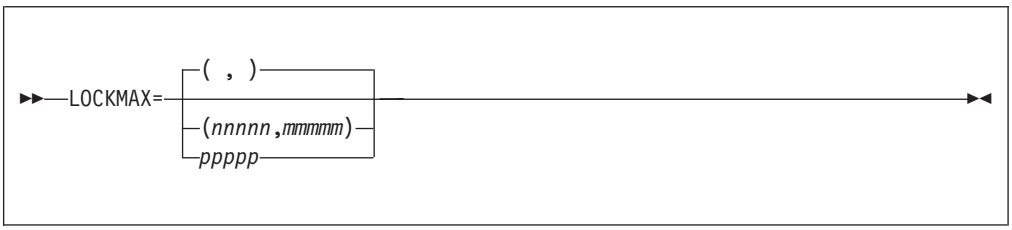
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(100,100)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set ULOADFRQ=N.

Note: The LOADFREQ parameter was new in FM/IMS version 8, and replaced the BTCHFREQ option from previous versions of FM/IMS.

LOCKMAX



LOCKMAX
Specifies the maximum number of locks (in units of 1000) that an FM/IMS function is allowed to hold at one time.

The valid range is 0 to 32767.

If you specify 0 for LOCKMAX there is no limit on the number of locks the function can hold at one time.

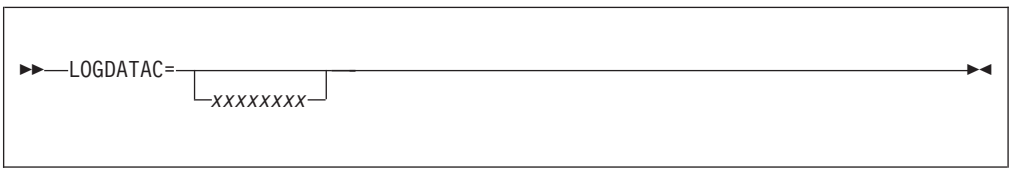
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or a null value (otherwise). When a null value is passed to the IMS region controller, IMS uses:

- The value specified for the LOCKMAX parameter on the PSBGEN macro statement (if specified on that statement), or 0 (no limit) (otherwise).

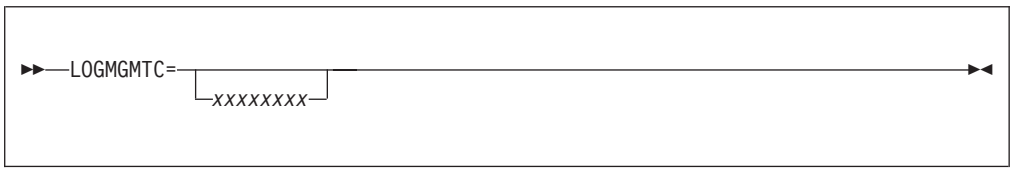
If you do not want the user to override the value you specify for this parameter, set ULOCKMAX=N.

LOGDATAC



LOGDATAC
Specifies the SMS data class that FM/IMS uses when allocating the IMS log. The maximum length you can specify for LOGDATAC is eight (8) bytes. LOGDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied.

LOGMGMTC

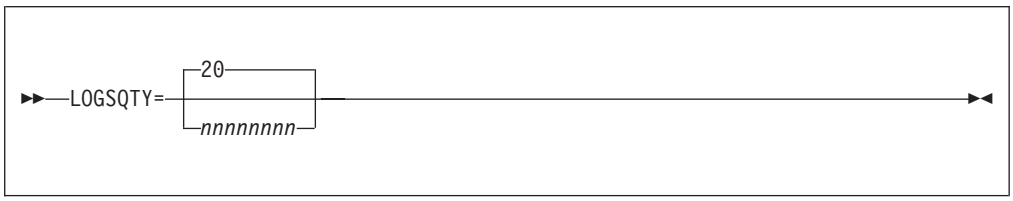


LOGMGMTC
Specifies the default SMS management class that FM/IMS uses when allocating the IMS Log. The maximum length you can specify for LOGMGMTC is eight (8) bytes. LOGMGMTC is optional. If omitted the SMS management class specified in PMGMT is used, if supplied.

LOGPQTY

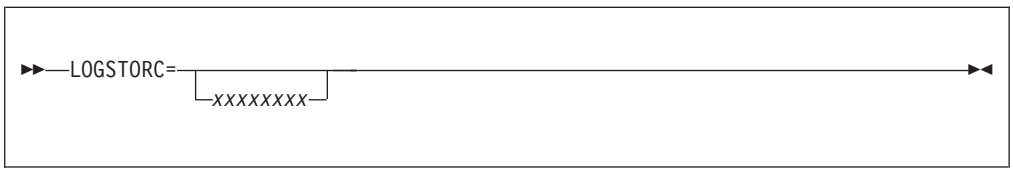
LOGPQTY
Specifies the amount of DASD space to be used for primary space allocation of the IMS Log. The range depends on the space unit specified and the DASD device type.

LOGSQTY



LOGSQTY
Specifies the amount of DASD space to be used for secondary space allocation of the IMS Log. The range depends on the space unit specified and the DASD device type.

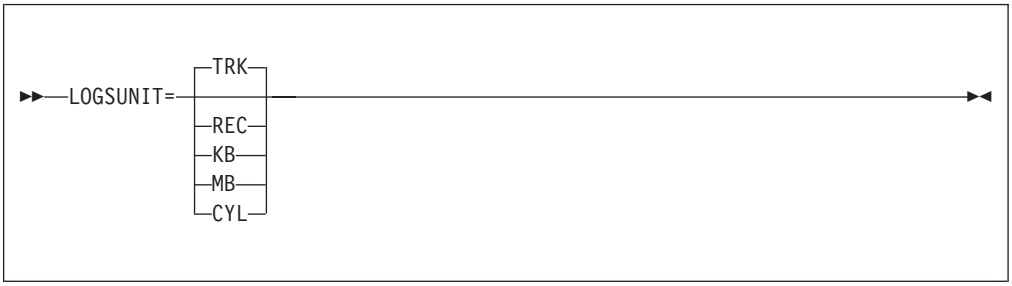
LOGSTORC



LOGSTORC
Specifies the default SMS storage class that FM/IMS uses when allocating the File Log. The maximum length you can specify for LOGSTORC is eight

(8) bytes. LOGSTORC is optional. If omitted the SMS storage class specified in PSTORC is used, if supplied.

LOGSUNIT

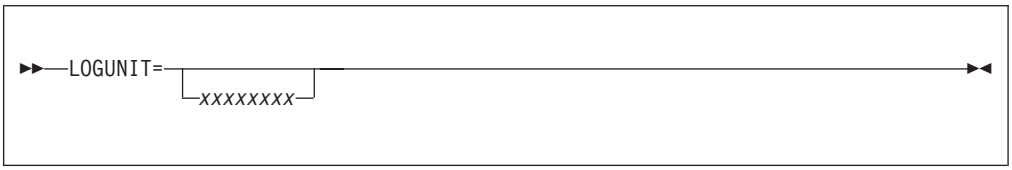


LOGSUNIT

Specifies the unit of primary and secondary space to be allocated to the IMS Log.

REC Record of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

LOGUNIT



LOGUNIT

Specifies the default permanent unit that FM/IMS when allocating the IMS Log. The maximum length you can specify for LOGUNIT is eight (8) bytes. LOGUNIT is optional. If omitted the SMS storage class specified in PUNIT is used.

MACLIB

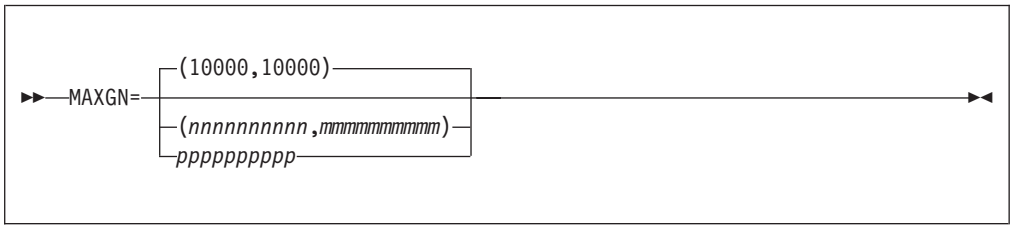
MACLIB

Specifies the name of the IMS macro library. FM/IMS uses the IMS macros in the specified library when generating a dynamic PSB.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or IMS.SDFSMAC.

The user cannot override the value you specify for this parameter, when a function is run in BMP mode. If you do not want the user to override the value you specify for this parameter, when a function is run in DLI mode, set UMACLIB=N.

MAXGN



MAXGN
Specifies the maximum number of Get Next calls that a Find/Change command can issue for a database search.

When the MAXGN limit is reached, the Search Interrupt popup is displayed, giving the user the option of continuing or discontinuing the search.

The valid range is 0 to 2147483647.

If you specify 0 for MAXGN there is no limit on the number of Get Next calls that a Find/Change command can issue.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(10000,10000)" (otherwise).

The user cannot override the values you specify for this parameter.

PARDLI

PARDLI
Specifies the parallel DL/I option used by FM/IMS functions running in BMP mode.

0 DL/I processing is performed in the BMP region.

1 All DL/I processing for the BMP is performed in the IMS control region. This prevents control region system 113 abends resulting from system X22 abends in the BMP region.

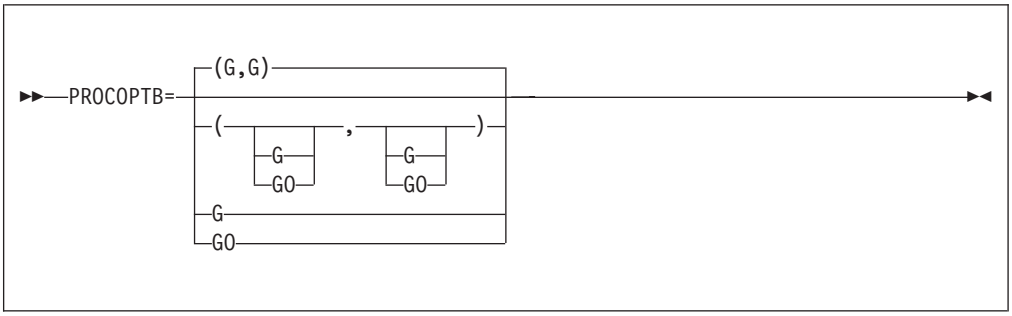
This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or 0 (otherwise).

If you do not want the user to override the value you specify for this parameter, set UPARDLI=N.

Note: If you specify PARDLI=1, parallel DL/I is disabled. This can degrade performance.

For important information about the use of PARDLI and IMS control region abends, see page 181.

PROCOPTB



PROCOPTB

Specifies the PSB processing option (PROCOPT) that a dynamic PSB generated for a Browse uses to read the database.

G The Browse reads with integrity

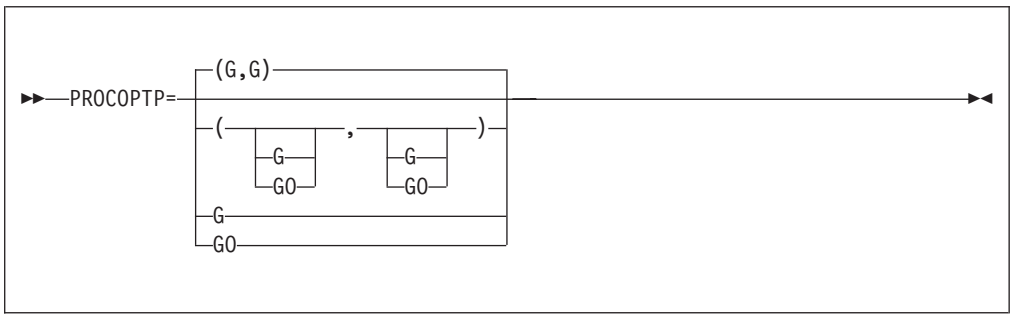
GO The Browse reads without integrity

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(G,G)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set UPROCOPB=N.

PROCOPTP



PROCOPTP
Specifies the PSB processing option (PROCOPT) that a dynamic PSB generated for a Print uses to read the database.

G The Print reads with integrity

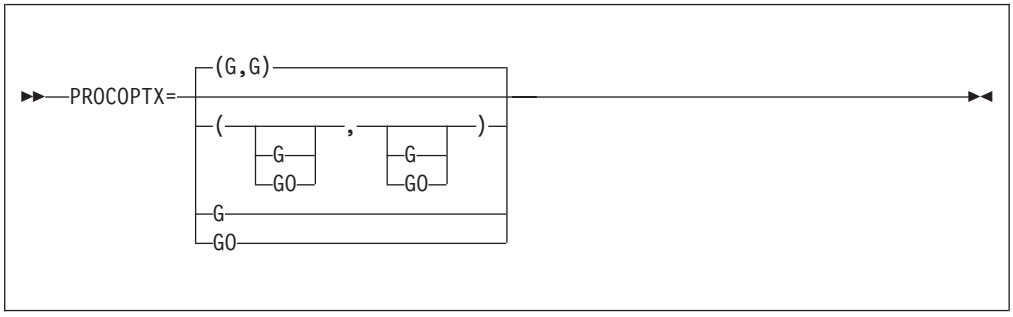
GO The Print reads without integrity

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(G,G)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set UPROCOPP=N.

PROCOPTX



PROCOPTX
Specifies the PSB processing option (PROCOPT) that a dynamic PSB generated for an Extract uses to read the database.

G The Extract reads with integrity

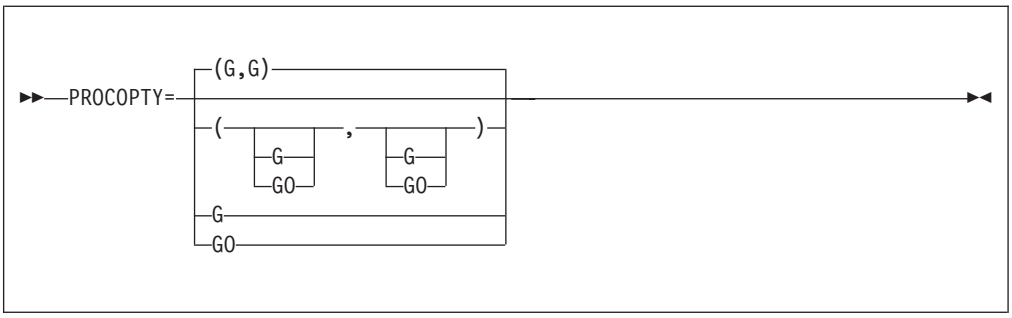
GO The Extract reads without integrity

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(G,G)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set UPROCOPX=N.

PROCOPTY



PROCOPTY
Specifies the PSB processing option (PROCOPT) that a dynamic PSB generated for a Batch Browse uses to read the database.

G The Batch Browse reads with integrity.

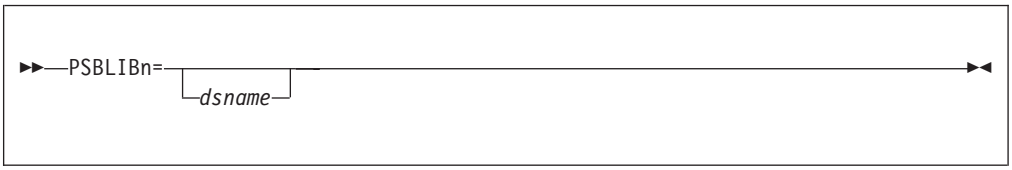
GO The Batch Browse reads without integrity.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(G,G)" (otherwise).

If you do not want the user to override the value you specify for this parameter, set UPROCOPY=N.

PSBLIB*n*



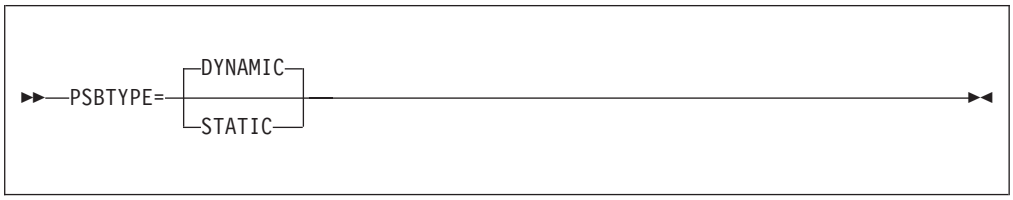
PSBLIB*n*
Where *n* is a number in the range 1 to 6, specifies the names of the data sets containing the static PSBs that FM/IMS functions are to use.

These parameters are optional. But if you specify a PSBLIB parameter, you must also specify the PSBLIB parameters that precede it. So, for example, if you specify the PSBLIB3 parameter, you must also specify the PSBLIB1 parameter and the PSBLIB2 parameter.

If no PSBLIB parameters are specified on the FMN1POPI macro statement, FM/IMS uses either the PSBLIB values specified on the FMN1POPD macro statement (if specified on that statement), or IMS.PSBLIB (otherwise).

If you do not want the user to override the values you specify for these parameters, set UPSBLIB=N.

PSBTYPE



PSBTYPE
Specifies the type of PSB that the FM/IMS Extract, Load, Print, Batch Edit and Batch Browse functions use to access databases when no PSBTYPE= statement is specified in the FMNIMSIN input for the function.

DYNAMIC
The batch functions use a temporary PSB that they build when they are invoked.

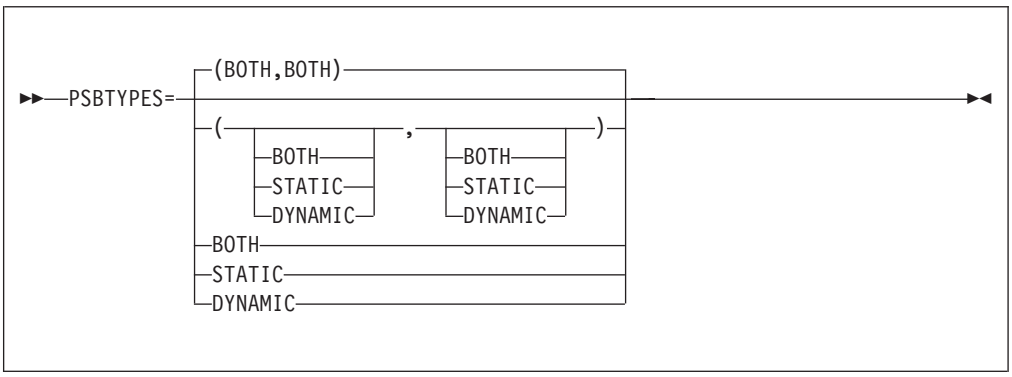
STATIC
The batch functions use an existing PSB that the user specifies in the JCL.

This parameter can only be specified on the FMN1POPD macro statement.

This parameter is optional. The default is DYNAMIC.

If you do not want the user to override the value you specify for this parameter, set UPSBTYPE=N.

PSBTYPES



PSBTYPES
Specifies the PSB types that can be used to access databases in the IMS subsystem.

BOTH Both dynamic and static PSBs can be used.

STATIC
Only static PSBs can be used.

DYNAMIC
Only dynamic PSBs can be used.

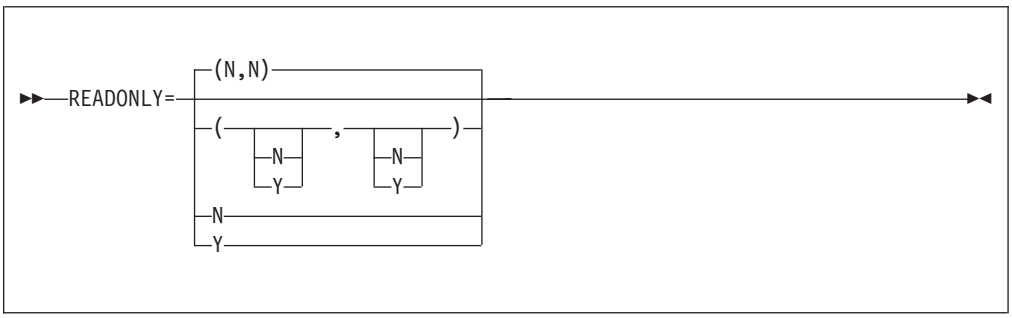
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(BOTH,BOTH)" (otherwise).

The user cannot override the value you specify for this parameter.

The PSBTYPES parameter was new in FM/IMS version 8.

READONLY



READONLY
Specifies whether or not the IMS subsystem is to be defined as Read Only.

When an IMS subsystem is defined read-only, only the FM/IMS non-update functions, that is, Browse, Batch Browse, Extract and Print, can

be run against databases in the IMS subsystem; FM/IMS prevents the user running the update functions, that is, Edit, Batch Edit, Load, Initialize and Delete/Define.

N The IMS subsystem is not to be defined as Read Only.

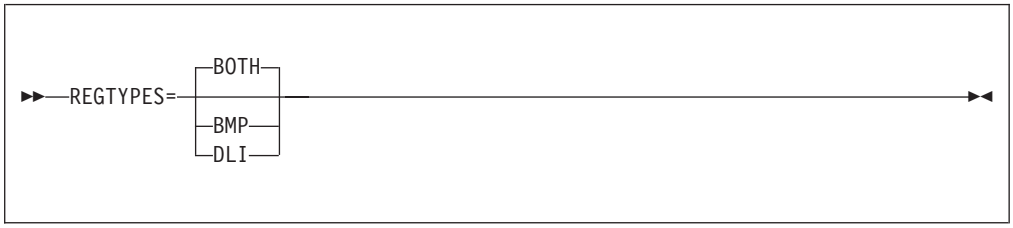
Y The IMS subsystem is to be defined as Read Only.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,N)" (otherwise).

The user cannot override the value you specify for this parameter.

REGTYPES



REGTYPES

Specifies the modes in which databases in the IMS subsystem can be accessed.

BOTH Databases in the IMS subsystem can be accessed in both BMP mode and DLI mode.

BMP Databases in the IMS subsystem can be accessed in BMP mode only.

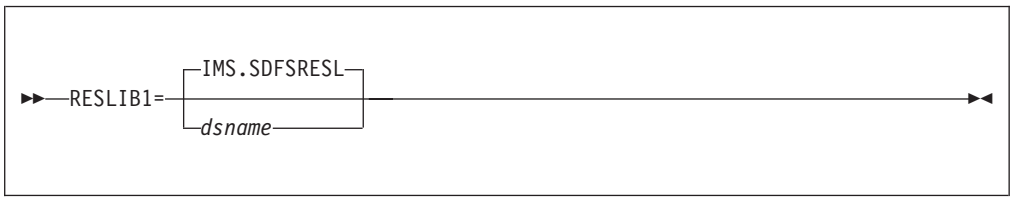
DLI Databases in the IMS subsystem can be accessed in DLI mode only.

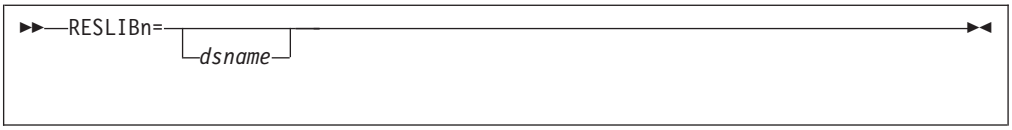
This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or BOTH (otherwise).

The user cannot override the value you specify for this parameter.

The REGTYPES parameter was new in FM/IMS version 8.

RESLIB*n*





RESLIB*n*

Where *n* is a number in the range 1 to 6, specifies the names of the data sets containing the IMS SVC modules, the DFSMDA dynamic allocation modules and the IMS exit routines. When an FM/IMS function accesses the subsystem, the specified data sets are allocated to the TASKLIB DD and those data sets that are APF-authorized are allocated to the DFSRESLB DD.

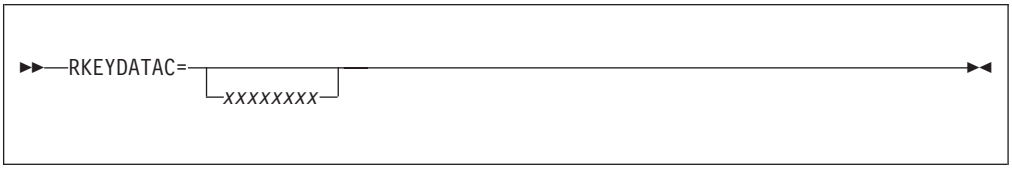
These parameters are optional. But if you specify a RESLIB parameter, you must also specify the RESLIB parameters that precede it. So, for example, if you specify the RESLIB3 parameter, you must also specify the RESLIB1 parameter and the RESLIB2 parameter.

If no RESLIB parameters are specified on the FMN1POPI macro statement, FM/IMS uses either the RESLIB values specified on the FMN1POPD macro statement (if specified on that statement), or IMS.SDFSRESL (otherwise).

The user cannot override the values you specify for these parameters when a function is run in BMP mode.

If you do not want the user to override the values you specify for these parameters, when a function is run in DLI mode, set URESLIB=N.

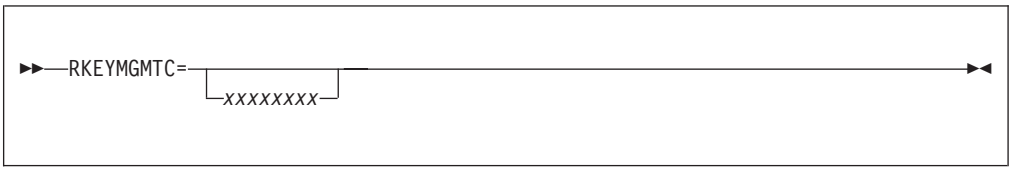
RKEYDATAC



RKEYDATAC

Specifies the SMS data class that FM/IMS uses when allocating the Extract root key VSAM work file. The maximum length you can specify for RKEYDATAC is eight (8) bytes. RKEYDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied.

