IBM File Manager for z/OS®
Customization Guide
14.1.15

Note

Before using this document, read the general information under Notices on page 587.

Edition Notice

This edition (published October 2021) applies to Version 14 Release 1 Modification Level 15 of IBM® File Manager for z/OS® (program number 5655-Q42) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Customization Guide

This publication provides information needed to plan for, customize, maintain, and diagnose problems with the File Manager for z/OS base component, Db2® component (FM/DB2), IMS™ component (FM/IMS), and CICS® component (FM/CICS).

About this document

These topics provide information needed to plan for, customize, maintain, and diagnose problems with IBM® File Manager for z/OS®.

File Manager comprises four components:

- File Manager base function
- IBM® File Manager for z/OS® Db2® component (FM/Db2)
- IBM® File Manager for z/OS® IMS™ component (FM/IMS)
- IBM® File Manager for z/OS® CICS® component (FM/CICS)

To install and use FM/Db2, FM/IMS, or FM/CICS, you **must** first install the same version of File Manager base function. You cannot use FM/Db2, FM/IMS, or FM/CICS with any other version of File Manager. You can install any of FM/Db2, FM/IMS, and FM/CICS in any order, at any time after you have installed File Manager base function.

This book is divided into five sections:

- Customizing File Manager on page 12, which provides information about the base function of File Manager.
- Customizing File Manager Db2 Component on page 150.
- Customizing File Manager IMS Component on page 242.
- Customizing File Manager CICS Component on page 323.
- The appendixes provide information about the File Manager options, the library management system exit, and how to maintain File Manager.

You will need to read Part 1. Read Parts 2 to 4 if you intend to install that component.

If you intend to use FM/CICS with the ADFzCC server, or intend to use the File Manager plug-in for Eclipse, you will need to read Customizing the ADFz Common Components server on page 558.

Parts 1 to 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.

This book also provides information needed to customize the Japanese and Korean language versions of File Manager and the FM/Db2, FM/IMS, and FM/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 to work with various storage media and maintain data.

File Manager provides production and development logical file manipulation for Websphere MQ queue data, z/OS® Unix files, CICS® files, TS and TD queues, and QSAM and VSAM data sets. VSAM data sets include ESDS, KSDS, RRDS, and VRRDS. File Manager provides applications programmers 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 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 formats. In addition, selection of data can be performed using simple REXX-like conditions that can easily be tailored by the programmer. All of the features available online to manipulate data can also be applied in batch jobs.

You can access File Manager functions in several ways:

- In full-screen mode as an ISPF or CICS® application.
- Through an Eclipse plug-in.
- For routine tasks, you can use File Manager in batch jobs using control statements.

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

FM/Db2

The File Manager Db2® component extends this functionality to the Db2® environment in the following ways:

- 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 embedded 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

FM/IMS

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.

FM/CICS

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 also incorporates access to CICS® resources from batch, ISPF, or the File Manager Eclipse plug-in.

FM/CICS uses Interactive Panel Viewer, a feature of ADFz 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.

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, see the following fast path sections in the File Manager for z/OS 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 for z/OS Program Directory.

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

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.

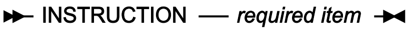
  The ———> symbol indicates that the statement syntax is continued on the next line.

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

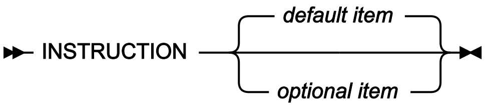
  The ——>< indicates the end of a statement.

  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).

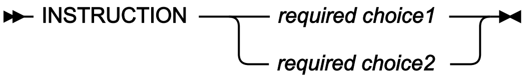
  Variables appear in all lowercase letters in a special typeface (for example, *integer*). They represent user-supplied names or values.
- If punctuation marks, parentheses, or such symbols are shown, they must be entered as part of the syntax.
- Required items appear on the horizontal line (the main path).
  Figure 1.

- Optional items appear below the main path. If the item is optional and is the default, the item appears above the main path.
  Figure 2.