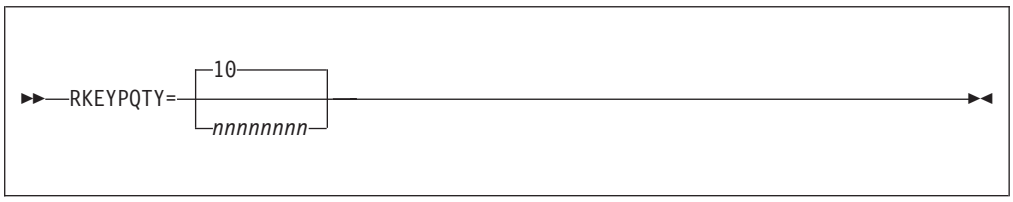
RKEYMGMTC



RKEYMGMTC

Specifies the default SMS management class that FM/IMS uses when allocating the Extract root key VSAM work file. The maximum length you can specify for RKEYMGMTC is eight (8) bytes. RKEYMGMTC is optional. If omitted the SMS management class specified in PMGMT is used, if supplied.

RKEYPQTY

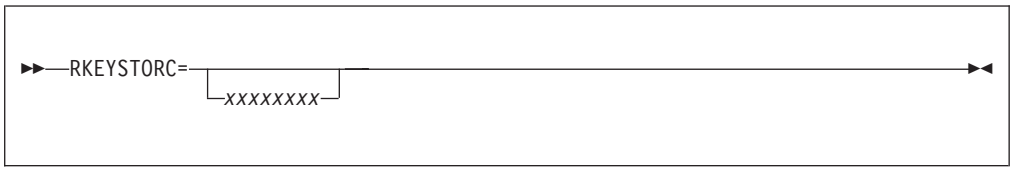


RKEYPQTY
Specifies the amount of DASD space to be used for primary space allocation of the Extract root key VSAM work file. The range depends on the space unit specified and the DASD device type.

RKEYSQTY

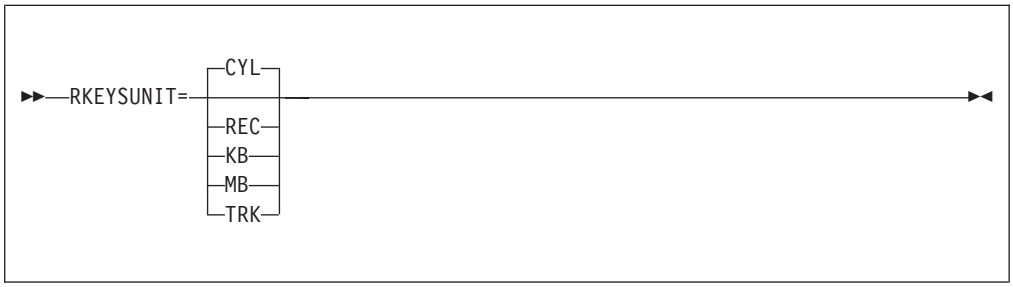
RKEYSQTY
Specifies the amount of DASD space to be used for secondary space allocation of the Extract root key VSAM work file. The range depends on the space unit specified and the DASD device type.

RKEYSTORC



RKEYSTORC
Specifies the default SMS storage class that FM/IMS uses when allocating the Extract root key VSAM work file. The maximum length you can specify for RKEYSTORC is eight (8) bytes. RKEYSTORC is optional. If omitted the SMS storage class specified in PSTORC is used, if supplied.

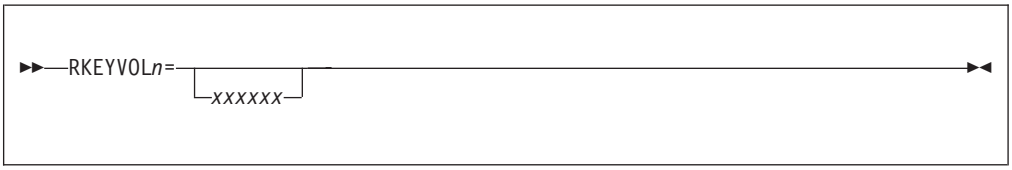
RKEYSUNIT



RKEYSUNIT
Specifies the unit of primary and secondary space to be allocated to the Extract root key VSAM work file.

REC Record of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

RKEYVOL*n*



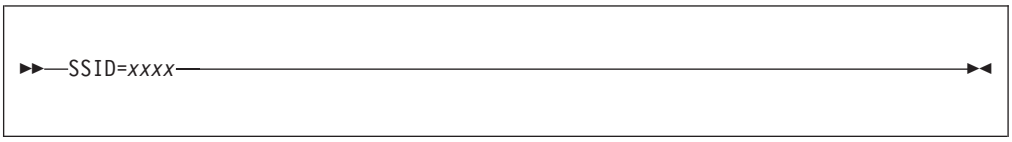
RKEYVOL*n*
Where *n* is a number in the range 1 to 3, specifies the default serial number of the volume which is to contain the Extract root key VSAM work file. This field should only be used for sites without SMS controlled data sets as SMS overrides this value.

SFMNSLIB

SFMNSLIB
The name of the File Manager target skeleton library.

This parameter is used in the FMN1FTEX job control skeleton. This parameter can only be specified on the FMN1POPD macro statement. The parameter is optional.

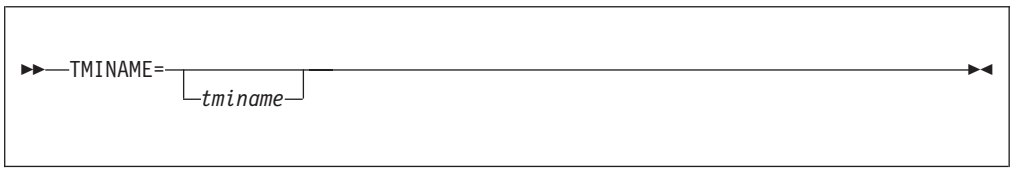
SSID



SSID Specifies the 1 to 4-character IMS subsystem identifier.

This parameter is required, and can only be specified on the FMN1POPI macro statement.

TMINAME



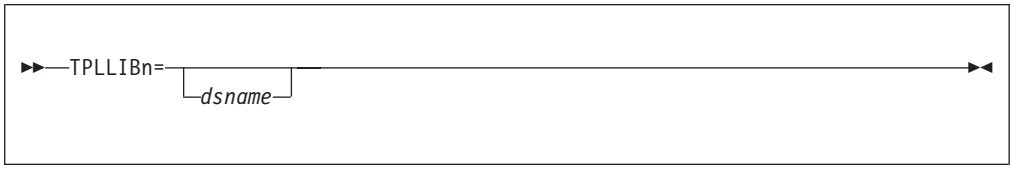
TMINAME
If the IMS subsystem is part of a Remote Site Recovery (RSR) complex and the activity performed by FM/IMS functions running in DLI mode is to be tracked by RSR, specify the one to four-character Transport Manager Instance (TMI) name that the functions are to use.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or a null value (otherwise).

When the activity is tracked by RSR and a null value is passed to the IMS region controller, IMS uses:
- The value specified for the TMINAME parameter on the IMSCTRL macro statement (if specified on that statement), or blanks (otherwise).

If you do not want the user to override the value you specify for this parameter, set URSR=N.

TPLLIB*n*



TPLLIB*n*
Where n is a number in the range 1 to 6, specifies the names of the data sets containing the templates that FM/IMS functions are to use.

These parameters are optional. But if you specify a TPLLIB parameter, you must also specify the TPLLIB parameters that precede it. So, for example, if you specify the TPLLIB3 parameter, you must also specify the TPLLIB1 parameter and the TPLLIB2 parameter.

If no TPLLIB parameters are specified on the FMN1POPI macro statement, FM/IMS uses either the TPLLIB values specified on the FMN1POPD macro statement (if specified on that statement), or FMN.TEMPLATE (otherwise).

If you do not want the user to override the values you specify for these parameters, set UTPLLIB=N.

UACBLIB

UACBLIB
Specifies whether or not the user can override the data set specified in the ACBLIB parameter.

Y The user can override the ACBLIB parameter.

N The user cannot override the ACBLIB parameter.

If you specify UACBLIB=N, the ACBLIB **Data set name** field on the DLI Mode Data Set 2 panel is protected.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

The UACBLIB parameter was new in FM/IMS version 8.

UAGNS

UAGNS
Specifies whether or not the IMS subsystem uses AGNs to secure dependent regions.

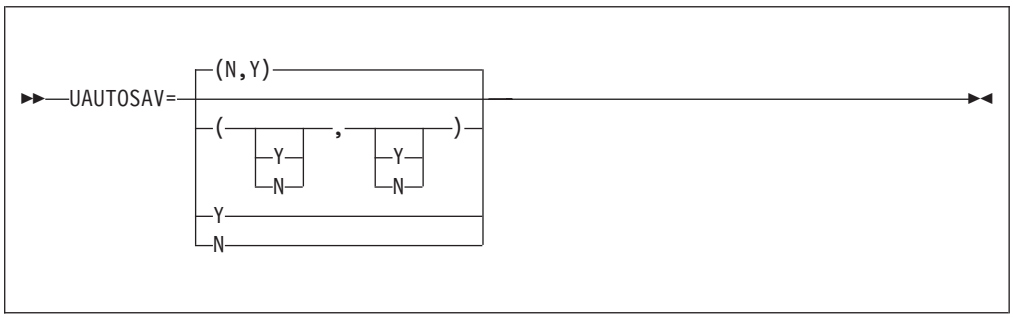
N The IMS subsystem does not use AGNs to secure dependent regions.

Y The IMS subsystem uses AGNs to secure dependent regions.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or N (otherwise).

The user cannot override the value you specify for this parameter.

UAUTOSAV



UAUTOSAV
Specifies whether or not the user can override the values specified for the AUTOSAVE, CHGAFREQ and EDITFREQ parameters.

N The user cannot override the parameters.

Y The user can override the parameters.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

If you specify UAUTOSAV=N for BMP mode, the Autosave, Edit Checkpoint Frequency, and Change All/Repeat All Checkpoint Frequency fields on the BMP Mode Options panel are protected.

If you specify UAUTOSAV=N for DLI mode, the Autosave, Edit Checkpoint Frequency, and Change All/Repeat All Checkpoint Frequency fields on the DLI Mode Options panel are protected.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

The UAUTOSAV parameter was new in FM/IMS version 8.

UBUF

UBUF Specifies whether or not the user can specify the BUF parameter passed to the IMS region controller when FM/IMS functions run in DLI mode.

Y The user can specify the BUF parameter.

N The user cannot specify the BUF parameter.

If you specify UBUF=N:
- The **BUF** parameter field on the DLI Mode Parameters panel is protected.

- IMSBUF= statements in the FMIMSIN input of batch jobs are ignored.
- No IMSBUF= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

The UBUF parameter was new in FM/IMS version 8.

UDBDLIB

UDBDLIB
Specifies whether or not the user can override the data sets specified in the DBDLIBn parameters.

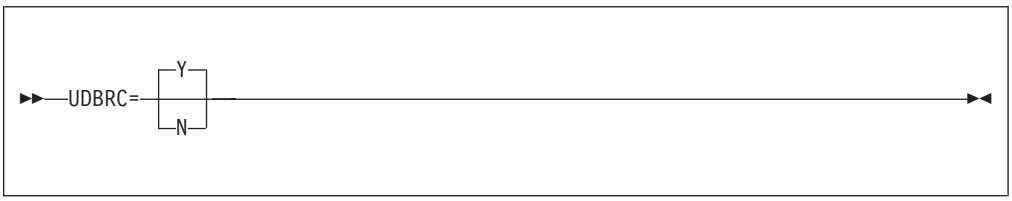
Y The user can override the DBDLIBn data set parameters.

N The user cannot override the DBDLIBn data set parameters.

If you specify UDBDLIB=N then:
- The DBDLIB Data set name fields on the PSB and DBD Data Sets panel are protected.
- DBDDSN=, DBDDSN2=, DBDDSN3=, DBDDSN4=, DBDDSN5= and DBDDSN6= statements in the FMNIMSIN input of batch jobs are ignored.
- No DBDDSN=, DBDDSN2=, DBDDSN3=, DBDDSN4=, DBDDSN5= and DBDDSN6= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

UDBRC



UDBRC
Specifies whether or not the user can override the value specified for the DBRC parameter.

Y The user can override the DBRC parameter.

N The user cannot override the DBRC parameter.

If you specify UDBRC=N:
- The **DBRC** option field on the DLI Mode Parameters panel is protected.

- DBRC= statements in the FMIMSIN input of batch jobs are ignored.
- No DBRC= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

UDFSVSMP

UDFSVSMP
Specifies whether or not the user can override the data set specified in the DFSVSAMP parameter and the member specified in the VSMPMEM parameter.

Y The user can override the DFSVSAMP and VSMPMEM parameters.

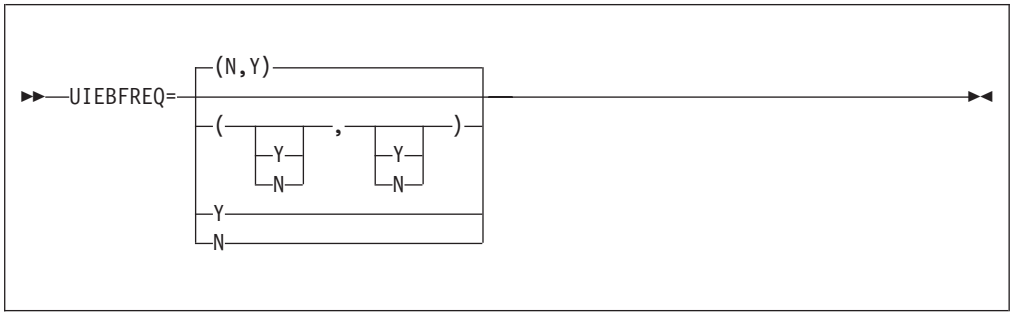
N The user cannot override the DFSVSAMP and VSMPMEM parameters.

If you specify UDFSVSMP=N:
- The DFSVSAMP **Data set name** and **Member** fields on the DLI Mode Data Sets 1 panel are protected.
- DFSVSAMP= and VSMPMEM= statements in the FMIMSIN input of batch jobs are ignored.
- No DFSVSAMP= and VSMPMEM statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

The UDFSVSMP parameter was new in FM/IMS version 8.

UIEBFREQ



UIEBFREQ
Specifies whether or not the user can override the value specified for the IEBFREQ parameter.

N The user cannot override the IEBFREQ parameter.

Y The user can override the IEBFREQ parameter.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

If you specify UIEBFREQ=N for BMP mode:
- The **Batch Edit** checkpoint frequency field on the BMP Mode Options panel is protected.
- CHKPFREQ= statements in the FMIMSIN input of Batch Edit jobs run in BMP mode are ignored.
- No CHKPFREQ= statements are included in the JCL generated by the Batch Edit dialog when the Batch Edit is to run in BMP mode.

If you specify UIEBFREQ=N for DLI mode:
- The **Batch Edit** checkpoint frequency field on the DLI Mode Options panel is protected.
- CHKPFREQ= statements in the FMIMSIN input of Batch Edit jobs run in DLI mode are ignored.
- No CHKPFREQ= statements are included in the JCL generated by the Batch Edit dialog when the Batch Edit is to run in DLI mode.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

The UIEBFREQ parameter was new in FM/IMS version 8.

UIMSBKO

UIMSBKO

Specifies whether or not the user can override the value specified for the IMSBKO parameter.

Y The user can override the IMSBKO parameter.

N The user cannot override the IMSBKO parameter.

If you specify UIMSBKO=N:
- The **Dynamic backout** option field on the DLI Mode Parameters panel is protected.
- IMSBKO= statements in the FMIMSIN input of Load and Batch Edit jobs are ignored.
- No IMSBKO= statements are included in the JCL generated by the Load and Batch Edit dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

UIMSNBA

UIMSNBA

Specifies whether or not the user can override the values specified for the IMSNBA and IMSOBA parameters.

Y The user can override the IMSNBA and IMSOBA parameters.

N The user cannot override the IMSNBA and IMSOBA parameters.

If you specify UIMSNBA=N:
- The **NBA** and **OBA** Fast Path Buffer Allocation fields on the BMP Mode Parameters panel are protected.
- IMSNBA= and IMSOBA= statements in the FMIMSIN input of batch jobs are ignored.
- No IMSNBA= and IMSOBA= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

The UIMSNBA parameter was new in FM/IMS version 8.

UIRLM

UIRLM

Specifies whether or not the user can override the value specified for the IRLM and IRLMNAME parameters.

Y The user can override the IRLM and IRLMNAME parameters.

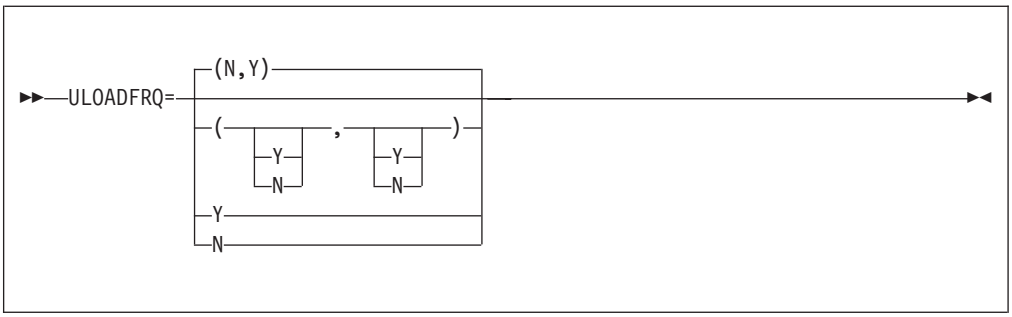
N The user cannot override the IRLM and IRLMNAME parameters.

If you specify UIRLM=N:
- The **IRLM** option and the **IRLMNM** parameter fields on the DLI Mode Parameters panel are protected.
- IRLM= and IRLMNAME= statements in the FMIMSIN input of batch jobs are ignored.

- No IRLM= and IRLMNAME= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

ULOADFRQ



ULOADFRQ
Specifies whether or not the user can override the value specified for the LOADFREQ parameter.

N The user cannot override the LOADFREQ parameter.

Y The user can override the LOADFREQ parameter.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

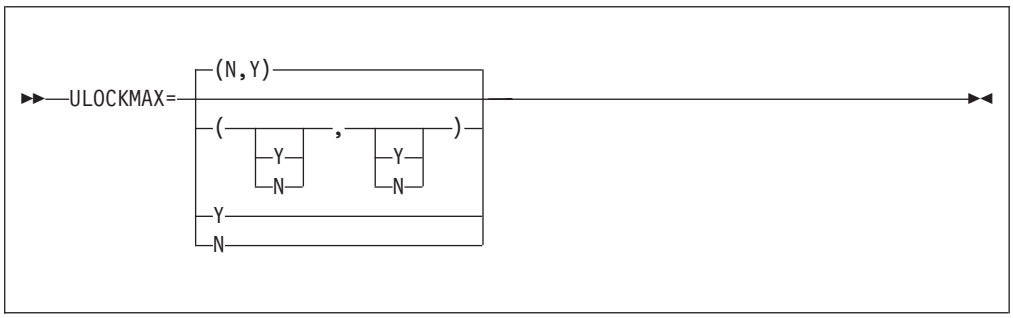
If you specify ULOADFRQ=N for BMP mode:
- The **Load** checkpoint frequency field on the BMP Mode Options panel is protected.
- CHKPFREQ= statements in the FMIMSIN input of Load jobs run in BMP mode are ignored.
- No CHKPFREQ= statements are included in the JCL generated by the Load dialog when the Load is to run in BMP mode.

If you specify ULOADFRQ=N for DLI mode:
- The **Load** checkpoint frequency field on the DLI Mode Options panel is protected.
- CHKPFREQ= statements in the FMIMSIN input of Load jobs run in DLI mode are ignored.
- No CHKPFREQ= statements are included in the JCL generated by the Load dialog when the Load is to run in DLI mode.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

The ULOADFRQ parameter was new in FM/IMS version 8.

ULOCKMAX



ULOCKMAX
Specifies whether or not the user can override the value specified for the LOCKMAX parameter.

N The user cannot override the LOCKMAX parameter.

Y The user can override the LOCKMAX parameter.

When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

If you specify ULOCKMAX=N for BMP mode:
- The **LOCKMAX** parameter field on the BMP Mode Parameters panel is protected.
- LOCKMAX= statements in the FMIMSIN input of batch jobs run in BMP mode are ignored.
- No LOCKMAX= statements are included in the JCL generated by FM/IMS dialogs when the job is to run in BMP mode.

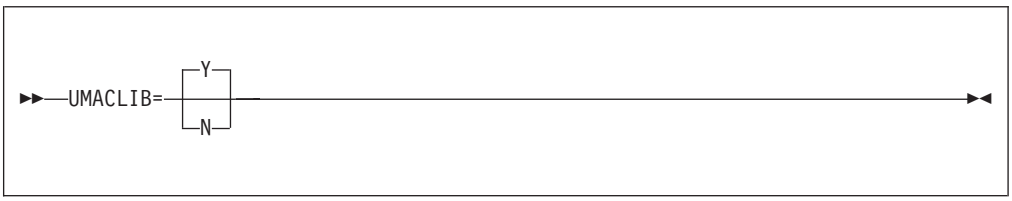
If you specify ULOCKMAX=N for DLI mode:
- The **LOCKMAX** parameter field on the DLI Mode Parameters panel is protected.
- LOCKMAX= statements in the FMIMSIN input of batch jobs run in DLI mode are ignored.
- No LOCKMAX= statements are included in the JCL generated by FM/IMS dialogs when the job is to run in DLI mode.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

The ULOCKMAX parameter was new in FM/IMS version 8.

UMACLIB



UMACLIB
Specifies whether or not the user can override the data set specified in the MACLIB parameter, when a function is run in DLI mode. (When a function is run in BMP mode, irrespective of the setting of this parameter, the user cannot override the data set specified in the MACLIB parameter.)

Y The user can override the MACLIB parameter.

N The user cannot override the MACLIB parameter.

If you specify UMACLIB=N:
- The IMS Macros **Data set name** field on the DLI Mode Data Sets 1 panel is protected.
- MACLIB= statements in the FMIMSIN input of batch jobs are ignored.
- No MACLIB= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

The UMACLIB parameter was new in FM/IMS version 8.

UPARDLI

UPARDLI
Specifies whether or not the user can override the value specified in the PARDLI parameter.

N The user cannot override the PARDLI parameter.

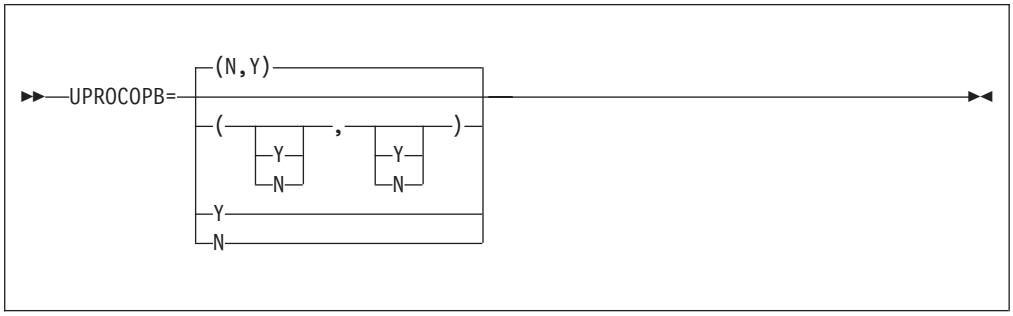
Y The user can override the PARDLI parameter.

If you specify UPARDLI=N:
- The **PARDLI** option field on the BMP Mode Parameters panel is protected.
- PARDLI= statements in the FMIMSIN input of batch jobs are ignored,
- No PARDLI= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or N (otherwise).

The UPARDLI parameter was new in FM/IMS version 8.

UPROCOPB



UPROCOPB
Specifies whether or not the user can override the value specified for the PROCOPTB parameter.

N The user cannot override the PROCOPTB parameter.

Y The user can override the PROCOPTB parameter.

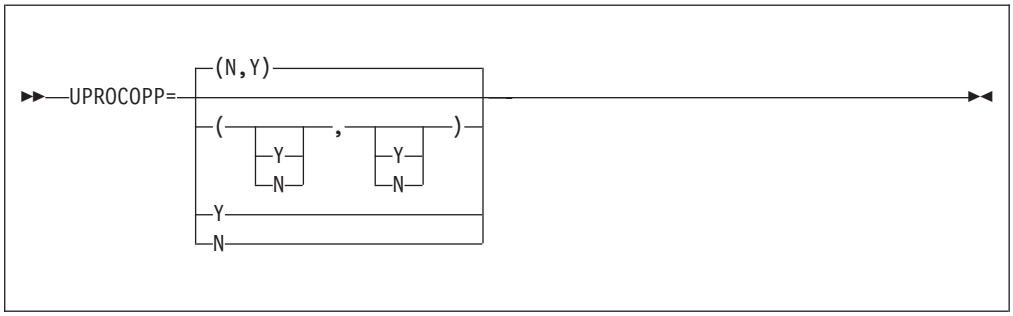
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

If you specify UPROCOPB=N for BMP mode, the **Browse** PSB Processing Option field on the BMP Mode Options panel is protected.

If you specify UPROCOPB=N for DLI mode, the **Browse** PSB Processing Option field on the DLI Mode Options panel is protected.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

UPROCOPP



UPROCOPP
Specifies whether or not the user can override the value specified for the PROCOPTP parameter.

N The user cannot override the PROCOPTP parameter.

Y The user can override the PROCOPTP parameter.

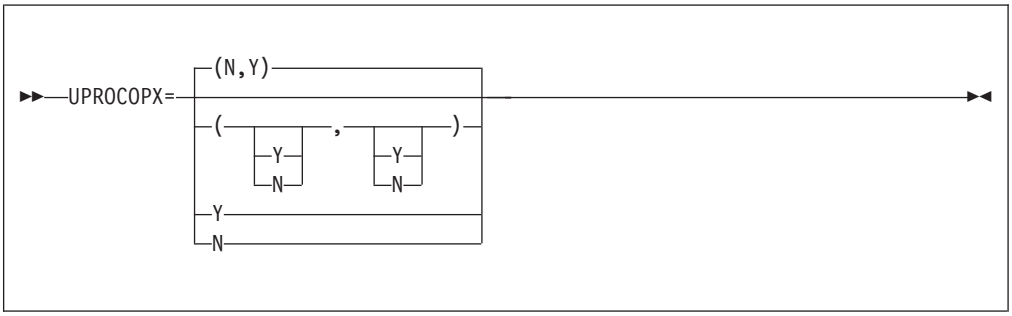
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

If you specify UPROCOPP=N for BMP mode:
- The **Print** PSB Processing Option field on the BMP Mode Options panel is protected.
- PROCOPT= statements in the FMIMSIN input of Print jobs run in BMP mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Print dialog when the Print is to run in BMP mode.