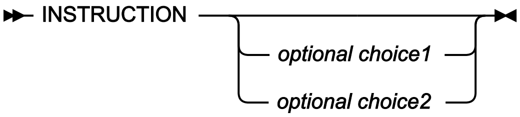
  

- When you can choose from two or more items, they appear vertically in a stack.
  If you **must** choose one of the items, one item of the stack appears on the main path.
  Figure 3.

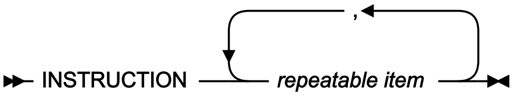
  If choosing one of the items is optional, the whole stack appears below the main path.

Figure 4.



- 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.

  Figure 5.



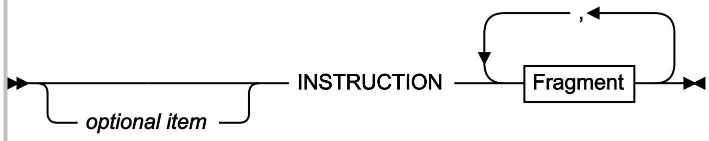
A repeat arrow above a stack indicates that you can make more than one choice from the stacked items, or repeat a single choice.

The following example shows how the syntax is used.

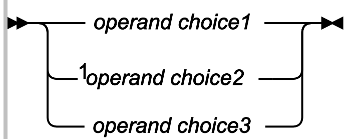
Syntax diagram

❶ ❷ ❸

Figure 6.



Fragment



Notes:

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

❶

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

❷

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

❸

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.

Summary of changes

October 2021, V14R1M15

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 15. Here are the major changes to this document from the previous edition for Version 14 Release 1.

- New keyword, CERTRUST, added to the FMN4POPT options module. For more information, see Customizing miscellaneous options in FMN4POPT on page 383.

These changes, and smaller corrections and additions, are indicated by a "|" change bar in the left margin of the page.

March 2021, V14R1M13

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 13. Here are the major changes to this document from the previous edition for Version 14 Release 1.

- Random and repeatable scrambling can now be optimized with a new option. The option can be customized at installation time (default behavior for scrambling). The default behavior can be overridden by specifying the option when copying and scrambling data. The option can be set to one of the following values: `OFF`, `ON`, or `MAX`. See OPSCRAM on page 409.

These changes, and smaller corrections and additions, are indicated by a "|" change bar in the left margin of the page.

October 2020, V14R1M12

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 12. Here are the major changes to this document from the previous edition for Version 14 Release 1.

- Support for using the IMS™ catalog to generate IMS™ database templates. See Templates on page 254, DYNTPLT on page 481, VCURUDT on page 525, and VCURULE on page 526.

These changes, and smaller corrections and additions, are indicated by a "|" change bar in the left margin of the page.

June 2020, V14R1M11

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 11. It contains minor changes and clarifications to the information in the previous version.

- A sample Thai translation table named FMNTRTHT is distributed in FMN.SFMNSAM1. See Changing the print and display translation tables for languages other than English on page 98.

These changes, and smaller corrections and additions, are indicated by a "|" change bar in the left margin of the page.

March 2020, V14R1M10

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 10. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-08.

- FM/IMS now supports dynamic PSBs when IMS™ management of ACBs is enabled. For more information, see IMS management of ACBs on page 252 and Providing a DOPT ACBLIB data set on page 262. Refer to FMN1POPD and FMN1POPI macros on page 469 for details of the macro parameters that are added (ACBSHR; CATALIAS; REGCATLG) or changed (ACBMGMT; BSDSHLQ; DBDLIBn; DBRC; DFSDF; DYNACB; DYNPRFX; PROCLIB; PSBLIBn).
- XDOPTLB on page 527 describes a change to the behavior of the XDOPTLB parameter, and explains a check that you might need to perform if XDOPTLB was previously set to N.
- A new topic explains how to secure access to IBM® MQ resources when using File Manager functions. See Setting up the security environment for IBM MQ on page 73.