If you specify UPROCOPP=N for DLI mode:
- The **Print** PSB Processing Option field on the DLI Mode Options panel is protected.
- PROCOPT= statements in the FMIMSIN input of Print jobs run in DLI mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Print dialog when the Print is to run in DLI mode.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

UPROCOPX



UPROCOPX

Specifies whether or not the user can override the value specified for the PROCOPTX parameter.

N The user cannot override the PROCOPTX parameter.

Y The user can override the PROCOPTX parameter.

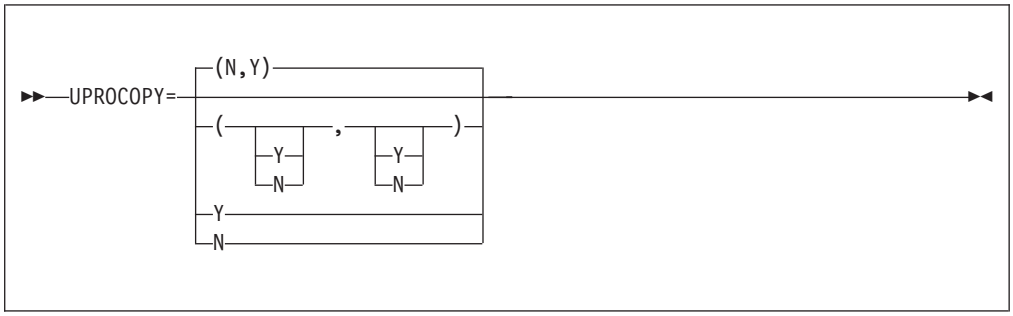
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

If you specify UPROCOPX=N for BMP mode:
- The **Extract** PSB Processing Option field on the BMP Mode Options panel is protected.
- PROCOPT= statements in the FMIMSIN input of Extract jobs run in BMP mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Extract dialog when the Extract is to run in BMP mode.

If you specify UPROCOPX=N for DLI mode:
- The **Extract** PSB Processing Option field on the DLI Mode Options panel is protected.
- PROCOPT= statements in the FMIMSIN input of Extract jobs run in DLI mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Extract dialog when the Extract is to run in DLI mode.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

UPROCOPY



UPROCOPY

Specifies whether or not the user can override the value specified for the PROCOPTY parameter.

N The user cannot override the PROCOPTY parameter.

Y The user can override the PROCOPTY parameter.

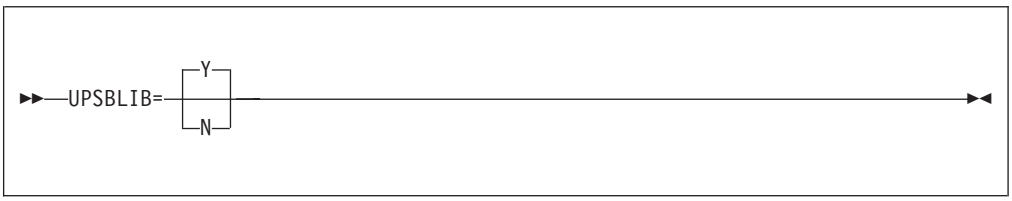
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

If you specify UPROCOPY=N for BMP mode:
- The **Batch Browse** PSB Processing Option field on the BMP Mode Options panel is protected.
- PROCOPT= statements in the FMIMSIN input of Batch Browse jobs run in BMP mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Batch Browse dialog when the Batch Browse is to run in BMP mode.

If you specify UPROCOPY=N for DLI mode:
- The **Batch Browse** PSB Processing Option field on the DLI Mode Options panel is protected.
- PROCOPT= statements in the FMIMSIN input of Batch Browse jobs run in DLI mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Batch Browse dialog when the Batch Browse is to run in DLI mode.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or "(N,Y)" (otherwise).

UPSBLIB



UPSBLIB

Specifies whether or not the user can override the data sets specified in the PSBLIBn parameters.

Y The user can override the PSBLIBn data set parameters.

N The user cannot override the PSBLIBn data set parameters.

If you specify UPSBLIB=N:
- The PSBLIB **Data set name** fields on the PSB and DBD Data Sets panel are protected.
- PSBDSN=, PSBDSN2=, PSBDSN3=, PSBDSN4=, PSBDSN5= and PSBDSN6= statements in the FMNIMSIN input of batch jobs are ignored.
- No PSBDSN=, PSBDSN2=, PSBDSN3=, PSBDSN4=, PSBDSN5= and PSBDSN6= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

UPSBTYPE

UPSBTYPE

Specifies whether or not the user can override the value specified in the PSBTYPE parameter.

Y The user can override the PSBTYPE parameter.

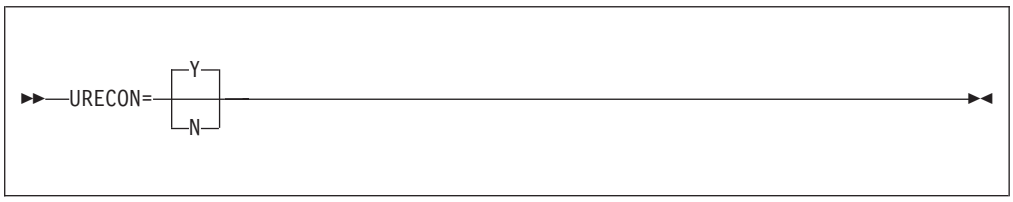
N The user cannot override the PSBTYPE parameter.

This parameter can only be specified on the FMN1POPD statement.

If you specify UPSBTYPE=N:
- PSBTYPE= statements in the FMIMSIN input of batch jobs are ignored.
- No PSBTYPE= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. The default is Y.

URECON



URECON

Specifies whether or not the user can specify RECON data sets.

Y The user can specify RECON data sets.

N The user cannot specify RECON data sets.

If you specify URECON=N:
- The RECON **Primary data set**, **Secondary**, and **Spare** data set name fields on the DLI Mode Data Sets 2 panel are protected.
- RECON1=, RECON2= and RECON3= statements in the FMIMSIN input of batch jobs are ignored.
- No RECON1=, RECON2= and RECON3= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

The URECON parameter was new in FM/IMS version 8.

URESLIB

URESLIB

Specifies whether or not the user can override the data sets specified in the RESLIBn parameters, when a function is run in DLI mode. (When a function is run in BMP mode, irrespective of the setting of this parameter, the user cannot override the data sets specified in the RESLIB*n* parameters.)

Y The user can override the RESLIBn data set parameters.

N The user cannot override the RESLIBn data set parameters.

If you specify URESLIB=N:
- The RESLIB Data set name fields on the DLI Mode Data Sets 1 panel are protected.
- RESLIB1=, RESLIB2=, RESLIB3=, RESLIB4=, RESLIB5= and RESLIB6= statements in the FMIMSIN input of batch jobs are ignored.
- No RESLIB1=, RESLIB2=, RESLIB3=, RESLIB4=, RESLIB5= and RESLIB6= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

The URESLIB parameter was new in FM/IMS version 8.

URSR

URSR Specifies whether or not the user can override the values specified in the GSGNAME and TMINAME parameters.

Y The user can override the GSGNAME and TMINAME parameters.

N The user cannot override the GSGNAME and TMINAME parameters.

If you specify URSR=N:
- The **GSGNAME** and **TMINAME** parameter fields on the DLI Mode Parameters panel are protected.
- GSGNAME= and TMINAME= statements in the FMIMSIN input of batch jobs are ignored.
- No GSGNAME= and TMINAME= statements are included in the JCL generated by FM/IMS dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

UTPLLIB

UTPLLIB

Specifies whether or not the user can override the data sets specified in the TPLLIBn parameters.

Y The user can override the TPLLIBn data set parameters.

N The user cannot override the TPLLIBn data set parameters.

If you specify UTPLLIB=N:
- The Template data set name fields on the Template Data Sets panel are protected.
- The TCINDD DD statements in the Batch Edit and Batch Browse JCL is ignored.

- No TCINDD DD statements are included in the JCL generated by the Batch Edit and Batch Browse dialogs.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement, (if specified on that statement), or Y (otherwise).

VCURULE

VCURULE
Specifies the view or criteria set usage rule.

T A view or criteria set can be used with a database when the template for the database is the same as the template that the view or criteria set was created from.

S A view or criteria set can be used with a database when the view or criteria set provides one or more layouts for each segment in the database.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or T (otherwise).

The user cannot override the value you specify for this parameter.

VSMPMEM

VSMPMEM
If the data set specified in the DFSVSAMP parameter is a PDS or PDSE, VSMPMEM specifies the member of that data set which contains the parameters defining the IMS buffer subpools for OSAM and VSAM data sets.

The specified member is allocated to the DFSVSAMP DD when FM/IMS functions are run in DLI mode.

This parameter is optional. If it is not specified on the FMN1POPI macro statement and the data set specified in the DFSVSAMP parameter is a PDS or a PDSE, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or DFSVSM00 (otherwise).

If you do not want the user to override the value you specify for this parameter, set UDFSVSMP=N.

XDOPTLB

XDOPTLB
Specifies whether or not the members of the DOPT ACBLIB data set are deleted and the DOPT ACBLIB data set compressed before FM/IMS executes an ACBGEN.

N FM/IMS deletes the PSB from the DOPT ACBLIB data set after BMP termination. FM/IMS does not delete any other members of the DOPT ACBLIB data set and does not compress the data set.

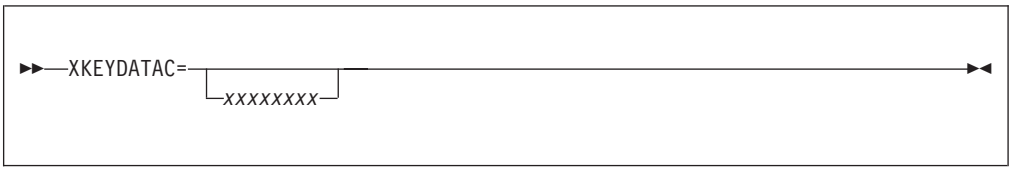
Y FM/IMS executes the ACBGEN with BUILD PSB=ALL specified on the SYSIN control statement. With this control statement specified, the ACBGEN deletes all PSBs and DBDs (and any other modules) from the DOPT ACBLIB data set and makes all their space available for reuse, before building blocks for the dynamic PSB.

This option may only be used when FM/IMS is the sole user of the DOPT ACBLIB data set.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or N (otherwise).

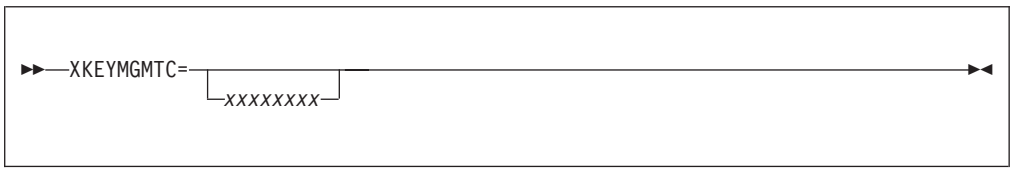
The user cannot override the value you specify for this parameter.

XKEYDATAC



XKEYDATAC
Specifies the SMS data class that FM/IMS uses when allocating the Extract keys file. The maximum length you can specify for XKEYDATAC is eight (8) bytes. XKEYDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied. This parameter can only be specified on the FMN1POPD macro statement.

XKEYMGMTC



XKEYMGMTC
Specifies the default SMS management class that FM/IMS uses when allocating the Extract keys file. The maximum length you can specify for XKEYMGMTC is eight (8) bytes. XKEYMGMTC is optional. If omitted the SMS management class specified in PMGMT is used, if supplied. This parameter can only be specified on the FMN1POPD macro statement.

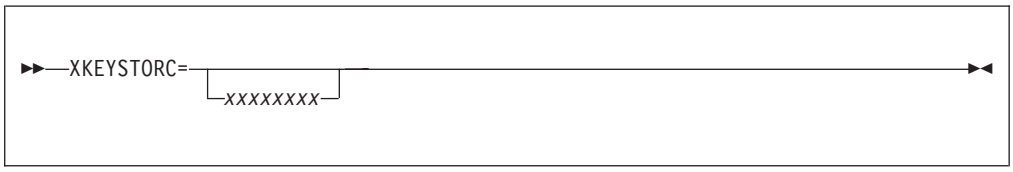
XKEYPQTY

XKEYPQTY
Specifies the amount of DASD space to be used for primary space allocation of the Extract keys file. The range depends on the space unit specified and the DASD device type. This parameter can only be specified on the FMN1POPD macro statement.

XKEYSQTY

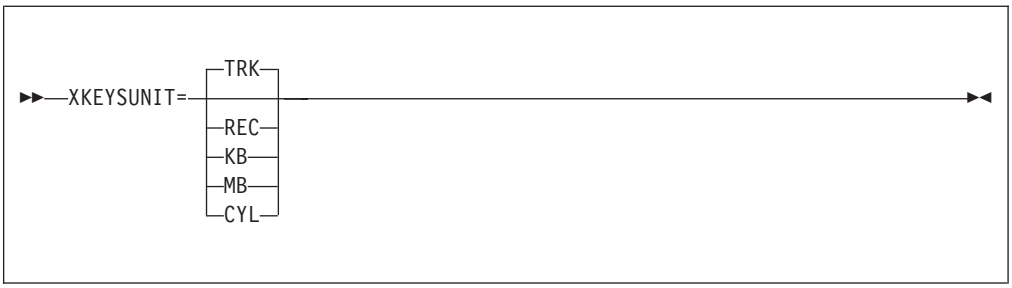
XKEYSQTY
Specifies the amount of DASD space to be used for secondary space allocation of the Extract keys file. The range depends on the space unit specified and the DASD device type. This parameter can only be specified on the FMN1POPD macro statement.

XKEYSTORC



XKEYSTORC
Specifies the default SMS storage class that FM/IMS uses when allocating the Extract keys file. The maximum length you can specify for XKEYSTORC is eight (8) bytes. XKEYSTORC is optional. If omitted the SMS storage class specified in PSTORC is used, if supplied. This parameter can only be specified on the FMN1POPD macro statement.

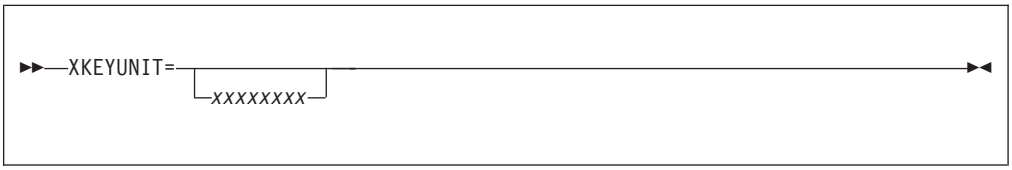
XKEYSUNIT



XKEYSUNIT
Specifies the unit of primary and secondary space to be allocated to the Extract keys file. This parameter can only be specified on the FMN1POPD macro statement.

REC Record of average size.
KB Kilobyte, a kilobyte is 1024 bytes.
MB Megabyte, a megabyte is 1048576 bytes.
TRK Track of a direct access storage device (DASD).
CYL Cylinder of a DASD.

XKEYUNIT



XKEYUNIT
Specifies the default permanent unit that FM/IMS uses when allocating the Extract keys file. The maximum length you can specify for XKEYUNIT is eight (8) bytes. XKEYUNIT is optional. If omitted the SMS storage class specified in PUNIT is used. This parameter can only be specified on the FMN1POPD macro statement.

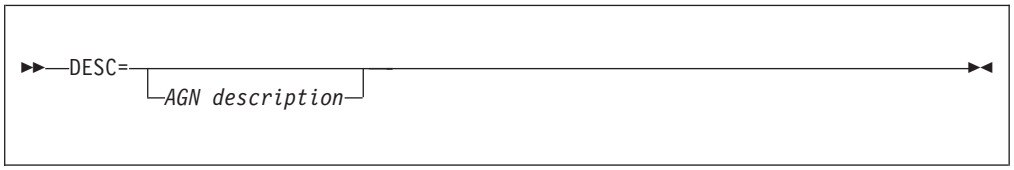
FMN1AGNT macro

AGN

AGN Specifies the 1- to 8-character AGN (Application Group Name) that you want FM/IMS to use when it starts a BMP region.

This parameter is required.

DESC



DESC Specifies a description of the AGN, to be displayed on the AGN Selection panel. The maximum length of the description is 45 characters. The description should be enclosed in quotation marks.

The parameter is optional. The default is no description.

SSID

SSID Specifies the 1- to 4-character identifier of the subsystem that the AGN is for. You can only specify FMN1AGNT macros for subsystems that you have provided FMN1POPI macro statement for.

This parameter is required.

Appendix D. File Manager options specified in PARMLIB members

This section describes the File Manager options specified in PARMLIB members. You can modify these File Manager options to suit your requirements.

File Manager options specified in FMN0PARM

This section describes the File Manager options specified in parmlib member FMN0PARM. You can modify these File Manager options to suit your requirements.

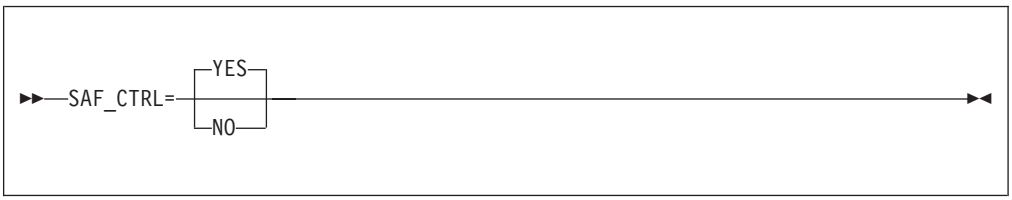
File Manager will not start unless the FMN0PARM PARMLIB member is present.

The FMN0PARM member provides a secure location for sensitive File Manager parameters used to control, for example, audit logging. File Manager uses system macros to locate this member in the current logical parmlib concatenation, making it difficult to over-ride or bypass the contents, which can occur when using the FMN0POPT module. When used to specify SAF- rule controlled audit logging, particularly to SMF, a high level of integrity over the auditing process is possible. The FMN0PARM member can also be used to force the use of a particular version of the FMN0POPT module, by specifying a specific data set from which the module should be loaded. This can be used to prevent a knowledgeable user creating a private version of the FMN0POPT module and loading it from a data set allocated to their TSO session, thereby bypassing installation-specified parameters.

This section includes the description of the options that can be specified in the FMN0PARM member, followed by a description of the facilities that can be used to customize the FMN0PARM member. Customization is entirely optional, but might be necessary, for example, in a sysplex environment where not all z/OS images have the same auditing requirements. It is also possible to specify different auditing options for different users using FMN0PARM statements; these changes can be made directly to the FMN0PARM member. When changes are made to the FMN0PARM member, no IPL is required and the changes become active the next time the user starts File Manager.

FMAUDIT

SAF_CTRL



SAF_CTRL
Specifies whether File Manager should use SAF to control all aspects of File Manager audit logging. See Chapter 1, “Preparing to customize File Manager,” on page 3 and Chapter 6, “Customizing the File Manager audit

© Copyright IBM Corp. 2000, 2011

facility (SAF-rule controlled auditing),” on page 51 for more information. SAF_CTRL is optional. The default is "YES", to use SAF.

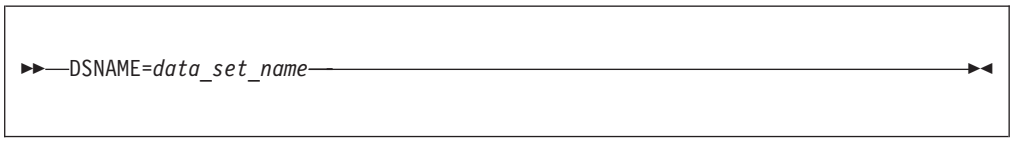
Note: When SAF-rule controlled auditing is in effect (SAF_CTRL=YES), File Manager ignores the values specified for the AUDITLOG and SMFNO parameters on the FMN0POPI macro statement.

SAF_SMFNO

SAF_SMFNO
Specifies the SMF record type for audit logging, where *number* is record type. You can specify a value from 128 to 255, or 0. The default is 0. If you specify an invalid SMF record number then auditing to SMF is inoperative.

FMOPTMOD

DSNAME



DSNAME
Specifies the name of the library from which File Manager is to load the File Manager options module - usually FMN0POPT. Note that the member keyword can be used to specify an alternative name to FMN0POPT, if required.

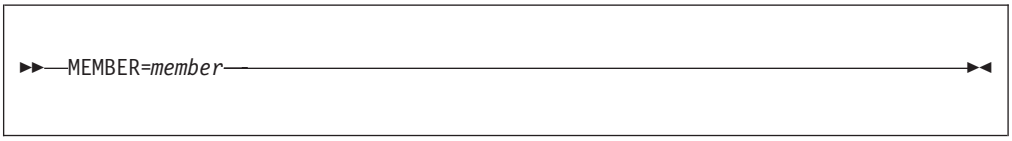
If the DSNAME keyword is specified, File Manager bypasses the usual library search when loading the File Manager options module and only searches the specified library for the module.

When the DSNAME keyword is omitted, File Manager searches load libraries allocated to the user's session in the normal order, for example, LIBDEF, ISPLLIB, STEPLIB, JOBLIB, link list.

One use of this option is in an environment with stringent security requirements, where it is necessary to ensure that all File Manager users access a particular File Manager options module, and that the use of this option module cannot be avoided.

The DSNAME keyword is optional.

MEMBER



MEMBER
Specifies the name of the member that contains the File Manager options module - usually FMN0POPT. The MEMBER keyword is optional.

Facilities for customizing the FMN0PARM definitions

You can use the FMN0PARM definitions to provide different options for File Manager users, based on the following environmental settings:

- The current zOS system ID
- The current version of File Manager
- The user's logonid

You can also split the FMN0PARM definitions across two or more members using the % (include) specification, and include comments using the * specification.

All of the facilities described in this section are optional. You do not need to use any of them.

Tags

A tag is an entry in the FMN0PARM member used to specify different options based on the current settings of certain environment variables. The following table shows the tags that can be specified.

Table 66. File Manager Parmlib member tag types

Tag Type	Priority	Description	Usage
Z	1	z/OS system	In a sysplex environment with multiple z/OS images.
V	2	FM Version	In a environment with multiple concurrent versions of File Manager.
U	3	User's logonid	Different options for specific users, typically by exception.

Rules for specifying tags

- Each tag should be entered on a separate line.
- Every initial tag must have a corresponding end tag. The initial tag is called the “start tag”.
- A start tag begins with a '<' and ends with a '>' character.
- The first character of the line containing a start tag must be '<', followed by the tag type.
- An end tag is specified as '<Ex>' or '</x>, where x is the tag type.
- The first character of the line containing an end tag must be '<', followed immediately by the rest of the end tag.
- One or more values can be specified for each tag. Separate values using a space or comma. Do not specify an '=' after the tag type.

- Values specified for each tag can include the * wildcard character. When specified, the '*' matches 0, 1, or many characters. Example: F* matches F, FRED, FUNNY, but not AFTER. The ** specification and other wildcard character(s) are not supported.
- Tags should be specified in order of priority to ensure correct processing.

Example of tag usage

Tags can be used to specify different FMN0PARM options based on certain environment variables (the z/OS system ID, the File Manager version and the user's TSO logonid). The most usual usage of tags is to provide an exception to the normal processing, for some particular set of circumstances.

Example 1:

Suppose SAF-rule controlled auditing is turned off for all users in a single z/OS-image. The FMN0PARM member would be coded as:

```
FMAUDIT SAF_CTRL=NO
```

Now suppose that there is a requirement to implement SAF-rule controlled auditing, and that one particular logonid (TEST1) has been selected for testing purposes. Assuming all the relevant SAF rules have been written, SAF-rule controlled auditing can be turned on for logonid TEST1 using:

```
FMAUDIT SAF_CTRL=NO     (1)
<U TEST1>          (2)
FMAUDIT SAF_CTRL=YES      (3)
<EU>             (4)
```

Explanation:

When File Manager parses the above, the line (1) is processed for all users, and turns SAF-rule controlled auditing off. When line (2) is processed, the user's TSO logonid is compared with the value TEST1. For all users other than TEST1, the test fails. When a tag fails to match a condition, all lines from the start tag to the matching end tag (inclusive) are ignored, so the net result is that, in the above example, only line (1) is processed. This ensures that the default of no SAF-rule controlled auditing applies to all users other than TEST1.

For user TEST1 however, the tag comparison on line (2) matches, so line (3) is included. File Manager re-processes the FMAUDIT statement for line (3), this time with SAF_CTRL=YES, resulting in SAF-rule controlled auditing being turned on for user TEST1.

Note: This example demonstrates two important principles when processing the statements within an FMN0PARM member:

- Multiple statements for the same option are allowed.
- If multiple statements for the same option are encountered, the last statement processed is the one that determines the setting.

Example 2:

Suppose in a sysplex environment there are nine z/OS images, with system IDs SYS1, SYS2, SYS3 ... SYS9. File Manager is available on all images, and SAF-rule controlled auditing is not required on SYS1, SYS2, ... SYS7 inclusive, but is required on SYS8 and SYS9. Further, on z/OS system SYS8, only TSO logonids that commence with DEV should be subject to SAF rule controlled audit; all other TSO

logonids are exempt. On system SYS9 however, all TSO logonids should be subject to SAF-rule controlled audit, with the single exception of TSO logonid MASTER1.

This could be coded as follows:

```
FMAUDIT SAF_CTRL=NO     (1)
<Z SYS8>            (2)
<U DEV*>            (3)
FMAUDIT SAF_CTRL=YES      (4)
</U>              (5)
</Z>              (6)
<Z SYS9>            (7)
FMAUDIT SAF_CTRL=YES      (8)
<U MASTER1>         (9)
FMAUDIT SAF_CTRL=NO     (10)
</U>              (11)
</Z>              (12)
```

Explanation:

Line (1) sets the default, which is not to use SAF- rule controlled auditing.

For all users on z/OS systems SYS1 ... SYS7 inclusive, the Z tags for SYS8 on line (2) and SYS9 on line (7) will not match the current environment, resulting in lines 2-12 inclusive being ignored.

For users running on z/OS system SYS8, line (2)-(6) inclusive are considered; lines (7)-(12) are ignored. Any TSO logonid that does not start with DEV (for example, PROD1) will not match the U tag (line 3), resulting in lines 3-5 inclusive being ignored. This leaves only line (1) to consider, which sets SAF-rule controlled auditing off. For a TSO logonid such as DEV76, the U tag on line (3) matches, so line (4) is included. File Manager processes the FMAUDIT statement on line (4) and sets SAF-rule controlled auditing on.

For users running on z/OS system SYS9, lines (2)-(6) inclusive are ignored; lines (7)-(12) inclusive are considered. Line (8) changes the default (for all users on system SYS9) to use SAF- rule controlled auditing. For all users other than TSO logonid MASTER1, lines (9)-(11) are ignored, resulting in the new default (line 8) being used. This turns SAF-rule controlled auditing on. For TSO logonid MASTER1 only, the U tag on line (9) matches and the FMAUDIT statement on line (10) is included. This turns SAF-rule controlled auditing off for user MASTER1.

Included members

You can divide the FMN0PARM across multiple members using the include specification. The format for an included member is as follows:

- The first character of the line must be a '%' character.
- Specify the member name to be included after the % character.
- Specify a single member name only after the % character.

Multiple levels of include nesting are supported, circular (recursive) definitions will result in an error and File Manager will not initialize.

Empty included members are ignored.

Example 1:

Suppose in a sysplex environment there are 4 z/OS images, with system IDs SYS1, SYS2, SYS3, and SYS4. File Manager is available on all images, with different auditing requirements on each image.

Include members can be used to code this as follows:

Member FMN0PARM

```
<Z SYS1>
%FMNPSYS1
</Z>
<Z SYS2>
%FMNPSYS2
</Z>
<Z SYS3>
%FMNPSYS3
</Z>
<Z SYS4>
%FMNPSYS4
</Z>
```

Member FMNPSYS1

```
FMAUDIT SAF_CTRL=YES
```

Member FMNPSYS2

```
FMAUDIT SAF_CTRL=NO
```

Member FMNPSYS3

```
FMAUDIT SAF_CTRL=NO
```

Member FMNPSYS4

```
FMAUDIT SAF_CTRL=YES
```

This has the advantage of consolidating all the options for each z/OS image in a specific member.

Care should be used when selecting member names for included members, to avoid any conflicts with other member names that might exist in other libraries in the logical parmlib concatenation. This applies only when FMN0POPT controlled auditing (see “Defining the FMN0PARM member” on page 4 for more information) is used to add the FMN0PARM member to the logical parmlib concatenation.

Comments

You can add comments to the FMN0PARM member as follows:

- The first character of a comment line must be a '*' character.
- The rest of the comment line is ignored.

Continuing specifications across multiple lines

You can specify the File Manager options across multiple lines as follows:

- The last character of each line to be continued must be a comma.
- Continue the text on the next line starting at any position (1-71).

Example:

```
FMAUDIT SAF_CTRL=YES
FMOPTMOD  DSNAME=FILEMAN.OTHER.OPTIONS,
    MEMBER=FMN0POPX
```

FM/IMS options specified in FMN1PARM

This section describes the FM/IMS options specified in parmlib member FMN1PARM. You can modify these FM/IMS options to suit your requirements.

FM/IMS will not start unless the FMN1PARM PARMLIB member is present.

The FMN1PARM member provides a secure location for sensitive FM/IMS parameters used to control, for example, audit logging. FM/IMS uses system macros to locate this member in the current logical parmlib concatenation, making it difficult to over-ride or bypass the contents, which can occur when using the FMN1POPT module. When used to specify SAF-rule controlled audit logging, particularly to SMF, a high level of integrity over the auditing process is possible. The FMN1PARM member can also be used to force the use of a particular version of the FMN1POPT module, by specifying a specific data set from which the module should be loaded. This can be used to prevent a knowledgeable user creating a private version of the FMN1POPT module and loading it from a data set allocated to their TSO session, thereby bypassing installation-specified parameters.

This section includes the description of the options that can be specified in the FMN1PARM member, followed by a description of the facilities that can be used to customize the FMN1PARM member. Customization is entirely optional, but might be necessary, for example, in a sysplex environment where not all z/OS images have the same auditing requirements. It is also possible to specify different auditing options for different users using FMN1PARM statements; these changes can be made directly to the FMN1PARM member. When changes are made to the FMN1PARM member, no IPL is required and the changes become active the next time the user starts FM/IMS.

FMAUDIT

SAF_CTRL

SAF_CTRL
Specifies whether FM/IMS should use SAF to control all aspects of FM/IMS audit logging. See Chapter 19, “Preparing to customize FM/IMS,” on page 163 and Chapter 24, “Customizing the FM/IMS audit facility (SAF-rule controlled auditing),” on page 205 for more information. SAF_CTRL is optional. The default is "YES", to use SAF.

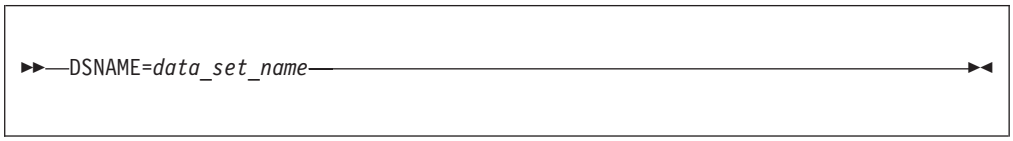
Note: When SAF-rule controlled auditing is in effect (SAF_CTRL=YES), FM/IMS ignores both the values specified for the IMSAUDLG parameter on the FMN1POPD and FMN1POPI macro statements and the value specified for the SMFNO parameter on the FMN0POPI macro statement.

SAF_SMFNO

SAF_SMFNO
Specifies the SMF record type for audit logging, where *number* is record type. You can specify a value from 128 to 255, or 0. The default is 0. If you specify an invalid SMF record number then auditing to SMF is inoperative.

FMOPTMOD

DSNAME



DSNAME
Specifies the name of the library from which FM/IMS is to load the FM/IMS options module - usually FMN1POPT. Note that the member keyword can be used to specify an alternative name to FMN1POPT, if required.

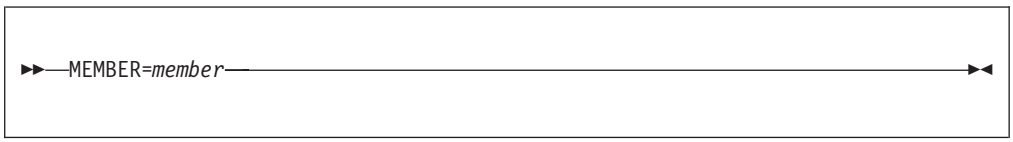
If the DSNAME keyword is specified, FM/IMS bypasses the usual library search when loading the FM/IMS options module and only searches the specified library for the module.

When the DSNAME keyword is omitted, FM/IMS searches load libraries allocated to the user's session in the normal order, for example, LIBDEF, ISPLLIB, STEPLIB, JOBLIB, link list.

One use of this option is in an environment with stringent security requirements, where it is necessary to ensure that all FM/IMS users access a particular FM/IMS options module, and that the use of this option module cannot be avoided.

The DSNAME keyword is optional.

MEMBER



MEMBER
Specifies the name of the member that contains the FM/IMS options module - usually FMN1POPT. The MEMBER keyword is optional.

Facilities for customizing the FMN1PARM definitions

You can use the FMN1PARM definitions to provide different options for FM/IMS users, based on the following environmental settings:

- The current zOS system ID
- The current version of FM/IMS
- The user's logonid

You can also split the FMN1PARM definitions across two or more members using the % (include) specification, and include comments using the * specification.

All of the facilities described in this section are optional. You do not need to use any of them.

Tags

A tag is an entry in the FMN1PARM member used to specify different options based on the current settings of certain environment variables. The following table shows the tags that can be specified.

Table 67. File Manager Parmlib member tag types

Tag Type	Priority	Description	Usage
Z	1	z/OS system	In a sysplex environment with multiple z/OS images.
V	2	FM Version	In a environment with multiple concurrent versions of File Manager.
U	3	User's logonid	Different options for specific users, typically by exception.

Rules for specifying tags

- Each tag should be entered on a separate line.
- Every initial tag must have a corresponding end tag. The initial tag is called the “start tag”.
- A start tag begins with a '<' and ends with a '>' character.
- The first character of the line containing a start tag must be '<', followed by the tag type.
- An end tag is specified as '<Ex>' or '</x>, where x is the tag type.
- The first character of the line containing an end tag must be '<', followed immediately by the rest of the end tag.
- One or more values can be specified for each tag. Separate values using a space or comma. Do not specify an '=' after the tag type.
- Values specified for each tag can include the * wildcard character. When specified, the '*' matches 0, 1, or many characters. Example: F* matches F, FRED, FUNNY, but not AFTER. The ** specification and other wildcard character(s) are not supported.
- Tags should be specified in order of priority to ensure correct processing.

Example of tag usage

Tags can be used to specify different FMN1PARM options based on certain environment variables (the z/OS system ID, the FM/IMS version and the user's TSO logonid). The most usual usage of tags is to provide an exception to the normal processing, for some particular set of circumstances.

Example 1:

Suppose SAF-rule controlled auditing is turned off for all users in a single z/OS-image. The FMN1PARM member would be coded as:

```
FMAUDIT SAF_CTRL=NO
```

Now suppose that there is a requirement to implement SAF-rule controlled auditing, and that one particular logonid (TEST1) has been selected for testing purposes. Assuming all the relevant SAF rules have been written, SAF-rule controlled auditing can be turned on for logonid TEST1 using:

```
FMAUDIT SAF_CTRL=NO     (1)
<U TEST1>          (2)
FMAUDIT SAF_CTRL=YES      (3)
<EU>             (4)
```

Explanation:

When FM/IMS parses the above, the line (1) is processed for all users, and turns SAF-rule controlled auditing off. When line (2) is processed, the user's TSO logonid is compared with the value TEST1. For all users other than TEST1, the test fails. When a tag fails to match a condition, all lines from the start tag to the matching end tag (inclusive) are ignored, so the net result is that, in the above example, only line (1) is processed. This ensures that the default of no SAF-rule controlled auditing applies to all users other than TEST1.

For user TEST1 however, the tag comparison on line (2) matches, so line (3) is included. FM/IMS re-processes the FMAUDIT statement for line (3), this time with SAF_CTRL=YES, resulting in SAF-rule controlled auditing being turned on for user TEST1.

Note: This example demonstrates two important principles when processing the statements within an FMN1PARM member:

- Multiple statements for the same option are allowed.
- If multiple statements for the same option are encountered, the last statement processed is the one that determines the setting.

Example 2:

Suppose in a sysplex environment there are nine z/OS images, with system IDs SYS1, SYS2, SYS3 ... SYS9. File Manager is available on all images, and SAF-rule controlled auditing is not required on SYS1, SYS2, ... SYS7 inclusive, but is required on SYS8 and SYS9. Further, on z/OS system SYS8, only TSO logonids that commence with DEV should be subject to SAF rule controlled audit; all other TSO logonids are exempt. On system SYS9 however, all TSO logonids should be subject to SAF-rule controlled audit, with the single exception of TSO logonid MASTER1.

This could be coded as follows:

```
FMAUDIT SAF_CTRL=NO     (1)
<Z SYS8>            (2)
<U DEV*>            (3)
FMAUDIT SAF_CTRL=YES      (4)
</U>              (5)
</Z>              (6)
<Z SYS9>            (7)
FMAUDIT SAF_CTRL=YES      (8)
<U MASTER1>         (9)
FMAUDIT SAF_CTRL=NO     (10)
</U>              (11)
</Z>              (12)
```

Explanation:

Line (1) sets the default, which is not to use SAF- rule controlled auditing.

For all users on z/OS systems SYS1 ... SYS7 inclusive, the Z tags for SYS8 on line (2) and SYS9 on line (7) will not match the current environment, resulting in lines 2-12 inclusive being ignored.

For users running on z/OS system SYS8, line (2)-(6) inclusive are considered; lines (7)-(12) are ignored. Any TSO logonid that does not start with DEV (for example, PROD1) will not match the U tag (line 3), resulting in lines 3-5 inclusive being ignored. This leaves only line (1) to consider, which sets SAF-rule controlled auditing off. For a TSO logonid such as DEV76, the U tag on line (3) matches, so line (4) is included. FM/IMS processes the FMAUDIT statement on line (4) and sets SAF-rule controlled auditing on.

For users running on z/OS system SYS9, lines (2)-(6) inclusive are ignored; lines (7)-(12) inclusive are considered. Line (8) changes the default (for all users on system SYS9) to use SAF- rule controlled auditing. For all users other than TSO logonid MASTER1, lines (9)-(11) are ignored, resulting in the new default (line 8) being used. This turns SAF-rule controlled auditing on. For TSO logonid MASTER1 only, the U tag on line (9) matches and the FMAUDIT statement on line (10) is included. This turns SAF-rule controlled auditing off for user MASTER1.

Included members

You can divide the FMN1PARM across multiple members using the include specification. The format for an included member is as follows:

- The first character of the line must be a '%' character.
- Specify the member name to be included after the % character.
- Specify a single member name only after the % character.

Multiple levels of include nesting are supported, circular (recursive) definitions will result in an error and FM/IMS will not initialize.

Empty included members are ignored.

Example 1:

Suppose in a sysplex environment there are 4 z/OS images, with system IDs SYS1, SYS2, SYS3, and SYS4. File Manager is available on all images, with different auditing requirements on each image.

Include members can be used to code this as follows:

Member FMN1PARM

```
<Z SYS1>
%FMNPSYS1
</Z>
<Z SYS2>
%FMNPSYS2
</Z>
<Z SYS3>
%FMNPSYS3
</Z>
<Z SYS4>
%FMNPSYS4
</Z>
```

Member FMNPSYS1

```
FMAUDIT SAF_CTRL=YES
```

Member FMNPSYS2

```
FMAUDIT SAF_CTRL=NO
```

Member FMNPSYS3

```
FMAUDIT SAF_CTRL=NO
```

Member FMNPSYS4

```
FMAUDIT SAF_CTRL=YES
```

This has the advantage of consolidating all the options for each z/OS image in a specific member.

Care should be used when selecting member names for included members, to avoid any conflicts with other member names that might exist in other libraries in the logical parmlib concatenation. This applies only when FMN1POPT controlled auditing (see "Defining the FMN1PARM member" on page 169 for more information) is used to add the FMN1PARM member to the logical parmlib concatenation.

Comments

You can add comments to the FMN1PARM member as follows:
- The first character of a comment line must be a '*' character.
- The rest of the comment line is ignored.

Continuing specifications across multiple lines

You can specify the FM/IMS options across multiple lines as follows:
- The last character of each line to be continued must be a comma.
- Continue the text on the next line starting at any position (1-71).

Example:

```
FMAUDIT SAF_CTRL=YES
FMOPTMOD  DSNAME=FILEMAN.OTHER.OPTIONS,
    MEMBER=FMN1POPX
```

FM/DB2 options specified in FMN2PARM

This section describes the FM/DB2 options specified in parmlib member FMN2PARM. You can modify these FM/DB2 options to suit your requirements.

FM/DB2 will not start unless the FMN2PARM PARMLIB is present.

The FMN2PARM member provides a secure location for sensitive FM/DB2 parameters used to control, for example, audit logging. FM/DB2 uses system macros to locate this member in the current logical parmlib concatenation, making it difficult to over-ride or bypass the contents, which can occur when using the FMN2POPT module. When used to specify SAF- rule controlled audit logging, particularly to SMF, a high level of integrity over the auditing process is possible. The FMN2PARM member can also be used to force the use of a particular version of the FMN2POPT module, by specifying a specific data set from which the module should be loaded. This can be used to prevent a knowledgeable user

creating a private version of the FMN2POPT module and loading it from a data set allocated to their TSO session, thereby bypassing installation-specified parameters.

This section includes the description of the options that can be specified in the FMN2PARM member, followed by a description of the facilities that can be used to customize the FMN2PARM member. Customization is entirely optional, but might be necessary, for example, in a sysplex environment where not all z/OS images have the same auditing requirements. It is also possible to specify different auditing options for different users using FMN2PARM statements; these changes can be made directly to the FMN2PARM member. When changes are made to the FMN2PARM member, no IPL is required and the changes become active the next time the user starts FM/DB2.

FMAUDIT

SAF_CTRL

SAF_CTRL
Specifies whether FM/DB2 should use SAF to control all aspects of FM/DB2 audit logging. See Chapter 12, “Preparing to customize FM/DB2,” on page 99 and Chapter 16, “Customizing the FM/DB2 audit facility (SAF-rule controlled auditing),” on page 135 for more information. SAF_CTRL is optional. The default is "YES", to use SAF.

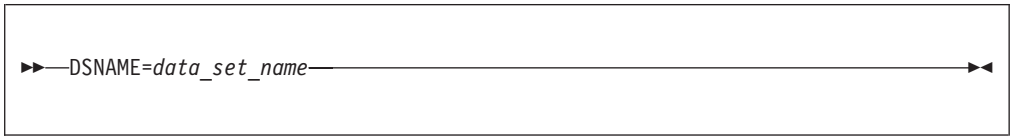
Note: When SAF-rule controlled auditing is in effect (SAF_CTRL=YES), FM/DB2 ignores both the values specified for the AUDIT parameter on the FMN2SSDM macro statements and the value specified for the SMFNO parameter on the FMN0POPI macro statement.

SAF_SMFNO

SAF_SMFNO
Specifies the SMF record type for audit logging, where *number* is record type. You can specify a value from 128 to 255, or 0. The default is 0. If you specify an invalid SMF record number then auditing to SMF is inoperative.

FMOPTMOD

DSNAME



DSNAME
Specifies the name of the library from which FM/DB2 is to load the FM/DB2 options module - usually FMN2POPT. Note that the member keyword can be used to specify an alternative name to FMN2POPT, if required.

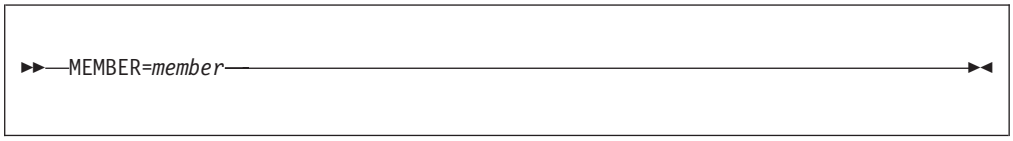
If the DSNAME keyword is specified, FM/DB2 bypasses the usual library search when loading the FM/DB2 options module and only searches the specified library for the module.

When the DSNAME keyword is omitted, FM/DB2 searches load libraries allocated to the user's session in the normal order, for example, LIBDEF, ISPLLIB, STEPLIB, JOBLIB, link list.

One use of this option is in an environment with stringent security requirements, where it is necessary to ensure that all FM/DB2 users access a particular FM/DB2 options module, and that the use of this option module cannot be avoided.

The DSNAME keyword is optional.

MEMBER



MEMBER
Specifies the name of the member that contains the FM/DB2 options module - usually FMN2POPT. The MEMBER keyword is optional.

Facilities for customizing the FMN2PARM definitions

You can use the FMN2PARM definitions to provide different options for FM/DB2 users, based on the following environmental settings:
- The current zOS system ID
- The current version of FM/DB2
- The user's logonid

You can also split the FMN2PARM definitions across two or more members using the % (include) specification, and include comments using the * specification.

All of the facilities described in this section are optional. You do not need to use any of them.

Tags

A tag is an entry in the FMN2PARM member used to specify different options based on the current settings of certain environment variables. The following table shows the tags that can be specified.

Table 68. File Manager Parmlib member tag types

Tag Type	Priority	Description	Usage
Z	1	z/OS system	In a sysplex environment with multiple z/OS images.
V	2	FM Version	In a environment with multiple concurrent versions of File Manager.
U	3	User's logonid	Different options for specific users, typically by exception.

Rules for specifying tags

- Each tag should be entered on a separate line.
- Every initial tag must have a corresponding end tag. The initial tag is called the “start tag”.
- A start tag begins with a '<' and ends with a '>' character.
- The first character of the line containing a start tag must be '<', followed by the tag type.
- An end tag is specified as '<Ex>' or '</x>, where x is the tag type.
- The first character of the line containing an end tag must be '<', followed immediately by the rest of the end tag.
- One or more values can be specified for each tag. Separate values using a space or comma. Do not specify an '=' after the tag type.
- Values specified for each tag can include the * wildcard character. When specified, the '*' matches 0, 1, or many characters. Example: F* matches F, FRED, FUNNY, but not AFTER. The ** specification and any other wildcard character(s) are not supported.
- Tags should be specified in order of priority to ensure correct processing.

Example of tag usage

Tags can be used to specify different FMN2PARM options based on certain environment variables (the z/OS system ID, the FM/DB2 version and the user's TSO logonid). The most usual usage of tags is to provide an exception to the normal processing, for some particular set of circumstances.

Example 1:

Suppose SAF-rule controlled auditing is turned off for all users in a single z/OS-image. The FMN2PARM member would be coded as:

```
FMAUDIT SAF_CTRL=NO
```

Now suppose that there is a requirement to implement SAF-rule controlled auditing, and that one particular logonid (TEST1) has been selected for testing purposes. Assuming all the relevant SAF rules have been written, SAF-rule controlled auditing can be turned on for logonid TEST1 using:

```
FMAUDIT SAF_CTRL=NO     (1)
<U TEST1>          (2)
FMAUDIT SAF_CTRL=YES      (3)
<EU>             (4)
```

Explanation:

When FM/DB2 parses the above, the line (1) is processed for all users, and turns SAF-rule controlled auditing off. When line (2) is processed, the user's TSO logonid is compared with the value TEST1. For all users other than TEST1, the test fails. When a tag fails to match a condition, all lines from the start tag to the matching end tag (inclusive) are ignored, so the net result is that, in the above example, only line (1) is processed. This ensures that the default of no SAF-rule controlled auditing applies to all users other than TEST1.

For user TEST1 however, the tag comparison on line (2) matches, so line (3) is included. FM/DB2 re-processes the FMAUDIT statement for line (3), this time with SAF_CTRL=YES, resulting in SAF-rule controlled auditing being turned on for user TEST1.

Note: This example demonstrates two important principles when processing the statements within an FMN2PARM member:

- Multiple statements for the same option are allowed.
- If multiple statements for the same option are encountered, the last statement processed is the one that determines the setting.

Example 2:

Suppose in a sysplex environment there are nine z/OS images, with system IDs SYS1, SYS2, SYS3 ... SYS9. File Manager is available on all images, and SAF-rule controlled auditing is not required on SYS1, SYS2, ... SYS7 inclusive, but is required on SYS8 and SYS9. Further, on z/OS system SYS8, only TSO logonids that commence with DEV should be subject to SAF rule controlled audit; all other TSO logonids are exempt. On system SYS9 however, all TSO logonids should be subject to SAF-rule controlled audit, with the single exception of TSO logonid MASTER1.

This could be coded as follows:

```
FMAUDIT SAF_CTRL=NO     (1)
<Z SYS8>            (2)
<U DEV*>            (3)
FMAUDIT SAF_CTRL=YES      (4)
</U>              (5)
</Z>              (6)
<Z SYS9>            (7)
FMAUDIT SAF_CTRL=YES      (8)
<U MASTER1>         (9)
FMAUDIT SAF_CTRL=NO     (10)
</U>              (11)
</Z>              (12)
```

Explanation:

Line (1) sets the default, which is not to use SAF- rule controlled auditing.

For all users on z/OS systems SYS1 ... SYS7 inclusive, the Z tags for SYS8 on line (2) and SYS9 on line (7) will not match the current environment, resulting in lines 2-12 inclusive being ignored.

For users running on z/OS system SYS8, line (2)-(6) inclusive are considered; lines (7)-(12) are ignored. Any TSO logonid that does not start with DEV (for example, PROD1) will not match the U tag (line 3), resulting in lines 3-5 inclusive being ignored. This leaves only line (1) to consider, which sets SAF-rule controlled

auditing off. For a TSO logonid such as DEV76, the U tag on line (3) matches, so line (4) is included. FM/DB2 processes the FMAUDIT statement on line (4) and sets SAF-rule controlled auditing on.

For users running on z/OS system SYS9, lines (2)-(6) inclusive are ignored; lines (7)-(12) inclusive are considered. Line (8) changes the default (for all users on system SYS9) to use SAF- rule controlled auditing. For all users other than TSO logonid MASTER1, lines (9)-(11) are ignored, resulting in the new default (line 8) being used. This turns SAF-rule controlled auditing on. For TSO logonid MASTER1 only, the U tag on line (9) matches and the FMAUDIT statement on line (10) is included. This turns SAF-rule controlled auditing off for user MASTER1.

Included members

You can divide the FMN2PARM across multiple members using the include specification. The format for an included member is as follows:

- The first character of the line must be a '%' character.
- Specify the member name to be included after the % character.
- Specify a single member name only after the % character.

Multiple levels of include nesting are supported, circular (recursive) definitions will result in an error and File Manager will not initialize.

Empty included members are ignored.

Example 1:

Suppose in a sysplex environment there are 4 z/OS images, with system IDs SYS1, SYS2, SYS3, and SYS4. File Manager is available on all images, with different auditing requirements on each image.

Include members can be used to code this as follows:

Member FMN2PARM

```
<Z SYS1>
%FMNPSYS1
</Z>
<Z SYS2>
%FMNPSYS2
</Z>
<Z SYS3>
%FMNPSYS3
</Z>
<Z SYS4>
%FMNPSYS4
</Z>
```

Member FMNPSYS1

```
FMAUDIT SAF_CTRL=YES
```

Member FMNPSYS2

```
FMAUDIT SAF_CTRL=NO
```

Member FMNPSYS3

```
FMAUDIT SAF_CTRL=NO
```

Member FMNPSYS4

```
FMAUDIT SAF_CTRL=YES
```

This has the advantage of consolidating all the options for each z/OS image in a specific member.

Care should be used when selecting member names for included members, to avoid any conflicts with other member names that might exist in other libraries in the logical parmlib concatenation. This applies only when FMN2POPT controlled auditing (see "Defining the FMN2PARM member" on page 100 for more information) is used to add the FMN2PARM member to the logical parmlib concatenation.