These changes, and smaller corrections and additions, are indicated by a "|" change bar in the left margin of the page.

SC27-9042-08: December 2019, V14R1M9

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 8. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-07.

- New DSINFO File Manager option provides additional information in batch reports about input and output data sets for the DSP, DSM, DSC, and FCH commands. See DSINFO on page 398.

SC27-9042-07: September 2019, V14R1M8

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 8. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-06.

- New Database Access Control facility determines whether a user is authorized to access an IMS™ database through an FM/IMS function running in BMP mode. See The Database Access Control facility on page 280.
- New BDY option specifies whether fields that start beyond the current record boundary should be suppressed. See BDY on page 393.
- Template repository data sets can now be specified using PARMLIB options. See Setting up File Manager to use a template repository on page 41.

SC27-9042-06: June 2019, V14R1M7

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 7. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-05.

- Support for invoking File Manager edit models from an ISPF Edit session. Refer to Implementing File Manager edit models in ISPF Edit on page 27.
- PassTicket support for FM/CICS logon. See Modifying and submitting FMN3INST on page 333 and Modifying and submitting FMN3PRDU on page 336.

SC27-9042-05: March 2019, V14R1M6

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 6. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-04.

- File Manager Service Provider for IBM® z/OS® Connect Enterprise Edition now enables client applications to access IMS™ and Db2® data sources using z/OS® Connect. Refer to Customizing File Manager Service Provider for z/OS Connect on page 105.

SC27-9041-04: September 2018, V14R1M4

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 4. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-03.

- New option specifies the method by which JCL data is processed when running FCH or DSC with JCL=YES. See JCL on page 405.
- New template repository facility. See Setting up File Manager to use a template repository on page 41.

SC27-9041-03: June 2018, V14R1M3

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 3. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-02.

- New option specifies whether VSAM RLS support is active. See RLS on page 419.

SC27-9041-02: March 2018, V14R1M2

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 2. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-01.

- File Manager service provider enables customers to create client applications to access IBM Z data sources using z/OS Connect. See Customizing File Manager Service Provider for z/OS Connect on page 105.
- New FMN2SSDM macro statement AUTO_COMMIT. See AUTO_COMMIT on page 436.

SC27-9041-01: December 2017, V14R1M1

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 1. Here are the major changes to this document from the previous edition for Version 14 Release 1, SC27-9041-00.

- New option specifies whether VSAM RLS support is active. See RLS on page 419.
- Updated product names to reflect current offerings.

SC27-9041-00: September 2017, V14R1M0

This edition of the document provides information applicable to File Manager Version 14 Release 1 Modification Level 0. Here are the major changes to this document from the previous edition, for Version 13 Release 1, SC19-4118-10.

- Full support for TSO/ISPF LIBDEF invocation of all functions and features using FMNELIBD. See Preparing File Manager to run with LIBDEFs on page 21.
- New topics Preparing File Manager to run with LIBDEFs on page 21 and Invocation of File Manager using LIBDEFs on page 22.

Part I. Customizing File Manager

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: Summary of steps for customizing File Manager and the operating environment on page 13 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
__ 1	Concatenate File Manager libraries to the LINKLIST. See Concatenating libraries to the LINKLIST on page 16
__ 2	Modify the TSO logon procedure. See Modifying the TSO logon procedure on page 17.
__ 3	Enable/Register File Manager See Enable/Register File Manager on page 20.
__ 4	Add File Manager to the ISPF menu. See Adding File Manager to the ISPF menu on page 25.
__ 5	Define File Manager in an ISPF command table. See Defining File Manager in an ISPF command table on page 26.
__ 6	Make File Manager the default VSAM editor. See Making File Manager the default VSAM editor on page 26.
__ 7	Enable WebSphere® MQ support. See Enabling WebSphere MQ support on page 29.
__ 8	Customize to use COBOL copybooks. See Customizing for processing COBOL copybooks on page 30.
__ 9	Customize to use PL/I include books. See Customizing for processing PL/I include books on page 34.
__ 10	Customize to use HLASM copybooks. See Customizing for processing HLASM copybooks on page 34.
__ 11	Customize for DFSORT™ to improve performance. See Customizing to use DFSORT to improve File Manager performance on page 35.
__ 12	Bind Db2® for OAM functions. See Binding Db2 to use File Manager object access method (OAM) functions on page 37.
__ 13	Authorize File Manager. See Planning for running File Manager with or without APF-authorization on page 17 and Running File Manager with APF-authorization on page 39.
__ 14	Change the default options. See Changing the default options on page 41.