Comments

You can add comments to the FMN2PARM member as follows:
- The first character of a comment line must be a '*' character.
- The rest of the comment line is ignored.

Continuing specifications across multiple lines

You can specify the FM/DB2 options across multiple lines as follows:
- The last character of each line to be continued must be a comma.
- Continue the text on the next line starting at any position (1-71).

Example:

```
FMAUDIT SAF_CTRL=YES
FMOPTMOD  DSNAME=FILEMAN.OTHER.OPTIONS,
    MEMBER=FMN2POPX
```

FM/CICS options specified in FMN3PARM

This section describes the FM/CICS options specified in parmlib member FMN3PARM. You can modify these FM/CICS options to suit your requirements.

FM/CICS will not start unless the FMN3PARM PARMLIB member is present.

The FMN3PARM member provides a secure location for sensitive FM/CICS parameters used to control, for example, audit logging. FM/CICS uses system macros to locate this member in the current logical parmlib concatenation, making it difficult to over-ride or bypass the contents, which can occur when using the FMN3POPT module. When used to specify SAF- rule controlled audit logging, particularly to SMF, a high level of integrity over the auditing process is possible. The FMN3PARM member can also be used to force the use of a particular version of the FMN3POPT module, by specifying a specific data set from which the module should be loaded. This can be used to prevent a knowledgeable user creating a private version of the FMN3POPT module and loading it from a data set allocated to their TSO session, thereby bypassing installation-specified parameters.

This section includes the description of the options that can be specified in the FMN3PARM member, followed by a description of the facilities that can be used to customize the FMN3PARM member. Customization is entirely optional, but might be necessary, for example, in a sysplex environment where not all z/OS images have the same auditing requirements. It is also possible to specify different auditing options for different users using FMN3PARM statements; these changes can be made directly to the FMN3PARM member. When changes are made to the FMN3PARM member, no IPL is required and the changes become active the next time the user starts FM/CICS.

FMAUDIT

SAF_CTRL

SAF_CTRL

Specifies whether FM/CICS should use SAF to control all aspects of FM/CICS audit logging. See Chapter 27, “Preparing to customize FM/CICS,” on page 231 and Chapter 33, “Customizing the FM/CICS audit facility (SAF-rule controlled auditing),” on page 259 for more information. SAF_CTRL is optional. The default is "YES", to use SAF.

Note: When SAF-rule controlled auditing is in effect (SAF_CTRL=YES), FM/CICS ignores the values specified for the AUDITLOG and SMFNO parameters on the FMN0POPI macro statement.

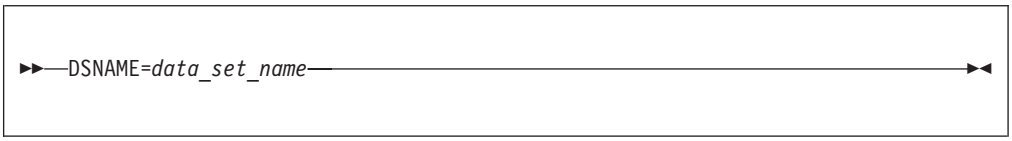
SAF_SMFNO

SAF_SMFNO

Specifies the SMF record type for audit logging, where *number* is record type. You can specify a value from 128 to 255, or 0. The default is 0. If you specify an invalid SMF record number then auditing to SMF is inoperative.

FMOPTMOD

DSNAME



DSNAME

Specifies the name of the library from which FM/CICS is to load the FM/CICS options module - usually FMN3POPT. Note that the member keyword can be used to specify an alternative name to FMN3POPT, if required.

If the DSNAME keyword is specified, FM/CICS bypasses the usual library search when loading the FM/CICS options module and only searches the specified library for the module.

When the DSNAME keyword is omitted, FM/CICS searches load libraries allocated to the user's session in the normal order, for example, LIBDEF, ISPLLIB, STEPLIB, JOBLIB, link list.

One use of this option is in an environment with stringent security requirements, where it is necessary to ensure that all FM/CICS users access a particular FM/CICS options module, and that the use of this option module cannot be avoided.

The DSNAME keyword is optional.

MEMBER

MEMBER
Specifies the name of the member that contains the FM/CICS options module - usually FMN3POPT. The MEMBER keyword is optional.

Facilities for customizing the FMN3PARM definitions

You can use the FMN3PARM definitions to provide different options for FM/CICS users, based on the following environmental settings:

- The current zOS system ID
- The current version of FM/CICS
- The user's logonid

You can also split the FMN3PARM definitions across two or more members using the % (include) specification, and include comments using the * specification.

All of the facilities described in this section are optional. You do not need to use any of them.

Tags

A tag is an entry in the FMN3PARM member used to specify different options based on the current settings of certain environment variables. The following table shows the tags that can be specified.

Table 69. File Manager Parmlib member tag types

Tag Type	Priority	Description	Usage
Z	1	z/OS system	In a sysplex environment with multiple z/OS images.
V	2	FM Version	In a environment with multiple concurrent versions of File Manager.
U	3	User's logonid	Different options for specific users, typically by exception.

Rules for specifying tags

- Each tag should be entered on a separate line.
- Every initial tag must have a corresponding end tag. The initial tag is called the “start tag”.

- A start tag begins with a '<' and ends with a '>' character.
- The first character of the line containing a start tag must be '<', followed by the tag type.
- An end tag is specified as '<Ex>' or '</x>, where x is the tag type.
- The first character of the line containing an end tag must be '<', followed immediately by the rest of the end tag.
- One or more values can be specified for each tag. Separate values using a space or comma. Do not specify an '=' after the tag type.
- Values specified for each tag can include the * wildcard character. When specified, the '*' matches 0, 1, or many characters. Example: F* matches F, FRED, FUNNY, but not AFTER. The ** specification and any other wildcard character(s) are not supported.
- Tags should be specified in order of priority to ensure correct processing.

Example of tag usage

Tags can be used to specify different FMN3PARM options based on certain environment variables (the z/OS system ID, the FM/CICS version and the user's TSO logonid). The most usual usage of tags is to provide an exception to the normal processing, for some particular set of circumstances.

Example 1:

Suppose SAF-rule controlled auditing is turned off for all users in a single z/OS-image. The FMN3PARM member would be coded as:

```
FMAUDIT SAF_CTRL=NO
```

Now suppose that there is a requirement to implement SAF-rule controlled auditing, and that one particular logonid (TEST1) has been selected for testing purposes. Assuming all the relevant SAF rules have been written, SAF-rule controlled auditing can be turned on for logonid TEST1 using:

```
FMAUDIT SAF_CTRL=NO     (1)
<U TEST1>          (2)
FMAUDIT SAF_CTRL=YES      (3)
<EU>             (4)
```

Explanation:

When FM/CICS parses the above, the line (1) is processed for all users, and turns SAF-rule controlled auditing off. When line (2) is processed, the user's TSO logonid is compared with the value TEST1. For all users other than TEST1, the test fails. When a tag fails to match a condition, all lines from the start tag to the matching end tag (inclusive) are ignored, so the net result is that, in the above example, only line (1) is processed. This ensures that the default of no SAF-rule controlled auditing applies to all users other than TEST1.

For user TEST1 however, the tag comparison on line (2) matches, so line (3) is included. FM/CICS re-processes the FMAUDIT statement for line (3), this time with SAF_CTRL=YES, resulting in SAF-rule controlled auditing being turned on for user TEST1.

Note: This example demonstrates two important principles when processing the statements within an FMN3PARM member:

- Multiple statements for the same option are allowed.
- If multiple statements for the same option are encountered, the last statement processed is the one that determines the setting.

Example 2:

Suppose in a sysplex environment there are nine z/OS images, with system IDs SYS1, SYS2, SYS3 ... SYS9. File Manager is available on all images, and SAF-rule controlled auditing is not required on SYS1, SYS2, ... SYS7 inclusive, but is required on SYS8 and SYS9. Further, on z/OS system SYS8, only TSO logonids that commence with DEV should be subject to SAF rule controlled audit; all other TSO logonids are exempt. On system SYS9 however, all TSO logonids should be subject to SAF-rule controlled audit, with the single exception of TSO logonid MASTER1.

This could be coded as follows:

```
FMAUDIT SAF_CTRL=NO     (1)
<Z SYS8>            (2)
<U DEV*>            (3)
FMAUDIT SAF_CTRL=YES      (4)
</U>              (5)
</Z>              (6)
<Z SYS9>            (7)
FMAUDIT SAF_CTRL=YES      (8)
<U MASTER1>         (9)
FMAUDIT SAF_CTRL=NO     (10)
</U>              (11)
</Z>              (12)
```

Explanation:

Line (1) sets the default, which is not to use SAF- rule controlled auditing.

For all users on z/OS systems SYS1 ... SYS7 inclusive, the Z tags for SYS8 on line (2) and SYS9 on line (7) will not match the current environment, resulting in lines 2-12 inclusive being ignored.

For users running on z/OS system SYS8, line (2)-(6) inclusive are considered; lines (7)-(12) are ignored. Any TSO logonid that does not start with DEV (for example, PROD1) will not match the U tag (line 3), resulting in lines 3-5 inclusive being ignored. This leaves only line (1) to consider, which sets SAF-rule controlled auditing off. For a TSO logonid such as DEV76, the U tag on line (3) matches, so line (4) is included. FM/CICS processes the FMAUDIT statement on line (4) and sets SAF-rule controlled auditing on.

For users running on z/OS system SYS9, lines (2)-(6) inclusive are ignored; lines (7)-(12) inclusive are considered. Line (8) changes the default (for all users on system SYS9) to use SAF- rule controlled auditing. For all users other than TSO logonid MASTER1, lines (9)-(11) are ignored, resulting in the new default (line 8) being used. This turns SAF-rule controlled auditing on. For TSO logonid MASTER1 only, the U tag on line (9) matches and the FMAUDIT statement on line (10) is included. This turns SAF-rule controlled auditing off for user MASTER1.

Included members

You can divide the FMN3PARM across multiple members using the include specification. The format for an included member is as follows:

- The first character of the line must be a '%' character.
- Specify the member name to be included after the % character.
- Specify a single member name only after the % character.

Multiple levels of include nesting are supported, circular (recursive) definitions will result in an error and FM/CICS will not initialize.

Empty included members are ignored.

Example 1:

Suppose in a sysplex environment there are 4 z/OS images, with system IDs SYS1, SYS2, SYS3, and SYS4. File Manager is available on all images, with different auditing requirements on each image.

Include members can be used to code this as follows:

Member FMN3PARM

```
<Z SYS1>
%FMNPSYS1
</Z>
<Z SYS2>
%FMNPSYS2
</Z>
<Z SYS3>
%FMNPSYS3
</Z>
<Z SYS4>
%FMNPSYS4
</Z>
```

Member FMNPSYS1

```
FMAUDIT SAF_CTRL=YES
```

Member FMNPSYS2

```
FMAUDIT SAF_CTRL=NO
```

Member FMNPSYS3

```
FMAUDIT SAF_CTRL=NO
```

Member FMNPSYS4

```
FMAUDIT SAF_CTRL=YES
```

This has the advantage of consolidating all the options for each z/OS image in a specific member.

Care should be used when selecting member names for included members, to avoid any conflicts with other member names that might exist in other libraries in the logical parmlib concatenation. This applies only when FMN3POPT controlled auditing (see "Defining the FMN3PARM member" on page 232 for more information) is used to add the FMN3PARM member to the logical parmlib concatenation.

Comments

You can add comments to the FMN3PARM member as follows:

- The first character of a comment line must be a '*' character.
- The rest of the comment line is ignored.

Continuing specifications across multiple lines

You can specify the FM/CICS options across multiple lines as follows:

- The last character of each line to be continued must be a comma.
- Continue the text on the next line starting at any position (1-71).

Example:

```
FMAUDIT SAF_CTRL=YES
FMOPTMOD  DSNAME=FILEMAN.OTHER.OPTIONS,
    MEMBER=FMN3POPX
```

Appendix E. Customizing the FM Server

This section describes how to customize the FM Server. You do this after you have installed File Manager and if you plan to use either:

- FM/CICS without using a batch initiator
- File Manager Eclipse plugin (in Eclipse, or CICS Explorer or RDz)

This server runs a z/OS UNIX process that will listen for a connection request on a specific port. The server can be started manually or during IPL by execution of a customized procedure. For usage by the FM/CICS interface, (i.e. START=TASK has been specified in FMN3POPT), when a user runs the FM transaction, a connection request will be received by the server from the CICS client. The server will then verify the credentials of the client (userid and password or passticket) and if valid, create a new FM process for that user. The userid of the newly created process will be changed to that of the client and all further communication will occur between the CICS client and the newly created FM process via TCP/IP. The server will then wait for another connection request from a new client. The above process is also true for RDz.

FM Server options

The default name of the FM Server is FMNCAS. You can change this name to suit your site's requirements.

You must provide and configure the server options in a data set specified on the FMNOPTS DD statement in the server procedure (FMNCAS). The data set defined to this DD statement must be sequential, with fixed 80 byte length records. The SERVER option statement contained in this data set is described below.

Syntax of the SERVER statement

```
SERVER PORT=port_num,MAXCONN=max_conn,PATH=/path_name/FMNSPAWN
```

PORT specifies the port number dedicated to the FM Server. The FM Server will listen for client connection requests on the specified port number.

You should reserve Port number to the started task by updating the reserved port list in the TCPIP profile to ensure the server task owns the specified port.

MAXCONN
specifies the maximum number of client processes, created by the FM Server, that can be concurrently active. Once the number of currently active client processes created by the FM Server reaches the specified *max_conn* value, no additional client connection requests will be accepted by the FM Server until one or more currently active client processes terminate.

MAXCONN is an optional parameter, and if specified, must be a number in the range of 1-99999999. If MAXCONN is not specified, or 0 is specified, there is no FM-enforced limit of the number of active client processes.

PATH Specifies the location of a one-time writeable area for the server to create an external link file (FMNSPAWN) and a SSL certificate file (fmnserv). This location is specified including the file name FMNSPAWN. The path name

© Copyright IBM Corp. 2000, 2011

can be any name. However, the path must exist and be accessible from the SERVER task and have write access for the first time the server is launched.

The path is used to invoked the File Manager load module FMNSPAWN which must be available in the active steplib or link list for the server task during the spawn process. The path location is also used to hold an SSL certificate file for SSL-encrypted communication with the File Manager Eclipse plugin. For example the SERVER statement might be specified as:

```
SERVER PORT=2071,PATH=/etc/FMNSPAWN
```

Modifying FMNCAS

Once you have created the FMNOPTS data set, you need to modify FMNCAS to suit your site's requirements. You do this as follows:

1. Copy the sample member FMNCAS from FMN.SFMNSAM1 into your own procedure library.
2. Modify this sample in your library, changing the name if required.
   - Change *#fmnhlq* to the high level qualifier for the File Manager target libraries.
   - Change *#ipvhlq* to the high level qualifier for the IBM Problem Determination Tools Common Components target libraries. IPV is required for the operation of FM/CICS. Information about installing IBM Problem Determination Tools Common Components is included in the *File Manager Program Directory*. If you do not plan to use the FM Server to access FM/CICS, comment out or remove the DD statements with the high level qualifier *#ipvhlq*. These statements are in the FMNLIB, IPVPLIB, and IPVTLIB concatenations in FMNCAS.
   - Change #fmnopts to the name of your FMNOPTS data set, containing the required FM Server parameters.
   - If the name of the TCP/IP address space is not the default value of 'TCPIP', uncomment the SYSTCPD DD statement and change *#tcpparms* to the data set and member name that contains the appropriate TCPIPJOBNAME parameter.
   - If you plan to use FM/CICS, using the FM Server, or the FM Eclipse Plugin, to access WebSphere MQ queues, uncomment the three MQ target library statements in the STEPLIB concatenation and change *#mqhlq* to the high-level qualifier of your MQ libraries. The required MQ libraries are: SCSQAUTH, SCSQANLE, and SCSQLOAD.
   - If you plan to access FM/DB2 from FM/CICS, using the FM Server, and use the DB2 DSN command processor, uncomment the DB2 load library statement in the STEPLIB concatenation and change *#db2hlq* to the high level qualifier of your DB2 load library. The DB2 load library is: SDSNLOAD.

     If you are using more than one version of DB2, then specify the load library of the earliest version of DB2 you are using.

Note: If you plan to use the same FM Server to access FM/CICS and the File Manager Plugin for Eclipse, then the PORT number specified in FMNOPTS must be the same as that specified in FMN3POPT.

Additional security considerations

The STEPLIB FMN.SFMNMODA must be APF-authorized, as the spawned address space needs to attach TSO to support authorized services such as SMF logging, IEBCOPY invocation and DB2 CLIST processing.

The use of RACF to control UNIX functions may require additional consideration and configuration for the FMNCAS server program to run correctly.

The FM Server performs functions similar to a UNIX daemon, and requires daemon authority or a userid with superuser authority, for example, UID 0. FMNCAS issues the INITACEE service to authenticate a client's userid and password before spawning the new address space with the *_bpx_userid=userid* environment variable to change the authority of the new address space.

Therefore, if enhanced program security is enabled or the BPX.DAEMON FACILITY is defined, at a minimum the following programs must be defined to program control, unless unless BPX.DAEMON.HFSCTL has been set up:

- FMNCAS
- FMNMODAC
- FMNMODTR
- FMNMOD16
- FMNCERT
- FMNCSSL1
- FMNCSSL2
- FMNCSSL3
- FMNCSSL4
- FMNCERT

Alternatively, you could define the library, FMN.SFMNMODA, to program control, rather than specify individual programs.

If enhanced program security is enabled, FMNCAS must be defined with the MAIN attribute via the APPLDATA operand on the PROGRAM profile.

Notes:

1. If you are using Japanese, then include the module FMN0MJPN in program control.
2. If you are using Korean, then include the module FMN0MKOR in program control.

Refer to the chapter, "Setting up for daemons", in *UNIX System Services Planning* for more information.

If RACF, or an equivalent security product is implemented, the FM Server (FMNCAS) started task must also be defined to the STARTED class. For example, to add FMNCAS as an STC, the following RACF commands could be used, where "FMNCAS" is the name of your FM Server procedure and *userid* is the userid that the started task will run under:

```
RDEFINE STARTED fmncas.* STDATA(USER(userid))
SETROPTS RACLIST(STARTED) REFRESH
```

Note: The userid that the FMNCAS procedure runs under must also have an OMVS RACF segment defined.

Refer to *z/OS Security Server RACF Security Administrator's Guide*, or equivalent documentation for your security product, for more information about started tasks and security.

Address space timeout

When an address space is launched for a client, and has completed its current function, the address space will be waiting for TCP/IP communications from the File Manager Plugin for Eclipse or FM/CICS client. Accordingly, the client address space may be subject to an s522 abend if waiting longer than the active site settings for job wait time. The job wait time is controlled by the **JWT** parameter of the **SMFPRM***xx* member, but may also be set to never time out by the site settings for **MAXCPUTIME** in the site's **BPXPRM***xx* member. These settings should be considered and set as desired by the site.

Appendix F. The library management system exit

This section describes the code and functions of the sample COBOL exit, FMNCRAEX, and also provides information on supporting multiple LMSs, and considerations when writing the exit in High Level Assembler.

Coding the exit: the basics

The main program and first argument: the operation code

The arguments to the exit follow standard 390 level two coding conventions: a pointer is provided to each argument.

The sample COBOL exit takes advantage of the fact that COBOL expects each argument to be a pointer. COBOL internally sets the address of the associated 01 structure in the LINKAGE SECTION.

The first argument in every call to the exit is a pointer to an operation code:

```
01 RAM-OPCODE PIC S9(9) BINARY.
01 RAM-OPCODE-VAL REDEFINES RAM-OPCODE PIC X(4).
88 OPCODE-INIT-RAM           VALUE X"09010000".
88 OPCODE-TERM-RAM           VALUE X"09010001".
88 OPCODE-VALIDATE-REPOS     VALUE X"09010004".
88 OPCODE-GET-1ST-MEM-REC    VALUE X"09010014".
88 OPCODE-GET-NEXT-MEM-REC   VALUE X"09010015".
88 OPCODE-GET-1ST-MEM-INFO   VALUE X"09010010".
88 OPCODE-GET-NEXT-MEM-INFO  VALUE X"09010011".
88 OPCODE-GET-DISPLAY-INFO   VALUE X"09010008".
```

The calls defined as level 88s in the COBOL code are the services which FMNCRAEX must provide.

Here is the main program showing how the OpCode is received by the FMNCRAEX program:

```
PROCEDURE DIVISION USING RAM-OPCODE,
                         RAM-WORK-AREA-PTR,
                         RAM-PARM2,
                         RAM-PARM3,
                         RAM-PARM4,
                         RAM-PARM5,
                         RAM-PARM6.
MAIN SECTION.

STARTUP-RAM.
    IF TRACELVL >= TRACEBAS THEN PERFORM TRACE-STARTUP.
    MOVE 0 TO RAM-RC.

SELECT-RAM-FUNCTION.
    EVALUATE TRUE
       WHEN OPCODE-INIT-RAM
          PERFORM INIT-RAM
       WHEN OPCODE-TERM-RAM
          PERFORM TERM-RAM
       WHEN OPCODE-VALIDATE-REPOS
          PERFORM VALIDATE-REPOS
       WHEN OPCODE-GET-1ST-MEM-REC
          PERFORM GET-1ST-MEM-REC
       WHEN OPCODE-GET-NEXT-MEM-REC
```

© Copyright IBM Corp. 2000, 2011

```
            PERFORM GET-NEXT-MEM-REC
         WHEN OPCODE-GET-1ST-MEM-INFO
            PERFORM GET-1ST-MEM-INFO
         WHEN OPCODE-GET-NEXT-MEM-INFO
            PERFORM GET-NEXT-MEM-INFO
         WHEN OPCODE-GET-DISPLAY-INFO
            PERFORM GET-DISPLAY-INFO
         WHEN OTHER
            SET RAM-RC-UNKNOWN-OPCODE TO TRUE
      END-EVALUATE.

RETURN-FROM-RAM.
      MOVE RAM-RC TO RETURN-CODE.
      GOBACK.
```

The return code

The return code from the sample COBOL exit is first set into a WORKING STORAGE variable RAM-RC, and at the end of the exit is placed into the special register RETURN-CODE.

Here are the possible return codes from the exit:

```
WORKING-STORAGE SECTION.

77 RAM-RC PIC S9(9) COMP.
   88 RAM-RC-OK VALUE 0.
   88 RAM-RC-UNKNOWN-OPCODE          VALUE 999.
   88 RAM-RC-REP-NOT-FOUND           VALUE 116.
   88 RAM-RC-INTERNAL-ERROR          VALUE 100.
   88 RAM-RC-REP-NOT-SUPPORTED       VALUE 120.
   88 RAM-RC-REP-IN-USE              VALUE 124.
   88 RAM-RC-MEM-ID-SPEC-BAD         VALUE 156.
   88 RAM-RC-REP-UNAVAIL-UNKNOWN     VALUE 132.
   88 RAM-RC-EOF                     VALUE 180.
   88 RAM-RC-REP-ERROR               VALUE 136.
   88 RAM-RC-BAD-FILTER              VALUE 152.
```

The return codes have the following meanings:

RAM-RC-UNKNOWN-OPCODE
Indicates that the requested function is not supported by the exit.

RAM-RC-REP-NOT-FOUND
Indicates that the library was not found. If the library name was a data set name, this means the data set was not found.

RAM-RC-INTERNAL-ERROR
Indicates some internal error occurred in the exit itself.

RAM-RC-REP-NOT-SUPPORTED
The library is not supported by this exit. File Manager should try to access the library using other access methods such as CA-Panvalet or PDS access.

RAM-RC-REP-IN-USE
The library is in exclusive use.

RAM-RC-MEM-ID-SPEC-BAD
The Member Identifier Specification (the member name) parameter was not usable.

RAM-RC-REP-UNAVAIL-UNKNOWN
The library is unavailable for unknown reasons.

RAM-RC-EOF
Normal End Of File was received during the process of extracting member records or getting member information or metadata (for a member selection list).

RAM-RC-REP-ERROR
Some error occurred in the library.

RAM-RC-BAD-FILTER
The "filter" argument to the GetMemberInfo calls, which is a member name with optional wildcard characters, was bad.

Initializing the exit

Initializing the exit is trivial for a High Level Language (HLL) like COBOL or PL/I. Language Environment (LE) obtains storage for HLLs. In the case of the sample exit, before COBOL starts, LE obtains the required COBOL working storage. Whenever the COBOL program is called, LE ensures it has access to its working storage. LE also maintains the data in working storage across calls.

If the user exit is written in HLASM, however, initializing the exit is more complex. You can use the INIT-RAM and TERM-RAM calls to get and release storage needed by the exit. The exit can set the second argument in every call to the exit, the RAM-WORK-AREA-PTR, at any time. File Manager preserves and passes this value with every call to the exit. This provides a way for the exit to gain addressability to storage on every call. The exit should release all storage when it receives a call to TERM-RAM.

The sample COBOL exit code does not use the RAM-WORK-AREA-PTR, because, as mentioned above, LE provides an alternative to this mechanism for HLLs.

The code in INIT-RAM in the sample exit simply sets the tracing level from the caller and sets up the callback logging function address.

The arguments are:

```
Parm0 = OpCode = InitializeRAM
Parm1 = RAM-WORK-AREA-PTR
Parm2 = (Reserved)
Parm3 = TRACELVL  (integer value from 0 to 3)
Parm4 = Trace callback function address
```

The sample exit has two statements, **PERFORM CLOSE-AND-FREE-DIRFILE** and **PERFORM CLOSE-AND-FREE-MEMFILE**, that you may need to change to free your own resources. This is done to ensure that even if a previous instance of the exit abended prior to freeing resources, the resources will be freed.

Tracing

You can trace your exit while it is running under File Manager. To do this, the exit writes to a log file. To write to a log file, define a DD with the name CRALOG. It should be a sequential file with RECFM FB and LRECL 132.

FMNCRAEX uses a logging callback function so that different logging functions can be provided depending on the caller of FMNCRAEX. In the sample TEST program, the log output is simply DISPLAYed. When File Manager calls FMNCRAEX, as described above, it writes the log information to the predefined DD.

The callback mechanism means that logging calls should be written in the following style:

```
IF TRACELVL >= TRACEBAS THEN
   MOVE SPACES TO LOG-TEXT
   MOVE TRACELVL TO INT-TO-CHAR
   STRING "Initializing RAM, TRACELVL = " DELIMITED BY SIZE
          INT-TO-CHAR                     DELIMITED BY SIZE
     INTO LOG-TEXT
  CALL LOGFUNC USING LOG-HOST, LOG-MODULE, LOG-TEXT
END-IF.
```

The levels provided for tracing are:

TRACEOFF
No tracing

TRACEERR
Trace only error messages

TRACEBAS
Trace basic exit function

TRACEFUL
Trace everything

Terminating the exit

For an HLL, there is little work to do. The code should ensure that all data sets are closed, all DD names are freed, and in general that all resources are released.

For an exit written in HLASM, free any additional storage that has been allocated.

The sample exit has two statements, **PERFORM CLOSE-AND-FREE-DIRFILE** and **PERFORM CLOSE-AND-FREE-MEMFILE**, that you may need to change to free your own resources.

Note: See also the discussion about initializing the exit, in "Initializing the exit" on page 441.

Coding the exit: required services

The services that are required from the exit are:

1. Validate the library.
2. Extract member contents.
3. Get Member Information (also called metadata).
4. Get Display Information (ISPF header data).

Common argument and the name of the library

The argument RAM-ARGUMENTS is passed to all of the functions providing required services. It contains the name of the library as well as additional information.

Here are the data declarations and code in the sample exit to access the library name in RAM-ARGUMENTS:

```
 01 RAM-ARGUMENTS.
    05 SOFTWARE-VERSION PIC S9(4) BINARY.
    05 REP-EXEC-SPEC-PTR          POINTER.
    05 REP-ID-SPEC-PTR            POINTER.
```

```
    05 REP-PARMS-SPEC-PTR         POINTER.

 01 REP-ID-SPEC.
    05 SOFTWARE-VERSION PIC S9(4) BINARY.
    05 RESOURCE-ID-TYPE PIC S9(4) BINARY.
    05 REP-NAME-STRING-PTR        POINTER.

 01 REP-NAME-STRING.
    05 REP-NAME-LEN PIC  9(4) BINARY.
    05 REP-NAME-GROUP.
       10 REP-NAME  PIC  X
                    OCCURS 1 TO 50 TIMES
                    DEPENDING ON REP-NAME-LEN.

GET-RAM-PARM2-RAM-ARGS.
     SET ADDRESS OF RAM-ARGUMENTS TO RAM-PARM2.
     SET ADDRESS OF REP-ID-SPEC TO REP-ID-SPEC-PTR.
     SET ADDRESS OF REP-NAME-STRING TO REP-NAME-STRING-PTR
     IF TRACELVL >= TRACEBAS THEN
       MOVE SPACES TO LOG-TEXT
       STRING "Repository name = " DELIMITED BY SIZE
              REP-NAME-GROUP DELIMITED BY SIZE
              INTO LOG-TEXT
       CALL LOGFUNC USING LOG-HOST, LOG-MODULE, LOG-TEXT
     END-IF.
```

Validate the library

The goal of the function "Validate library" is to return a return code indicating whether the library can be supported by the exit.

The key return codes are:

0 Library supported by this exit.

120 Library not supported. File Manager should try other access methods to try to read the data in the library.

(other)
Some other error occurred.

The sample COBOL exit validates a PDS by checking to see if it contains a member named $$FMN$$. If $$FMN$$ exists, the exit returns rc=0. If not, it returns rc=120.

There is a performance enhancement included in the sample code: if the library has already been validated, the validation is not repeated. This can only safely be done if the previous return code was 0 or 120. If it was some other value, the validation must be repeated. For example, if the previous validation found that the data set was in use, the validation must be repeated.

To support a customer LMS, you can replace the USER-REPOS-VALIDATE paragraph and leave the remaining code unchanged.

Get member records

In order to extract all records in a member, File Manager first makes a GetFirstMemberRecord call, followed by multiple GetNextMemberRecord calls. If there is a record, then the rc=0. At any time, the process can be normally terminated with an End Of File (RAM-RC-EOF) return code.

The GetFirstMemberRecord call requires a member name as a parameter. It is passed as follows:

```
01 MEMBER-ID-SPEC.
    05 SOFTWARE-VERSION PIC 9(4) BINARY.
    05 MEM-ID-SPEC-COLL-PTR POINTER.
01 MEM-ID-SPEC-COLL.
    05 SOFTWARE-VERSION PIC 9(4) BINARY.
    05 MEM-ID-COMPONENT-COUNT PIC 9(4) BINARY.
    05 MEM-ID-COMPONENT-PTR POINTER
           OCCURS 1 TO 10 TIMES
           DEPENDING ON MEM-ID-COMPONENT-COUNT.
01 MEM-ID-COMPONENT.
    05 MEM-ID-COMP-TYPE PIC 9(4) BINARY.
    05 MEM-ID-COMP-NAME-STRING.
       10 MEM-ID-COMP-NAME-LEN PIC 9(4) BINARY.
       10 MEM-ID-COMP-NAME-GROUP.
          15 MEM-ID-COMP-NAME PIC X
                OCCURS 1 TO 10 TIMES
                DEPENDING ON MEM-ID-COMP-NAME-LEN.
01 MEMBER-NAME-STRING.
    05 MEMBER-NAME-LEN PIC 9(4) BINARY.
    05 MEMBER-NAME-GROUP.
       10 MEMBER-NAME PIC X
                      OCCURS 1 TO 10 TIMES
                      DEPENDING ON MEMBER-NAME-LEN.
GET-RAM-PARM3-MEM-ID.
     SET ADDRESS OF MEMBER-ID-SPEC TO RAM-PARM3
     SET ADDRESS OF MEM-ID-SPEC-COLL
                 TO MEM-ID-SPEC-COLL-PTR
     IF MEM-ID-COMPONENT-COUNT NOT = 1 THEN
        SET RAM-RC-MEM-ID-SPEC-BAD TO TRUE
     ELSE
        SET ADDRESS OF MEM-ID-COMPONENT
            TO MEM-ID-COMPONENT-PTR(1)
        IF MEM-ID-COMP-TYPE NOT = 0 THEN
           SET RAM-RC-MEM-ID-SPEC-BAD TO TRUE
        END-IF
     END-IF
     IF RAM-RC-OK THEN
        SET ADDRESS OF MEMBER-NAME-STRING
           TO ADDRESS OF MEM-ID-COMP-NAME-STRING
        IF TRACELVL >= TRACEBAS THEN
          MOVE SPACES TO LOG-TEXT
          STRING "Member = " DELIMITED BY SIZE
                 MEMBER-NAME-GROUP DELIMITED BY SIZE
                 INTO LOG-TEXT
          CALL LOGFUNC USING LOG-HOST, LOG-MODULE, LOG-TEXT
        END-IF
     END-IF.
```

If you are writing a user exit in COBOL, it is unlikely you will need to change this code which gets the member name argument. This code describes the data structures and is provided for users who want to rewrite the exit into PL/I or HLASM.

Here is the code for getting the first member record:

```
GET-1ST-MEM-REC.
     PERFORM GET-RAM-PARM2-RAM-ARGS
     PERFORM VAL-REPOS-VS-RAM-ARGS
     IF RAM-RC-OK THEN
        PERFORM GET-RAM-PARM3-MEM-ID
        PERFORM USER-GET-1ST-MEM-REC
     END-IF.
```

It uses the VAL-REPOS-VS-RAM-ARGS routine to determine if the library has already been validated. If so, it does not need to be redone. Then it gets the

member name, and calls the USER-GET-1ST-MEM-REC paragraph. This is where you should insert your calls to the user library access methods.

The member record results are returned in a data area which looks like this:

```
01 REC-DATA-RETURN-AREA.
    05 REC-DATA-RETURN-ATTR1 PIC 9(4) BINARY VALUE 1.
    05 REC-DATA-RETURN-LEN1  PIC 9(4) BINARY VALUE 80.
    05 REC-DATA-RETURN-STR1  PIC X(80).
    05 REC-DATA-RETURN-END   PIC X(2)        VALUE X"FFFF".
```

The member record text is placed in REC-DATA-RETURN-STR1 and a return code of 0 returned.

If the member is empty, a RAM-RC-EOF (180) return code is returned.

Note: If you are providing support for an LMS which provides the capability to place proprietary INCLUDE statements in the member, the proprietary INCLUDEs should be **fully expanded**.

When getting the second and subsequent records in a member, the member name parameter is not passed to the exit.

Get member information (metadata)

The two calls associated with getting member metadata are GetFirstMemberInfo and GetNextMemberInfo. As in the case of getting member records, the normal return codes are 0 when member metadata is being returned, and 180 at end of file.

The argument required is not exactly the same as a member name. Instead, a filter is expected. The filter may be a member name, or it may contain characters and wildcard characters.

The wildcard characters supported by File Manager are * matching zero or more characters, and % matching exactly one character.

Here is how a filter argument is passed:

```
01 SIMPLE-FILTER-SPEC.
    05 SOFTWARE-VERSION PIC 9(4) BINARY.
    05 SIMPLE-FILTER-FIELD-ID PIC 9(4) BINARY.
       88 SIMPLE-FILTER-IS-MEMBER-NAME VALUE 210.
    05 SIMPLE-FILTER-LENGTH PIC 9(4) BINARY.
    05 SIMPLE-FILTER-MASK PIC X(64). GET-RAM-PARM3-FILTER.
     IF RAM-PARM3 = NULL THEN
        MOVE 1 TO PATTERN-LENGTH
        MOVE "*" TO PATTERN-CHARS
     ELSE
        SET ADDRESS OF SIMPLE-FILTER-SPEC TO RAM-PARM3
        IF SIMPLE-FILTER-LENGTH < 0
              OR SIMPLE-FILTER-LENGTH > 10 THEN
           SET RAM-RC-BAD-FILTER TO TRUE
        ELSE
           MOVE SIMPLE-FILTER-LENGTH TO PATTERN-LENGTH
           MOVE SIMPLE-FILTER-MASK(1:PATTERN-LENGTH)
             TO PATTERN-CHARS
        END-IF
     END-IF.
```

A COBOL nested program called MATCH is provided in the sample COBOL exit to provide function to match a filter (pattern) to a member name.

Note: File Manager strips any leading blanks prior to passing the filter (pattern), so you do not need to remove leading blanks from the filter yourself.

Note: Blanks are treated like any other character in the MATCH program, and if found, must be matched. Therefore, if the LMS returns leading blanks in member names you should strip them. If the member name includes trailing blanks, eliminate them from comparison by setting the length of the member name appropriately.

The metadata returned to File Manager has the following structure. Note that the sample COBOL exit has given some structure to the RHS, which you may want to change:

```
01 DIR-DATA-RETURN-AREA.
    05 DIR-DATA-RETURN-ATTR1 PIC 9(4) COMP-5 VALUE 20000.
    05 DIR-DATA-RETURN-LEN1  PIC 9(4) BINARY VALUE 10.
    05 DIR-DATA-RETURN-LHS   PIC X(10) VALUE " ".
    05 DIR-DATA-RETURN-ATTR2 PIC 9(4) COMP-5 VALUE 20001.
    05 DIR-DATA-RETURN-LEN2  PIC 9(4) BINARY VALUE 51.
    05 DIR-DATA-RETURN-RHS.       10 RHS-USERID PIC X(7).
       10 FILLER PIC XX VALUE " ".
       10 RHS-MOD-DATE PIC X(10).
       10 FILLER PIC X VALUE " ".
       10 RHS-MOD-TIME PIC X(8).
       10 FILLER PIC XX VALUE " ".
       10 RHS-NLINES PIC ZZZZ9 DISPLAY.
       10 FILLER PIC XX VALUE " ".
       10 RHS-CRE8-DATE PIC X(10).
       10 FILLER PIC X(4).
    05 DIR-DATA-RETURN-END   PIC X(2)        VALUE X"FFFF".
```

The member name should go into DISP-INFO-RETURN-LHS, padded with trailing blanks. The attribute information should go into DISP-INFO-RETURN-RHS. There is space for 8 to 10 characters for the member name (LHS) and the total line is limited to 80 characters. The RHS can contain the remaining characters. The size of the prompt field is variable. See “Get display information” for more information on the layout.

Get display information

Get Display returns the column headings for the member selection list panel when no member name or member name pattern is specified. The panel looks like this:

```
(Sel)  (LHS)    (Prompt) (RHS)

 Sel   Name      Prompt   Date Created    ...
 _   MEMBER01   *Browsed 08/02/2003 14:53 ...
 _   MEMBER02            12/21/2002 08:22 ...
```

The names in parentheses at the top of the illustration are used in this example to describe the columns. There is a selection prefix area called **Sel**. There is a member name field called LHS (meaning the left hand side of the prompt). There is a prompt field, sometimes blank, and sometimes filled with information from File Manager about past actions. The RHS contains any attribute information desired.

The GetDisplayInfo call returns the following static (unchanging) data structure with the heading:

```
01 DISP-INFO-RETURN-AREA.
    05 DISP-INFO-RETURN-ATTR1 PIC 9(4) COMP-5 VALUE 20000.
    05 DISP-INFO-RETURN-L11   PIC 9(4) BINARY VALUE 10.
    05 DISP-INFO-RETURN-L12   PIC 9(4) BINARY VALUE 10.
    05 DISP-INFO-RETURN-LHS   PIC X(8) VALUE " Name  ".
```

```
05 DISP-INFO-RETURN-ATTR2 PIC 9(4) COMP-5 VALUE 20001.
05 DISP-INFO-RETURN-L21   PIC 9(4) BINARY VALUE 51.
05 DISP-INFO-RETURN-L22   PIC 9(4) BINARY VALUE 51.
05 DISP-INFO-RETURN-RHS   PIC X(51)
    VALUE "  Id           Changed         Recs   Created".
05 DISP-INFO-RETURN-ATTR3 PIC 9(4) COMP-5 VALUE 20002.
05 DISP-INFO-RETURN-L31   PIC 9(4) BINARY VALUE 8.
05 DISP-INFO-RETURN-L32   PIC 9(4) BINARY VALUE 8.
05 DISP-INFO-RETURN-PROMP PIC X(8) VALUE " Prompt".
05 DISP-INFO-RETURN-ATTR4 PIC 9(4) COMP-5 VALUE 20003.
05 DISP-INFO-RETURN-L41   PIC 9(4) BINARY VALUE 4.
05 DISP-INFO-RETURN-L42   PIC 9(4) BINARY VALUE 4.
05 DISP-INFO-RETURN-PROMP PIC X(4) VALUE "Sel ".
05 DISP-INFO-RETURN-END   PIC X(2)        VALUE X"FFFF".
```

Supporting multiple library management systems

Considerations when supporting multiple library management systems:

- File Manager only calls one exit named FMNCRAEX.
- It is suggested that you write a separate exit for each LMS you need to support. Then, write a single FMNCRAEX that calls these LMS exits. Use the "Validate library" function (see “Validate the library” on page 443) to determine which LMS exit to use for a given library. This has many advantages, but the main one is that it is easy to add or remove supported LMSs.

Writing the exit in HLASM

The following considerations apply to writing the exit in HLASM:

- There is currently no sample exit written in HLASM. You will have to write this yourself.
- Use the HLASM register save, call and return macros provided by Language Environment. The caller code in File Manager is LE-enabled and assumes the callee is LE enabled.
- Use argument 1, referred to in the sample COBOL program as RAM-WORK-AREA-PTR, in all the calls to the exit so that File Manager remembers the address of your root block of storage. You can set this to any value and File Manager will remember it and provide it back to your exit on every call.
- Get memory during the INIT-RAM call and free all memory during the TERM-RAM call.
- Use the TEST sample COBOL program to call your HLASM exit routine and test it outside of File Manager.

Appendix G. File Manager audit records

This section describes the structure of, and information recorded in, File Manager audit records. This information applies to audit records produced by File Manager Version 11 and later only. Previous versions of File Manager used a different record format to that described here.

An audit log record may occupy one or more physical records in the audit file. When an audit log record requires two or more physical records, the second and subsequent log records are continuation records.

Each File Manager audit record is comprised of three parts:
- Record header
- Data item reference section
- Data section

Part 1 - record header

Table 70 shows the information contained in the record header. The record header is at the start of each physical record that makes up the audit log record.

Table 70. Audit record: Header

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	BIT(8)	1	LOG_FLAGS	Log flags
		1...		LOG_CONT	Continuation indicator
		.1..		LOG_IMS	Log record for FM/IMS
		..1.		LOG_DB2	Log record for FM/DB2
		...1		LOG_CICS	Log record for FM/CICS
		.... 111.		*	
		1		LOG_V2IND	Log record Version 2 or later
1	(1)	UNSIGNED BINARY	1	LOG_VERSION	Version number
2	(2)	UNSIGNED BINARY	4	LOG_RECNUM	Record count within the current audit session
6	(6)	CHARACTER	5	LOG_ID	Log record constant 'FMIMS' (IMS eyecatcher) 'FMDB2' (DB2 eyecatcher) 'FMCIC' (CICS eyecatcher) 'FMLOG' (base eyecatcher)
11	(B)	CHARACTER	3	*	
14	(E)	UNSIGNED PACKED DECIMAL	4	LOG_DATE	Date - yyyymmdd
18	(12)	UNSIGNED PACKED DECIMAL	4	LOG_TIME	Time - hhmmssth

© Copyright IBM Corp. 2000, 2011

Table 70. Audit record: Header (continued)

Offsets		Type	Len	Name (Dim)	Description
Dec	Hex				
22	(16)	CHARACTER	8	LOG_SYSID	MVS system ID (CVTSNAME)
30	(1E)	CHARACTER	8	LOG_USERID	Userid
38	(26)	CHARACTER	8	LOG_SSID	FM/DB2 System ID,IMS Subsystem name or CICS Applid
46	(2E)	CHARACTER	8	LOG_JOBNAME	Job Name
54	(36)	CHARACTER	8	LOG_JOBID	Job ID
62	(3E)	CHARACTER	1	LOG_SESSID	Session ID
63	(3F)	CHARACTER	8	LOG_FUNCOD	File manager function code
71	(47)	CHARACTER	8	LOG_DBNAME	IMS Database name
79	(4F)	CHARACTER	23	*	

Part 2 - data item reference section

The data item reference section is in the first physical record of the audit record, immediately after the audit record header. It consists of 2 parts: the data item reference header, followed by the data item reference array. Table 71 shows the information contained in the data item reference header.

Table 71. Audit record: Data item reference header

Offsets		Type	Len	Name (Dim)	Description
Dec	Hex				
102	(66)	UNSIGNED BINARY	2	LOG_ITEM_CNT	Number of data items in the data section of the audit record.
104	(68)	UNSIGNED BINARY	2	LOG_REC_COUNT	Number of physical records required to write the audit log record. A value greater than 1 implies there are continuation records.

The data item reference array is an array of entries, each entry describes a single data item. The LOG_ITEM_CNT value in the data item reference header (see Table 71) specifies the number of elements in the array.

Table 72 shows the information contained in a data item reference array element.

Table 72. Audit record: Data item reference section

Offsets		Type	Len	Name (Dim)	Description
Dec	Hex				
0	(0)	STRUCTURE	8	LOG_DATA_ITEM(*)	
0	(0)	UNSIGNED BINARY	2	LOG_ITEM_CODE	Item data code. See Table 75 on page 451.
2	(2)	CHARACTER	1	LOG_DATATYPE	Data type code. See Table 76 on page 452.
3	(3)	CHARACTER	1	*	Reserved/filler

Table 72. Audit record: Data item reference section (continued)

Offsets		Type	Len	Name (Dim)	Description
Dec	Hex				
4	(4)	SIGNED BINARY	4	LOG_ITEM_DATLEN	Fullword length

Part 3 - data section

The data section contains all of the data for the various data items defined in the data item reference section of the record. Each data item is stored in sequence and occupies exactly the length given in the data item reference section. There are no space bytes between data items.

The data items depend on the File Manager component and the record type. In general terms, each record type is associated with some data items that are always written, followed by some data items that are specific to the record type and the component that is in use.

Table 73 shows the information contained in the data section:

Table 73. Audit record: Data section

Offsets		Type	Len	Name (Dim)	Description
Dec	Hex				
0	(0)	STRUCTURE	*	LOG_DATAAREA	
0	(0)	UNSIGNED BINARY	2	LOG_DATALEN	Length of LOG_DATA
2	(2)	STRUCTURE	*	LOG_DATA	logging data

Note: For continuation records, the data for the incomplete data item continues from the previous record, immediately after the continuation record header, as shown in Table 74.

Table 74. Audit record: continued record

Offsets		Type	Len	Name (Dim)	Description
Dec	Hex				
0	(0)	CHARACTER	102	LOG_HEADER_FIXED	The Audit record header (see Table 70 on page 449)
102	(66)	UNSIGNED BINARY	2	LOG_CONT_LEN	Length of continued data
104	(68)	CHARACTER	*	LOG_CONT_DATA	The data for the incomplete data item (and possibly other data items)

Table 75. Data items common to all audit records

Data item	Usage	Usual length
4	Audit record type	1
3	File Manager component that wrote the record. A fixed field with the possible values: **0** Base component **1** FM/IMS component **2** FM/DB2 component **3** FM/CICS component	1

Table 75. Data items common to all audit records (continued)

Data item	Usage	Usual length
91[1]	File Manager function code	8
92	File Manager internal function code. This will usually be the same as the File manager function code.	8

Notes:

1. This data item is not included in any FM/IMS audit records except for the Template record (Audit record type T).

Table 76. Data item types common to all audit records

Data item type code	Data item type
B	Binary
C	Character (default)
U	Unformatted

File Manager (base) audit records

This section describes the File Manager (base) audit records.

File Manager (base) audit data items

The following table shows the data items that appear in File Manager (base) audit log records.

Table 77. Data items that appear in File Manager (base) audit log records

Data item identifier	Description	Usual length
15	Data set name	54
21	Template data set member name	54
22	Template type	1
31	Type of record	1
32	Record	Variable
41	Copy utility used	8
42	Copied count	4
43	Replaced count	4
44	Not copied count	4
45	IO error count	4
51	Output data set name	54
52	KSDS key location	4
53	KSDS key length	4
101	Template fd99 segment	36
102	Template	Variable

File Manager (base) audit record types

The following table shows the File Manager (base) record types - each record type has a single character code.

Table 78. File Manager (base) record types

Audit Record Type Code	Usage
A	New record.
D	Deleted record.
E	Record counts.
O	Original record. An 'O' record is written for an update (change) operation, and is followed by an 'R' record.
R	Replacement record. An 'R' record is written for an update (change) operation, and is preceded by an 'O' record.
S	Start of session record. Each 'S' record should have a corresponding 'Z' record.
T	Template record. The template is written in internal format.
V	Read record.
W	Written record. The record was written using the File Manager REXX WRITE function.
Z	End of session record.

Data items specific to File Manager (base) audit records

The following table shows the data items specific to File Manager (base) audit records.

Table 79. Data items specific to File Manager (base) audit records

Audit Record Type	Data items
A	31, 32
D	21, 32
E	41, 42, 43, 44, 45
O	31, 32
R	31, 32
S	15, 51, 52, 53
T	21, 22, 101, 102
V	31, 32
W	31, 51, 32
Z	See Table 75 on page 451

FM/DB2 audit records

This section describes the FM/DB2 audit records.

FM/DB2 audit data items

Table 80 shows the data items that appear in FM/DB2 audit log records.

Table 80. Data items that appear in FM/DB2 audit log records

Data item identifier	Description	Usual length
14	Input/Output indicator	1

Table 80. Data items that appear in FM/DB2 audit log records (continued)

Data item identifier	Description	Usual length
21	Template data set member name	10
22	Template type (input or output)	1
101	Template fd99 segment	36
102	Template	Variable
2003	Current server	16
2004	Current SQLID	8
2005	Connection type	5
2011	Object location	16
2012	Object database	8
2013	Object table space	8
2014	Object owner	8
2015	Object name	128
2031	Type of DB2 row	1
2032	DB2 row (data)	Variable
2041	Row count (in)	4
2042	Row count editor	4
2043	Row count (out)	4
2051	SQL Stmt type	3
2052	SQL Reason code	4
2053	Affected row count	4
2071	DB2 Object type (for DDL statements)	3
2072	DB2 Privilege type (For Grant/Revoke statements)	3
2501	SQL statement	Varies
2502	DB2 command	Varies

FM/DB2 audit record types

Table 81 shows the FM/DB2 audit record types - each record type has a single character code.

Table 81. FM/DB2 audit record types

Audit Record Type Code	Usage
A	New record or row.
D	Deleted record or row.
E	Enumeration record - reports the number of records or rows read or copied.
I	Identification record - contains information about the data set or DB2 object that is being processed.
J	Identification record - specific to DB2. Contains the SQL SELECT statement used to access DB2 object(s).

Table 81. FM/DB2 audit record types (continued)

Audit Record Type Code	Usage
O	Original record or row. An 'O' record is typically written for an update (change) operation, and is followed by an 'R' record.
Q	SQL statement. This is specific to FM/DB2. Contains the text for some SQL statement.
R	Replacement record or row. An 'R' record is typically written for an update (change) operation, and is preceded by an 'O' record.
S	Start of session record. An 'S' record is written at the start of auditing for some FM/DB2 function. Each 'S' record should have a corresponding 'Z' record.
T	Template record. Contains an FM template used to process a data set or DB2 object. The template is written in internal format.
Z	End of session record. A 'Z' record is written at the end of auditing for some FM/DB2 function. Each 'Z' record should have a corresponding 'S' record
2	DB2 command. This is specific FM/DB2. Contains the text for some DB2 command.

Data items specific to FM/DB2 audit records

Table 82. Data items specific to File Manager (base) audit records

Audit Record Type	Data items
A	2031, 2032
D	2031, 2032
E	2041, 2042, 2043
I	14, 2011, 2012, 2013, 2014, 2015
J	2501
O	2031, 2032
Q	2051, 2052, 2053, 2071, 2072, 2501
R	2031, 2032
S	2003, 2004, 2005
T	21, 22, 101, 102
Z	See Table 75 on page 451
2	2502

FM/IMS audit records

This section describes the FM/IMS audit records.

FM/IMS audit data items

The following table shows the data items that appear in FM/IMS audit log records.

Table 83. Data items that appear in FM/IMS audit log records

Data item identifier	Description	Type	Length
21	Template/view/criteria set data set and member name	CHARACTER	<=54
22	Template type: **C** Criteria set **T** Template **V** View	CHARACTER	1
101	Template FD99 segment		36
102	Template data		Variable
1001	Database name	CHARACTER	8
1002	Segment name	CHARACTER	8
1003	Segment level	UNSIGNED BINARY	1
1004	Segment description (as specified in template)	CHARACTER	0 or 15
1005	Identifier of template layout used to format data	SIGNED BINARY	4
1006	Segment update flags: Segment occurrence is a child of the deleted segment Dependents of deleted segment have not been logged Not used	BIT(8) 1... .... .1.. .... ..11 1111	1
1007	Concatenated key value		Variable
1008	Segment data		Variable
1009	Checkpoint ID	CHARACTER	8
1010	Session flags: Dynamic PSB? BMP region? Using view/criteria set? Using secondary index? Using key values? Initial load? Load replaces existing segments? REXX procedure in-stream? Last 8 bits not used.	BIT(16) 1... .... .1.. .... ..1. ....1 .... 1... .... .1.. .... ..1. 1	2
1011	Primary database name	CHARACTER	8
1012	PSB data set name	CHARACTER	0 or 44
1013	PSB name	CHARACTER	0 or 8
1014	Position of selected DB PCB in PSB's PCB list	SIGNED BINARY	2
1015	Name of secondary index	CHARACTER	0 or 8
1016	Application group name (AGN)	CHARACTER	0 or 8

Table 83. Data items that appear in FM/IMS audit log records (continued)

Data item identifier	Description	Type	Length
1017	REXX procedure data set name	CHARACTER	0 or 44
1018	REXX procedure member name	CHARACTER	0 or 8
1019	Key values data set and member name	CHARACTER	0 or 54

FM/IMS audit record types

The following table shows the FM/IMS audit record types - each record type has a single character code.

Table 84. FM/IMS audit record types

Audit Record Type Code	Usage
A	Inserted segment record.
B	Rollback record.
C	Checkpoint record.
D	Deleted segment record. FM/IMS generates one record for the segment occurrence that is deleted and one record for each of its dependent segments.
F	Delete failure record. When issued it indicates that a Delete failed, so the set of D records that were generated for this Delete should be ignored.
O	Segment replace record - before image.
R	Segment replace record - after image.
S	Start of session record.
T	Template record.
Z	End of session record.

Data items specific to FM/IMS audit records

The following table shows the data items specific to FM/IMS audit records.

Table 85. Data items specific to FM/IMS audit records

Audit Record Type	Data items[1]
A	1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008
B	1009
C	1009
D	1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008
F	No data items except the ones that are common to all FM/IMS record types.
O	1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008
R	1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008
S	1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019
T	21, 22, 101, 102
Z	No data items except the ones that are common to all FM/IMS record types.

Table 85. Data items specific to FM/IMS audit records (continued)

Audit Record Type	Data items[1]
Notes: 1. Data items that are common to all FM/IMS audit records (4, 3, 92) are not listed.	

FM/CICS audit records

This section describes the FM/CICS audit records.

FM/CICS audit data items

The following table shows the data items that appear in FM/CICS audit log records.

Table 86. Data items that appear in FM/CICS audit log records

Data item identifier	Description	Usual length
21	Template data set member name	54
22	Template type	1
31	Type of record	1
32	Record	Variable
101	Template fd99 segment	36
102	Template	Variable
3001	Resource type. The meaning of the bits from high-order to low-order is: • File • Intrapartition transient data queue • Extraparition transient data queue • Temporary storage queue • 4 bits reserved	1
3002	Resource name	16
3003	CICS applid of resource	8
3004	Resource pool name	8
3005	DDname	8
3006	Data set name	44

FM/CICS audit record types

The following table shows the FM/CICS audit record types - each record type has a single character code.

Table 87. FM/CICS audit record types

Audit Record Type Code	Usage
A	New record.
D	Deleted record.
O	Original record. An 'O' record is written for an update (change) operation, and is followed by an 'R' record.

Table 87. FM/CICS audit record types (continued)

Audit Record Type Code	Usage
R	Replacement record. An 'R' record is written for an update (change) operation, and is preceded by an 'O' record.
S	Start of session record. Each 'S' record should have a corresponding 'Z' record.
T	Template record. The template is written in internal format.
V	Read record.
Z	End of session record.

Data items specific to FM/CICS audit records

The following table shows the data items specific to FM/CICS audit records.

Table 88. Data items specific to FM/CICS audit records

Audit Record Type	Data items
A	31, 32
D	21, 32
O	31, 32
R	31, 32
S	3001, 3002, 3003, 3004, 3005, 3006
T	21, 22, 101, 102
V	31, 32
Z	See Table 75 on page 451

Appendix H. Support resources and problem solving information

This section shows you how to quickly locate information to help answer your questions and solve your problems. If you have to call IBM support, this section provides information that you need to provide to the IBM service representative to help diagnose and resolve the problem.

For a comprehensive multimedia overview of IBM software support resources, see the IBM Education Assistant presentation "IBM Software Support Resources for System z Enterprise Development Tools and Compilers products" at http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp?topic=/com.ibm.iea.debugt/debugt/6.1z/TrainingEducation/SupportInfoADTools/player.html.

- "Searching IBM support Web sites for a solution"
- "Obtaining fixes" on page 462
- "Receiving support updates through e-mail notification" on page 463
- "Receiving support updates through RSS feeds" on page 464
- "If you need to contact IBM Software Support" on page 464

Searching IBM support Web sites for a solution

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

- "Searching the information center"
- "Searching product support documents"
- "IBM Support Assistant" on page 462

Searching the information center

You can find this publication and documentation for many other products in the IBM System z Enterprise Development Tools & Compilers information center at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp. Using the information center, you can search product documentation in a variety of ways. You can search across the documentation for multiple products, search across a subset of the product documentation that you specify, or search a specific set of topics that you specify within a document. Search terms can include exact words or phrases, wild cards, and Boolean operators.

To learn more about how to use the search facility provided in the IBM System z Enterprise Development Tools & Compilers information center, you can view the multimedia presentation at http://publib.boulder.ibm.com/infocenter/pdthelp/v1r1/index.jsp?topic=/com.ibm.help.doc/InfoCenterTour800600.htm.

Searching product support documents

Use the System z Enterprise Development Tools & Compilers information center or the IBM support site at www.ibm.com/software/support to search for the latest, most complete information that might help you resolve your problem.

© Copyright IBM Corp. 2000, 2011

When you access the IBM support site, you can specify any of the following products for which you want information to be displayed:

- Application Performance Analyzer for z/OS
- Debug Tool for z/OS
- Enterprise COBOL for z/OS
- Enterprise PL/I for z/OS
- Fault Analyzer for z/OS
- File Manager for z/OS
- Optim Move for DB2
- WebSphere Developer Debugger for System z
- Workload Simulator for z/OS and OS/390 Support

When you access the IBM support site, you can also use the IBM Support Portal to customize the support information to be displayed and save product names that you specify. There is also a search facility provided with the IBM Support Portal that allows you to narrow the search scope and search only product support documents for the products that you specify. The IBM Support Portal can be accessed through the IBM support site at www.ibm.com/software/support or directly at www.ibm.com/support/entry/portal. For information about customizing your IBM support site experience using the IBM Support Portal, refer to https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos?lang=en_us.

IBM Support Assistant

The IBM Support Assistant (also referred to as ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. It provides quick access to support-related information. You can use the IBM Support Assistant to help you in the following ways:

- Search through IBM and non-IBM knowledge and information sources across multiple IBM products to answer a question or solve a problem.
- Find additional information through product and support pages, customer news groups and forums, skills and training resources and information about troubleshooting and commonly asked questions.

In addition, you can use the built in Updater facility in IBM Support Assistant to obtain IBM Support Assistant upgrades and new features to add support for additional software products and capabilities as they become available.

For more information, and to download and start using the IBM Support Assistant for IBM System z Enterprise Development Tools & Compilers products, please visit http://www.ibm.com/support/docview.wss?rs=2300&context=SSFMHB&dc=D600&uid=swg21242707&loc=en_US&cs=UTF-8&lang=en.

General information about the IBM Support Assistant can be found on the IBM Support Assistant home page at http://www.ibm.com/software/support/isa.

Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes and other updates are available, the following information is available from the IBM support site. You can also view the following information from the IBM Support Portal when you specify the applicable products.

- Latest PTFs for Application Performance Analyzer for z/OS
- Latest PTFs for Debug Tool for z/OS
- Latest PTFs for Fault Analyzer for z/OS
- Latest PTFs for File Export for z/OS
- Latest PTFs for File Manager for z/OS
- Latest fixes for Optim Move for DB2
- Latest PTFs for WebSphere Studio Asset Analyzer for Multiplatforms
- Latest PTFs for Workload Simulator for z/OS and OS/390

When you find a fix that you are interested in, click the name of the fix to read its description and to optionally download the fix.

The IBM Support Portal is a way for you to specify specific products for which you want to display support information. The Support Portal can be accessed through the IBM support site at www.ibm.com/software/support or directly at www.ibm.com/support/entry/portal. For information about how to customize your IBM support site experience using the IBM Support Portal, refer to https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos?lang=en_us.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/handbook.html.

Receiving support updates through e-mail notification

To receive e-mail notifications about fixes and other software support news, follow the steps below. Additional information is provided at http://www.ibm.com/support/docview.wss?rs=615&uid;=swg21172598.

1. Go to the IBM software support site at http://www.ibm.com/software/support.
2. Click **Request e-mail updates** in the **Additional support links** section of the page.
3. Click any **My Notifications** link on the page that is displayed.
4. If you have already registered for **My notifications**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
5. In the **My notifications** tool, click the **Subscribe** tab to specify products for which you want to receive e-mail updates.
6. To specify Problem Determination Tools products, click **Other software** and then select the products for which you want to receive e-mail updates, for example, **Debug Tool for z/OS** and **File Manager for z/OS**.
7. To specify a COBOL or PL/I compiler, click **Rational** and then select the products for which you want to receive e-mail updates, for example, **Enterprise COBOL for z/OS** .
8. After selecting all products that are of interest to you, scroll to the bottom of the list and click **Continue**.
9. Determine how you want to save your subscription. You can use the default subscription name or create your own by entering a new name in the **Name** field. It is recommended that you create your own unique subscription name using a something easily recognized by you. You can create a new folder by

entering a folder name in the **New** field or select an existing folder from the pulldown list. A folder is a container for multiple subscriptions.

10. Specify the types of documents you want and the e-mail notification frequency.
11. Scroll to the bottom of the page and click **Submit**.

To view your current subscriptions and subscription folders, click **My subscriptions**.

If you experience problems with the **My notifications** feature, click the **Feedback** link in the left navigation panel and follow the instructions provided.

Receiving support updates through RSS feeds

To receive RSS feeds about fixes and other software support news, go to the following web site and select the products in which you are interested:

- IBM Software Support RSS feeds.

If you need to contact IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli®, Lotus®, and Rational® products, as well as DB2 and WebSphere® products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online

  Go to the Passport Advantage Web site at http://www.lotus.com/services/passport.nsf/ WebDocs/Passport_Advantage_Home and click **How to Enroll**.

  By phone

  For the phone number to call in your country, go to the IBM Software Support Web site at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.
- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at https://techsupport.services.ibm.com/ssr/login.
- For customers with IBMLink, CATIA, Linux, S/390®, iSeries®, pSeries®, zSeries®, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to

the contacts page of the *IBM Software Support Handbook* on the Web at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. "Determining the business impact"
2. "Describing problems and gathering information"
3. "Submitting problems" on page 466

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1
: The problem has a **critical** business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
: The problem has a **significant** business impact. The program is usable, but it is severely limited.

Severity 3
: The problem has **some** business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
: The problem has **minimal** business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently.

To save time, if there is a Mustgather document available for the product, refer to the Mustgather document and gather the information specified. Mustgather documents contain specific instructions for submitting your problem to IBM and gathering information needed by the IBM support team to resolve your problem. To determine if there is a Mustgather document for this product, go to the product support page and search on the term Mustgather. At the time of this publication, the following Mustgather documents are available:

- Mustgather: Read first for problems encountered with Application Performance Analyzer for z/OS: http://www.ibm.com/support/docview.wss?rs=2300 &context=SSFMHB&q1=mustgather&uid=swg21265542&loc=en_US&cs=utf-8 ⟨=en
- Mustgather: Read first for problems encountered with Debug Tool for z/OS: http://www.ibm.com/support/docview.wss?rs=615&context=SSGTSD &q1=mustgather&uid=swg21254711&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with Fault Analyzer for z/OS:http://www.ibm.com/support/docview.wss?rs=273&context=SSXJAJ &q1=mustgather&uid=swg21255056&loc=en_US&cs=utf-8&lang=en

- Mustgather: Read first for problems encountered with File Manager for z/OS: http://www.ibm.com/support/docview.wss?rs=274&context=SSXJAV&q1=mustgather&uid=swg21255514&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with Enterprise COBOL for z/OS: http://www.ibm.com/support/docview.wss?rs=2231&context=SS6SG3&q1=mustgather&uid=swg21249990&loc=en_US&cs=utf-8&lang=en
- Mustgather: Read first for problems encountered with Enterprise PL/I for z/OS: http://www.ibm.com/support/docview.wss?rs=619&context=SSY2V3&q1=mustgather&uid=swg21260496&loc=en_US&cs=utf-8&lang=en

If the product does not have a Mustgather document, please provide answers to the following questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

Online

Click **Open service request** on the IBM Software Support site at http://www.ibm.com/software/support/probsub.html. In the Other support tools section, select IBMLink to open an Electronic Technical Response (ETR). Enter your information into the appropriate problem submission form.

By phone

Call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the contacts page of the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

After a Problem Management Record (PMR) is open, you can submit diagnostic MustGather data to IBM using one of the following methods:

- FTP diagnostic data to IBM
- If FTP is not possible, email diagnostic data to techsupport@mainz.ibm.com. You must add PMR xxxxx bbb ccc in the subject line of your email. xxxxx is your PMR number, bbb is your branch office, and ccc is your IBM country code. Click here http://itcenter.mainz.de.ibm.com/ecurep/mail/subject.html for more details.

Always update your PMR to indicate that data has been sent. You can update your PMR online or by phone as described above.

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 give 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, 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 at any time without notice.

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

© Copyright IBM Corp. 2000, 2011

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 information 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.

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 illustrates 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.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

Programming interface information

This Customization Guide documents information that is **not** intended to be used as Programming Interfaces of File Manager.

Trademarks

A current list of IBM trademarks is available on the Web at "Copyright and trademark information", `http://www.ibm.com/legal/copytrade.shtml`.

Bibliography

File Manager publications

File Manager Customization Guide, SC19-3133
File Manager User's Guide and Reference, SC19-3134
File Manager User's Guide and Reference for DB2 Data, SC19-3135
File Manager User's Guide and Reference for IMS Data, SC19-3136
File Manager User's Guide and Reference for CICS, SC19-3137
File Manager Fact Sheet, G325-2429
File Manager License Information, GC19-3132
File Manager Program Directory, GI10-8858

Related publications for DB2

See the following publication for your version of DB2:

DB2 Administration Tool for z/OS User's Guide and Reference

Related publications for CICS

IP CICS Sockets Guide, SC31-8807

See the publications for your version of CICS:

CICS Transaction Server for z/OS Installation Guide
CICS Transaction Server for z/OS CICS Customization Guide
CICS Transaction Server for z/OS RACF Security Guide
CICS Transaction Server for z/OS CICS Resource Definition Guide

Related publications for Communications Server

z/OS Communications Server: IP System Administrator Commands, SC31-8781
z/OS Communications Server: IP Configuration Guide, SC31-8775

Related publications for z/OS®

z/OS DFSMS Access Method Services for Catalogs, SC26-7394
z/OS ISPF Dialog Developer's Guide, SC34-4821
z/OS ISPF Planning and Customizing, GC34-4814
z/OS ISPF Services Guide, SC34-4819
z/OS ISPF User's Guide Vol II, SC34-4823
z/OS MVS System Management Facilities (SMF), SA22-7630
z/OS Security Server RACF Command Language Reference, SA22-7687
z/OS TSO/E Command Reference, SA22-7782

See the following publication for your version of DFSORT:

DFSORT Installation and Customization

Softcopy publications

The following collection kits contain File Manager and related product publications in BookManager® format:

- *z/OS Software Products Collection Kit*, SK3T-4270
- *OS/390 Collection Kit*, SK2T-6700

© Copyright IBM Corp. 2000, 2011

Index

A

ABENDCC processing option 289
ACBLIB option 362
access
  denying 194
  granting 194
access levels
  for SAF rules 53, 137, 261
AFP-authorization
  authorizing the COBOL compiler library 15, 16
AGN option 409
aliases for DBRMs 119
alternate libraries for ISPF 7
ALTLIB 103
AMODE, FMNSECUR 42
APAR PM31842 52, 136, 211, 260
APF-authorization
  authorizing FMNSMF for audit logging 49
  authorizing the COBOL compiler library 21
  diagnosing problems 21
  planning to run File Manager APF-authorized 7
  planning to run FM/DB2 APF-authorized 103
  planning to run FM/IMS APF-authorized 172
  running File Manager APF-authorized 21
ASCII processing option 290
ASCII translation tables, changing 29
ATTACH option 331
AUDDATAC processing option 290
audit
  controlling with SAF rules 53, 137, 261
audit facility
  activating changes to SYS1.PARMLIB 203
  customizing for File Manager 52
  customizing for File Manager base function
    FMN0POPT controlled auditing 49
    SAF-rule controlled auditing 51
  customizing for FM/CICS 260
    FMN0POPT controlled auditing 255
    SAF-rule controlled auditing 259
  customizing for FM/DB2 131, 135, 136
  customizing for FM/IMS
    FMN0POPT controlled auditing 201
    SAF-rule controlled auditing 205
  deciding how to customize 104
  FMNSMF load module 21, 49
  FMNSMFX sample job 50
  problems using SMF 21
audit facility *(continued)*
  recording to a data set 50, 133, 202, 256
  setting default options 49
  using SMF 49, 132, 201, 255
  using the AUDITHLQ option 49
  using the AUDITLOG option 49
audit log records
  determining if they should be written 54, 138, 262
AUDIT option 332
audit records
  File Manager (base) 452
  FM/CICS 458
  FM/DB2 453
  FM/IMS 455
  information held by 449
  structure 449
audit trail exit 196
AUDITBROWSE option 333
AUDITHLQ processing option 291
auditing
  controlling using SAF 51, 135, 259
AUDITLOG processing option 292
AUDMGMTC processing option 293
AUDPQTY processing option 293
AUDSQTY processing option 293
AUDSTORC processing option 293
AUDSUNIT processing option 294
AUDUNIT processing option 294
AUTOSAVE option 362
AUXDATAC processing option 294
AUXDSN processing option 295
AUXHLQ processing option 295
AUXMDSN processing option 296
AUXMGMTC processing option 296
AUXSTORC processing option 297

B

batch JCL skeleton
  changing for File Manager base function 24
  changing for FM/DB2 127
  changing for FM/IMS 189
  customizing for the COBOL compiler library 24
batch return codes
  customizing for File Manager 26
bibliography 470
binding DB2 for FM/DB2 119
  DB2 plans FMN2PLAN and FMN2GEN 120
  possible SQL errors 120
binding DB2 for OAM functions 20
BLP, controlling access 37
BMP mode 164

C

CATOWNER option 350
CATOWNERCDRM option 348
changing the ASCII translation tables 29
checklists
  File Manager customization 3
  FM/CICS customization 232
  FM/DB2 customization 100
  FM/IMS customization 163
CHGAFREQ option 363
COBDBCS processing option 298
COBDPC processing option 298
COBEXTND processing option 299
COBMAXRTN processing option 299
COBMCASE processing option 299
COBOL
  ABENDS306 from a batch job 21
  ADATA capability 13
  adding the COBOL compiler to the LINKLIST 15, 16
  authorizing the COBOL compiler library 21
  customizing for using COBOL copybooks 13
  options that must not be fixed in IGYCDOPT 15
  supported COBOL compilers 13
  suppressing COBOL compiler warning messages 15, 16
  using the FMNCOB DD statement 15, 16
  using the FMNTERM DD statement 15, 16
COBOL compiler library in the batch JCL skeleton 24
COBOL compiler library in the FM/IMS batch JCL skeleton 190
comments
  in the FMN0PARM member 416
  in the FMN1PARM member 422
  in the FMN2PARM member 428
  in the FMN3PARM member 433
comments, sending xviii
COMPAT option 363
COMPLANG processing option 300
compression exit 73
concatenating libraries to the LINKLIST
  FM/DB2 102
  FM/IMS 171
CONNECT option 352
controller parameters, IMS regions 166
CPYCPYN option 334
creating a segmented table space for FM/DB2 121
CREPLACEn processing option 300
CSYSLIBnn processing option 300
customer support
  See Software Support
customization
  verifying 219

© Copyright IBM Corp. 2000, 2011

customization checklists
File Manager base function 3
FM/CICS 232
FM/DB2 100
FM/IMS 163
customization of File Manager base function
verifying 93
customization of FM/CICS
verifying 275
customization of FM/DB2
verifying 157
customization of Japanese component
verifying 94
customization of Korean component
verifying 94
customizing for national languages
checklist for File Manager base function 61
checklist for FM/CICS 269
checklist for FM/DB2 151
checklist for FM/IMS 213
customizing IBM Problem Determination Tools Common Components 281
customizing to use LMS libraries
checklist for File Manager base function 67
CYLHD processing option 301

D

DASD, controlling fullpack access 35
data sharing groups, DB2 121, 122
database data set allocation exit 196
DATAHDR processing option 301
DB2 Admin Launchpad 112
DB2 catalog, granting access 109
DB2 data sharing groups, specifying for FM/DB2 121, 122
DB2 objects read access
controlling auditing of 144
DB2 objects update access
controlling auditing of 143
DB2 subsystems, specifying for FM/DB2 121, 122
DB2, binding FM/DB2 119
DB2CLIB option 334
DB2ELIB option 335
DB2LLIB option 335
DB2MLIB option 336
DB2PLIB option 336
DB2PROC option 337
DB2RLIB option 337
DB2SLIB option 338
DB2TLIB option 338
DBCS-capable terminal
FM/CICS 270
DBDLIB1 option 364
DBDLIBn option 364
DBRC option 365
DBRMs 119
default options, changing
FMN0POPT 23
FMN1AGNT 177
FMN1POPI 177
FMN1POPT 177
FMN2POPT 126
default options, changing *(continued)*
for File Manager base function 23
for FM/DB2 126
for FM/IMS 177
DESC option 338, 365, 410
DFSORT libraries
adding to the batch JCL skeleton 25
DFSORT, using to improve File Manager performance 17
DFSORT as primary sort product 18
DFSORT is not the primary sort product 19
DFSORT SVC 19
ICEDFSRB alias 18
ICEDFSRT alias 18
DFSRRC00 option 366
DFSVSAMP option 366
DISPLAY option 339
display translation tables
changing for File Manager base function 28
changing for languages other than English 61, 151, 213, 269
FMNTRTBS 28
PRTTRANS option 28
TRBRW translate table 28
TRINV translate table 28
TRPRT translate table 28
TRUP translate table 28
distribution library names 3
DLI mode 164
DSN8EAE1 157
DSNAME option 412, 418, 424, 429
DSPINC processing option 301
DSPMAX processing option 302
DSPMIN processing option 302
DSPNUM processing option 302
DUMP processing option 303
DYNACB option 367
DYNALLOC option 367
dynamic PSBs 165
DYNPRFN option 368
DYNPRFX option 368

E

Eclipse, plug-in for 285
EDITCAPS option 297, 353
EDITFREQ option 369
EDITPROC DSN8EAE1 157
EDMAXVIRT processing option 303
encryption exit 73
EOD processing option 303
example of FMN1AGNT macro 185
example of FMN1POPD macro 185
example of FMN1POPI macro 185
examples of FMN2POPI macros 125
examples of FMN2SSDM macro 122
exits
FM/IMS security exit 191
I/O exit 73
LMS exit 67, 439
scrambling exit 85
experienced installers fast path xv

F

fast path for experienced installers xv
File Manager functions
controlling auditing of 57
File Manager options 289
File Manager subsystem options
DSNAME 412
MEMBER 413
SAF_CTRL 411
SAF_SMFNO 412
File Manager user I/O exit 73
close call 81
exit control block description 75
exit protocol 74
initialization call 80
installing the exit 82
open call 81
read and write calls 81
termination call 81
using the I/O exit control block 79
writing the exit 74
File Manager user scrambling exit 85
exit control block description 86
exit protocol 85
installing the exit 90
using the scrambling exit control block 89
writing the exit 85
fixes, obtaining 462
FM Server
FMNCAS 435
FM/CICS functions
controlling auditing of 265
FM/CICS subsystem options
DSNAME 429
MEMBER 430
SAF_CTRL 429
SAF_SMFNO 429
FM/DB2
adding to the DB2 Admin Launchpad 112
binding DB2, required customization 119
creating a segmented table space, required customization 121
DB2 subsystem selection panel, required customization 121
DBRMs 119
granting access to DB2 catalog 109
running multiple versions 120
starting from an external application 114
starting from DB2 Admin Tool 113
FM/DB2 commands
customizing 125
FM/DB2 customization, required 99
FM/DB2 options
CATOWNER 350
CONNECT 352
EDITCAPS 353
FMN2POPI macro 348
LIST 355
SSIDCMD1 359
SSIDCMD2 359
FM/DB2 Options 331
FM/DB2 subsystem options
ATTACH 331

FM/DB2 subsystem options *(continued)*
AUDIT 332
AUDITBROWSE 333
CATOWNERCDRM 348
CPYCPYN 334
DB2CLIB 334
DB2ELIB 335
DB2LLIB 335
DB2MLIB 336
DB2PLIB 336
DB2PROC 337
DB2RLIB 337
DB2SLIB 338
DB2TLIB 338
DESC 338
DISPLAY 339
DSNAME 424
FMN2SSDM macro 331
LDFDDLN 339
LOCATION 340
LOCATION_NICKNAME 340
LODINDN 341
MEMBER 424
OPTEVT1 341
OPTEVT2 341
OPTEVT3 342
OPTEVT4 342
PLAN 342
PLAN2 342
RBXWRKN 343
ROGUNLN 343
SAF_CTRL 423
SAF_SMFNO 423
SLDJCL1 343
SLDJCL2 344
SLDJCL3 344
SLDJCL4 344
SSID 345
STMJCL1 345
STMJCL2 345
STMJCL3 346
STMJCL4 346
TABLE_LOCKING 346
TMPDDLN 347
TYPE 347
UNLPUNN 348
UNLUNLN 348
FM/IMS
customizing the security environment 191
FM/IMS audit trail exit 196
FM/IMS Batch JCL skeleton
customizing for the COBOL compiler library 190
FM/IMS database data set allocation exit 196
FM/IMS function security 166
FM/IMS functions
controlling access to 191
FM/IMS IMS initialization exit 197
FM/IMS IMS termination exit 197
FM/IMS options 361
ACBLIB 362
AUTOSAVE 362
CHGAFREQ 363
COMPAT 363
DBDLIB1 364
FM/IMS options *(continued)*
DBDLIBn 364
DBRC 365
DESC 365
DFSRRC00 366
DFSVSAMP 366
DYNACB 367
DYNALLOC 367
DYNPRFN 368
DYNPRFX 368
EDITFREQ 369
GSGNAME 370
IEBFREQ 370
IMSAUDLG 371
IMSBKO 371
IMSNBA 372
IMSOBA 372
IRLM 373
IRLMNAME 373
LOADFREQ 376
LOCKMAX 377
MACLIB 379
MAXGN 380
PARDLI 380
PROCOPTB 381
PROCOPTP 382
PROCOPTX 382
PROCOPTY 383
PSBLIB1 383
PSBLIBn 383
PSBTYPE 384
PSBTYPES 385
READONLY 385
REGTYPES 386
RESLIB1 386
RESLIBn 386
SFMNSLIB 389
SSID 389
TMINAME 390
TPLLIB1 390
TPLLIBn 390
UACBLIB 391
UAGNS 391
UAUTOSAV 392
UBUF 392
UDBDLIB 393
UDBRC 393
UDFSVSMP 394
UIEBFREQ 394
UIMSBKO 395
UIMSNBA 396
UIRLM 396
ULOADFRQ 397
ULOCKMAX 398
UMACLIB 399
UPARDLI 399
UPROCOPB 400
UPROCOPP 400
UPROCOPX 401
UPROCOPY 402
UPSBLIB 403
UPSBTYPE 403
URECON 404
URESLIB 404
URSR 405
UTPLLIB 405
VCURULE 406
FM/IMS options *(continued)*
VSMPMEM 406
XDOPTLB 407
FM/IMS Options
AGN 409
DESC 410
FMN1AGNT macro 409
SSID 410
FM/IMS security
AGN security 167
common exit parameters 197
controlling access 191
controlling access to individual functions 193
controlling access to read-only functions 192
controlling access to update functions 192
exit type A parameters 198
exit type D parameters 198
exit type I parameters 199
FMN1SXT 166, 167, 196
invoking FMN1SXT 197
invoking the security exit 197
RAS security 167
sample programs 200
securing dynamic PSBs in BMP mode 167
security exit 166
security exit parameters 198
types of security exit 196
FM/IMS security exit sample programs
FMN1AXIT 200
FMN1CXIT 200
FMN1SECC 200
FMN1UMDS 200
FMN1XITA 200
FMN1XITC 200
FM/IMS subsystem options
DSNAME 418
MEMBER 418
SAF_CTRL 417
SAF_SMFNO 418
FM/IMS, customizing IMS 174
FM/IMS, declaring dynamic PSBs 174
FM/IMS, providing a DOPT ACBLIB data set 174
FMAUDIT macro 411, 417, 423, 429
FMEDITOR processing option 304
FMN0MENU, national language version of 63
FMN0PARM
options specified in 411
FMN0PARM definitions
customizing 413
FMN0PARM member
tags 413
FMN0POPT 23
FMN0RETC 26
FMN1AGNT 177
FMN1AGNT macro 409
FMN1AGNT macro, example 185
FMN1FTEX JCL skeleton 189
FMN1MENU, modifying 214
FMN1PARM
options specified in 417

FMN1PARM definitions
customizing 419
FMN1PARM member
tags 419
FMN1POPD macro, example 185
FMN1POPI 177
FMN1POPI macro, example 185
FMN1POPT 177
FMN1SXT 167, 196
FMN1UMDM 214
FMN1UMDN 216
FMN1UMDP 177
FMN1UMDQ 217
FMN1UMDS 200
FMN2DENU, modifying 154
FMN2FTEX JCL skeleton 128
FMN2FTSL JCL skeleton 127
FMN2GEN 120
FMN2GEN, default FM/DB2 reverse engineering plan 120
FMN2INIT 102
FMN2MENU, modifying 152
FMN2PARM
options specified in 422
FMN2PARM definitions
customizing 424
FMN2PARM member
defining 100
tags 425
FMN2PLAN 120
FMN2POPI macro 348
FMN2POPT 126
FMN2RESS 103
FMN2SSDM macro 331
FMN2SSDM macro, examples 122
FMN2UMDB 128
FMN2UMDD 154
FMN2UMDE 128
FMN2UMDM 152
FMN2UMDN 156
FMN2UMDP 127
FMN2UMDQ 156
FMN3MENU, modifying 270
FMN3PARM
options specified in 428
FMN3PARM definitions
customizing 430
FMN3PARM member
tags 430
FMN3UMDM 270
FMN3UMDN 272
FMN3UMDQ 272
FMNBDIRS 20
FMNCAS
FM Server 435
FMNCOB DD statement 15, 16
FMNCRACJ 70
FMNCRAEX 70
FMNFTEXC JCL skeleton 24
FMNIOXCB 75
FMNIOXCC 75
FMNIOXCP 75
FMNIOXEA 75
FMNIOXEC 75
FMNIOXEP 75
FMNIOXHF 75
FMNS macro
exit routine environment 42
parameter list contents 43
registers at entry 42
registers at exit 43
syntax 31, 41
FMNSCXCA 86
FMNSCXCC 86
FMNSCXCP 86
FMNSCXEA 85
FMNSCXEC 85
FMNSCXEP 85
FMNSECUR 31, 38, 42
AMODE 42
installing into LPA 42
installing using a usermod 43
located below 16M 42
return codes 43
FMNSMF, authorizing for audit logging 21, 49
FMNTERM DD statement 15, 16
FMNTRJPN 65
FMNTRKOR 66, 156, 217, 272
FMNTRTBS 28
FMNUMODA 29
FMNUMODB 25
FMNUMODH 83
FMNUMODJ 65
FMNUMODK 66
FMNUMODM 63
FMNUMODN 65
FMNUMODP 23
FMNUMODQ 66
FMNUMODR 26
FMNUMODS 43
FMNUMODT 29
FMNUMODU 83
FMNUMODX 61
FMOPTMOD macro 412, 418, 424, 429
fullpack access to DASD volumes 35
controlling with FMNSECUR 41

G

granting access to DB2 catalog 109
GSGNAME processing option 370

H

HEADERPG processing option 304
High Level Assembler copybooks 17
HLASM
customizing for using HLASM copybooks 17
LANG=HLASM 17
HLASM copybooks 17
HLDBCS processing option 304
HLMAXRTN processing option 305
HLNOALIGN processing option 305
HSYSLIBnn processing option 305

I

I/O exit 73
IBM Problem Determination Tools Common Components Distribution library names 281
IBM Problem Determination Tools Common Components Library names 281
IBM Problem Determination Tools Common Components Target and distribution library names 281
IBM Support Assistant, searching for problem resolution 462
IEBFREQ option 370
IGYCDOPT 15
IMS databases supported 163
IMS initialization exit 197
IMS processing modes
BMP mode 164
DLI mode 164
IMS region controller parameters 166
IMS subsystem security 166
IMS subsystems
controlling access by read-only functions 193
controlling access by update functions 193
controlling access to 191
controlling access to individual subsystems by individual functions 194
IMS subsystems, specifying for FM/IMS 185
IMS termination exit 197
IMSAUDLG option 371
IMSBKO option 371
IMSNBA option 372
IMSOBA option 372
information centers, searching for problem resolution 461
installation
experienced installers xv
verifying 219
installation of File Manager base function
verifying 93
installation of FM/CICS
verifying 275
installation of FM/DB2
verifying 157
installation of Japanese component
verifying 94
installation of Korean component
verifying 94
Internet
searching for problem resolution 461
IRLM option 373
IRLMNAME option 373
ISPF alternate libraries 7
ISPF command table
defining File Manager base function 11
defining FM/DB2 107
defining FM/IMS 173
ISPF messages
translating for File Manager base function 63
translating for FM/DB2 152

ISPF messages *(continued)*
translating for FM/IMS 214
ISPF Primary Option Menu
adding File Manager base function 11
adding FM/DB2 107
adding FM/IMS 173
ISPFPACK processing option 306

J

Japanese component
verifying customization 94
verifying installation 94
Japanese national language
3277KN or 3278KN terminals 65
File Manager base function 65
FM/CICS 272
FM/DB2 156
FM/IMS 216
FMNTRJPN 65
JCL skeletons
adding the COBOL compiler 15, 16
File Manager base function 24
FM/DB2 127
FM/IMS 189

K

knowledge bases, searching for problem resolution 461
Korean component
verifying customization 94
verifying installation 94
Korean national language
3277KN or 3278KN terminals 66
File Manager base function 66
FM/CICS 272
FM/DB2 156
FM/IMS 217
FMNTRKOR 66

L

language environment
considerations 22
Language Environment
runtime library 17
LANGUAGE option keywords 64, 153, 215, 272, 283
LANGUAGE processing option 306
LDFDDLN option 339
libraries
concatenating to LINKLIST 6
Library Management System libraries (LMSs)
See also LMS sample exit
adding to the batch JCL skeleton 25
customizing File Manager base function 67
library names 3
LINKLIST
concatenating File Manager libraries 6
LINKLIST, add the COBOL compiler library 15, 16
LINKLIST, concatenating FM/DB2 libraries 102
LINKLIST, concatenating FM/IMS libraries 171
LIST option 355
LKEYDATAC processing option 374
LKEYMGMTC processing option 374
LKEYPQTY processing option 374
LKEYSQTY processing option 375
LKEYSTORC processing option 375
LKEYSUNIT processing option 375
LKEYVOLn processing option 376
LMS processing option 307
LMS sample exit 439
common argument 442
FMNCRACJ, JCL to install 71
getting display information 446
getting member information 445
getting member records 443
HLASM considerations 447
initializing the exit 441
library name 442
library validation 443
main program and first argument 439
metadata 445
return codes 440
supporting multiple LMSs 447
terminating 442
tracing 441
LMSUBSYS processing option 308
LOADFREQ option 376
LOCATION option 340
LOCATION_NICKNAME option 340
LOCKMAX option 377
LODINDN option 341
LOGDATAC processing option 377
LOGMGMTC processing option 378
logon messages
translating for FM/CICS 270
logon procedure, TSO 7, 102, 171
LOGPQTY processing option 378
LOGSQTY processing option 378
LOGSTORC processing option 378
LOGSUNIT processing option 379
LOGUNIT processing option 379
LPA, installing FMNSECUR 42

M

MACLIB option 379
MAXGN option 380
MEMBER option 413, 418, 424, 430
members
included 415, 421, 427, 432
messages
translating for File Manager base function 62
translating for FM/DB2 152
translating for FM/IMS 214
translating for IBM Problem Determination Tools Common Components 282
messages, using translated 64, 153, 215, 272, 283
MQREPHLQ processing option 308
MSGUPPER processing option 309

N

national language default
for File Manager base function 23
for FM/DB2 127
for FM/IMS 184
national languages
customizing File Manager base function 61
customizing FM/CICS 269
customizing FM/DB2 151
customizing FM/IMS 213
notices 469

O

OAM functions, binding DB2 for 20
OPTEVT1 option 341
OPTEVT2 option 341
OPTEVT3 option 342
OPTEVT4 option 342
options
ABENDCC 289
ASCII 290
AUDDATAC 290
AUDITHLQ 291
AUDITLOG 292
AUDMGMTC 293
AUDPQTY 293
AUDSQTY 293
AUDSTORC 293
AUDSUNIT 294
AUDUNIT 294
AUXDATAC 294
AUXDSN 295
AUXHLQ 295
AUXMDSN 296
AUXMGMTC 296
AUXSTORC 297
COBDBCS 298
COBDPC 298
COBEXTND 299
COBMAXRTN 299
COBMCASE 299
COMPLANG 300
CREPLACEn 300
CSYSLIBnn 300
CYLHD 301
DATAHDR 301
DSPINC 301
DSPMAX 302
DSPMIN 302
DSPNUM 302
DUMP 303
EDITCAPS 297
EDMAXVIRT 303
EOD 303
FMEDITOR 304
HEADERPG 304
HLDBCS 304
HLMAXRTN 305
HLNOALIGN 305
HSYSLIBnn 305
ISPFPACK 306
LANGUAGE 306
LKEYDATAC 374
LKEYMGMTC 374

options *(continued)*
LKEYPQTY 374
LKEYSQTY 375
LKEYSTORC 375
LKEYSUNIT 375
LKEYVOLn 376
LMS 307
LMSUBSYS 308
LOGDATAC 377
LOGMGMTC 378
LOGPQTY 378
LOGSQTY 378
LOGSTORC 378
LOGSUNIT 379
LOGUNIT 379
MQREPHLQ 308
MSGUPPER 309
PAD 310
PAGESIZE 310
PAGESKIP 310
PDATAC 311
PLI31DIGIT 311
PLI63BIT 311
PLIGRAPHIC 312
PLIMAXRTN 312
PLIUNALIGN 312
PMGMTC 313
PRINTDSN 313
PRINTLEN 313
PRINTOUT 314
PRTCLASS 314
PRTDISP 315
PRTMGMTC 315
PRTPQTY 315
PRTSQTY 316
PRTSTORC 316
PRTSUNIT 316
PRTTRANS 317
PRTUNIT 317
PSTORC 317
PSYSLIBnn 318
PTRDATAC 318
PUNIT 318
RECLIMIT 318
RKEYDATAC 387
RKEYMGMTC 387
RKEYPQTY 388
RKEYSQTY 388
RKEYSTORC 388
RKEYSUNIT 388
RKEYVOLn 389
SEC 319
SFMNMOD1 319
SHOWDATAC 357
SHOWMGMTC 357
SHOWPQTY 357
SHOWSQTY 358
SHOWSTORC 358
SHOWSUNIT 358
SHOWUNIT 359
SMFNO 320
TAPELBL 320
TDATAC 321
TEMPHLQ 321
TERMTYPE 322
TMGMTC 323
TRACECLS 323
options *(continued)*
TRACEOUT 323
TRCDATAC 324
TRCMGMTC 324
TRCPQTY 324
TRCSQTY 325
TRCSTORC 325
TRCSUNIT 325
TRCUNIT 326
TSTORC 326
TUNIT 326
USEIOX 326
VSAUTO 327
VSCHGAUTO 327
VSCHGFRQ 328
VSSAVE 328
WIDEPRT 329
XKEYDATAC 407
XKEYMGMTC 408
XKEYPQTY 408
XKEYSQTY 408
XKEYSTORC 408
XKEYSUNIT 409
XKEYUNIT 409
overview xiii

P

PAD processing option 310
PAGESIZE processing option 310
PAGESKIP processing option 310
panels
translating for File Manager base function 63
translating for FM/CICS 271
translating for FM/DB2 153
translating for FM/IMS 215
translating for IBM Problem Determination Tools Common Components 283
panels, using translated 64, 153, 215, 272, 283
PARDLI option 380
PARMLIB
options specified in 411
PDATAC processing option 311
performance
improving for FM/DB2 103
using DFSORT to improve 18
PL/I
customizing for using PL/I include books 17
Language Environment runtime library 17
PL/I include books 17
PLAN option 342
PLAN2 option 342
PLI31DIGIT processing option 311
PLI63BIT processing option 311
PLIGRAPHIC processing option 312
PLIMAXRTN processing option 312
PLIUNALIGN processing option 312
plug-in for Eclipse 285
PM31842, APAR 52, 136, 211, 260
PMGMTC processing option 313
Primary Option Menu panel
customizing FM/DB2 108
print translation tables
changing for File Manager base function 28
changing for languages other than English 61, 151, 213, 269
FMNTRTBS 28
PRTTRANS option 28
TRBRW translate table 28
TRINV translate table 28
TRPRT translate table 28
TRUP translate table 28
PRINTDSN processing option 313
PRINTLEN processing option 313
PRINTOUT processing option 314
problem determination
describing problems 465
determining business impact 465
submitting problems 466
problems with APF-authorizing, diagnosing 21
processing modes
BMP mode 164
DLI mode 164
PROCOPTB option 381
PROCOPTP option 382
PROCOPTX option 382
PROCOPTY option 383
profile names for protected functions 44
programming interface information 470
providing a national language version of FMN2DENU
FM/DB2 154
PRTCLASS processing option 314
PRTDISP processing option 315
PRTMGMTC processing option 315
PRTPQTY processing option 315
PRTSQTY processing option 316
PRTSTORC processing option 316
PRTSUNIT processing option 316
PRTTRANS processing option 317
PRTUNIT processing option 317
PSB processing
dynamic PSBs 165
static PSBs 165
PSBLIB1 option 383
PSBLIBn option 383
PSBTYPE option 384
PSBTYPES option 385
PSTORC processing option 317
PSYSLIBnn processing option 318
PTRDATAC processing option 318
publications
bibliography 470
PUNIT processing option 318

R

RACF, using to control access 31
RBXWRKN option 343
READONLY option 385
RECLIMIT processing option 318
REGTYPES option 386
release level
display using VER command 93, 158
RESLIB1 option 386
RESLIBn option 386

return codes
customizing for batch utilities 26
return codes, FMNSECUR 43
reverse engineering plan, FM/DB2 120
RKEYDATAC processing option 387
RKEYMGMTC processing option 387
RKEYPQTY processing option 388
RKEYSQTY processing option 388
RKEYSTORC processing option 388
RKEYSUNIT processing option 388
RKEYVOLn processing option 389
ROGUNLN option 343

S

SAF 33
used to control File Manager auditing 51
used to control FM/CICS auditing 259
used to control FM/DB2 auditing 135
SAF rules
access levels 53, 137, 261
how they control auditing 53, 137, 261
SAF_CTRL option 411, 417, 423, 429
SAF_SMFNO option 412, 418, 423, 429
SAF-rule controlled auditing
testing 145
SAF-rule tables 59, 146, 266
sample jobs
FMN1AXIT 200
FMN1CXIT 200
FMN1IVP 219
FMN1XITA 200
FMN1XITC 200
FMN2BIND 119
FMN2CGEN 121
FMN2CIX 104
FMN2VER 157
FMNBDIRS 20
FMNCRACJ 70
FMNCRAEX 70
FMNIOXEA 75
FMNIOXEC 75
FMNIOXEP 75
FMNIOXHF 75
FMNSCXEA 85
FMNSCXEC 85
FMNSCXEP 85
FMNSMFX 50
FMNVERIF 93
SCEERUN, LE runtime library 17
SCEERUN, LE runtime library, in LINKLIST 6
scrambling exit 85
SEC processing option 319
security
controlling access to BLP 37
controlling access using FMNSECUR 38
controlling access using RACF 31
controlling access using SAF 33
controlling fullpack access to DASD volumes 35
FM/IMS function 166
security *(continued)*
FMN1SXT 196
FMNSECUR exit 31
IMS subsystem 166
security environment
customizing for File Manager 31
customizing for FM/IMS 191
SERVER statement
syntax 435
setting the LANGUAGE option
FM/CICS 269
SFMNMOD1 processing option 319
SFMNSLIB option 389
SHOWDATAC processing option 357
SHOWMGMTC processing option 357
SHOWPQTY processing option 357
SHOWSORT command
using to determine if DFSORT is available 19
SHOWSQTY processing option 358
SHOWSTORC processing option 358
SHOWSUNIT processing option 358
SHOWUNIT processing option 359
SICELINK, used by File Manager 18
SLDJCL1 option 343
SLDJCL2 option 344
SLDJCL3 option 344
SLDJCL4 option 344
SMF (system management facilities)
problems with audit logging 21
using for audit logging 49, 132, 201, 255
SMFNO processing option 320
Software Support
contacting 464
describing problems 465
determining business impact 465
receiving updates 463, 464
submitting problems 466
SORTLPA, used by File Manager 18
specifications
continuing across multiple lines 416, 422, 428, 433
SQL errors from binding DB2 120
SSID option 345, 389, 410
SSIDCMD1 option 359
SSIDCMD2 option 359
static PSBs 165
STMJCL1 option 345
STMJCL2 option 345
STMJCL3 option 346
STMJCL4 option 346
subsystems, DB2 121, 122
subsystems, IMS 185
supported IMS databases 163
SVC, DFSORT 19
syntax notation used in this book xvi

T

TABLE_LOCKING option 346
tags
as entries in FMN0PARM member 413
as entries in FMN1PARM member 419
tags *(continued)*
as entries in FMN2PARM member 425
as entries in FMN3PARM member 430
example of usage 414, 419, 425, 431
rules for specifying 413, 419, 425, 430
TAPELBL processing option 320
target library names 3
TDATAC processing option 321
TEMPHLQ processing option 321
TERMTYPE processing option 322
TMGMTC processing option 323
TMINAME option 390
TMPDDLN option 347
TPLLIB1 option 390
TPLLIBn option 390
TRACECLS processing option 323
TRACEOUT processing option 323
trademarks 470
translating Japanese message text
IBM Problem Determination Tools Common Components 283
translating Korean message text
IBM Problem Determination Tools Common Components 284
translating logon messages
FM/CICS 270
translating message text
File Manager base function 62
FM/DB2 152
FM/IMS 214
IBM Problem Determination Tools Common Components 282
ISPF messages 63, 152, 214
modifying FMN0MENU 63
modifying FMN1MENU 214
modifying FMN2MENU 152
modifying FMN3MENU 270
translating object list utility statements
modifying FMN2DENU 154
translating panel text
File Manager base function 63
FM/CICS 271
FM/DB2 153
FM/IMS 215
IBM Problem Determination Tools Common Components 283
TRCDATAC processing option 324
TRCMGMTC processing option 324
TRCPQTY processing option 324
TRCSQTY processing option 325
TRCSTORC processing option 325
TRCSUNIT processing option 325
TRCUNIT processing option 326
TSO ALTLIB 103
TSO logon procedure, modifying
for File Manager base function 7
for FM/DB2 102
for FM/IMS 171
using a CLIST or REXX exec 102
TSTORC processing option 326
TUNIT processing option 326
type A audit trail exit 196
type A exit parameters 198
type D database data set allocation exit 196

type D exit parameters 198
type I exit parameters 199
type I IMS initialization exit 197
TYPE option 347
type T IMS termination exit 197

U

UACBLIB option 391
UAGNS option 391
UAUTOSAV option 392
UBUF option 392
UDBDLIB option 393
UDBRC option 393
UDFSVSMP option 394
UIEBFREQ option 394
UIMSBKO option 395
UIMSNBA option 396
UIRLM option 396
ULOADFRQ option 397
ULOCKMAX option 398
UMACLIB option 399
UNLPUNN option 348
UNLUNLN option 348
unprotected functions 44
UPARDLI option 399
UPROCOPB option 400
UPROCOPP option 400
UPROCOPX option 401
UPROCOPY option 402
UPSBLIB option 403
UPSBTYPE option 403
URECON option 404
URESLIB option 404
URSR option 405
USEIOX option
  indicating a user I/O exit 73
  providing a user I/O exit 82
USEIOX processing option 326
user I/O exit 73
user I/O exit copybooks
  FMNIOXCB 75
  FMNIOXCC 75
  FMNIOXCP 75
user I/O exit sample programs
  FMNIOXEA 75
  FMNIOXEC 75
  FMNIOXEP 75
  FMNIOXHF 75
user scrambling exit 85
user scrambling exit copybooks
  FMNSCXCA 86
  FMNSCXCC 86
  FMNSCXCP 86
user scrambling exit sample programs
  FMNSCXEA 85
  FMNSCXEC 85
  FMNSCXEP 85
usermods
  FMN1UMDM 214
  FMN1UMDN 216
  FMN1UMDP 177
  FMN1UMDQ 217
  FMN1UMDS 200
  FMN2UMDB 128
  FMN2UMDD 154
  FMN2UMDE 128
usermods *(continued)*
  FMN2UMDM 152
  FMN2UMDN 156
  FMN2UMDP 127
  FMN2UMDQ 156
  FMN3UMDM 270
  FMN3UMDN 272
  FMN3UMDQ 272
  FMNUMODA 29
  FMNUMODB 25
  FMNUMODH 83
  FMNUMODJ 65
  FMNUMODK 66
  FMNUMODM 63
  FMNUMODN 65
  FMNUMODP 23
  FMNUMODQ 66
  FMNUMODR 26
  FMNUMODS 43
  FMNUMODT 29
  FMNUMODU 83
  FMNUMODX 61
users who should use this book xv
using the translated messages and panels
  File Manager base function 64
  FM/CICS 272
  FM/DB2 153
  FM/IMS 215
  IBM Problem Determination Tools Common Components 283
UTPLLIB option 405

V

VCURULE option 406
VER command 93, 158
  using to determine APF-authorization 21
VSAM default editor and browser 12
VSAUTO processing option 327
VSCHGAUTO processing option 327
VSCHGFRQ processing option 328
VSMPMEM option 406
VSSAVE processing option 328

W

WebSphere MQ
  customizing to access 13
WIDEPRT processing option 329

X

XDOPTLB option 407
XKEYDATAC processing option 407
XKEYMGMTC processing option 408
XKEYPQTY processing option 408
XKEYSQTY processing option 408
XKEYSTORC processing option 408
XKEYSUNIT processing option 409
XKEYUNIT processing option 409

Product Number: 5655-W47

Printed in USA

SC19-3133-01