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

	Description
__ 15	Change the batch JCL skeleton. See Changing the JCL skeleton for batch mode on page 43.
__ 16	Customize the batch return codes. See Customizing File Manager batch return codes on page 45.
__ 17	Provide a user I/O exit. See Customizing File Manager to use an I/O exit on page 120.
__ 18	Change the print and display translation tables. See Changing the print and display translation tables on page 48.
__ 19	Change the ASCII translation tables. See Changing the ASCII translation tables on page 49.
__ 20	Customize the File Manager security environment. See Customizing the File Manager security environment on page 51.
__ 21	Decide how to customize the File Manager audit facility. See Alternatives for controlling File Manager Base auditing on page 18.
__ 22	Customize File Manager for national languages. See Customizing File Manager for national languages on page 98.
__ 23	Customize File Manager to use library management system libraries. See Customizing File Manager to use library management system libraries on page 113.

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: File Manager target and distribution libraries on page 14.

If you have installed a Japanese component, the Japanese target and distribution libraries will be those listed in Table 3: File Manager Japanese target and distribution libraries on page 15. If you have installed a Korean component, the Korean target and distribution libraries will be those listed in Table 4: File Manager Korean target and distribution libraries on page 15.

Table 2. File Manager target and distribution libraries

Target library	Distribution library	Usage
FMN.SFMNMOD1	FMN.AFMNMOD1	Load modules
FMN.SFMNMOD2	FMN.AFMNMOD2	FM/CICS-specific load modules
FMN.SFMNMODA	FMN.AFMNMODA	FM/CICS-specific load modules that must be APF-authorized

Table 2. File Manager target and distribution libraries (continued)

Target library	Distribution library	Usage
		Eclipse plug-in specific load modules that must be APF-authorized
FMN.SFMNSAM1	FMN.AFMNSAM1	Sample jobs and source code
FMN.SFMNPENU	FMN.AFMNPENU	English panels
FMN.SFMNMENU	FMN.AFMNMENU	English messages
FMN.SFMNSLIB	FMN.AFMNSLIB	Skeletons
FMN.SFMNTENU	FMN.AFMNTENU	English tables
FMN.SFMNMAC1	FMN.AFMNMAC1	Macros
FMN.SFMNEXEC	FMN.AFMNEXEC	Execs
FMN.SFMNCLIB	FMN.AFMNCLIB	CLISTs
FMN.SFMNDBRM	FMN.AFMNDBRM	DBRMs

Table 3. File Manager Japanese target and distribution libraries

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

Table 4. File Manager Korean target and distribution libraries

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

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.

! **Important:** If you implement LIBDEFs to allocate the File Manager load libraries for a TSO user, and that user also has access to File Manager load libraries without having to issue LIBDEFs, the File Manager load libraries must be the same (identical version and maintenance levels).

Example

The File Manager libraries are allocated in a user's TSO logon procedure. The ISPF main menu is customized to allow FM/Db2 as an option. Selecting this option runs an exec based on FMNINIT to allocate File Manager libraries using LIBDEFs.

The File Manager libraries allocated in the TSO logon procedure are different to the File Manager libraries allocated in the exec using LIBDEFs.

In this situation problems will arise if the user has FM/Db2 running on one ISPF logical session, and File Manager base component or FM/IMS running on a second ISPF logical session.

Explanation

Most File Manager modules are reentrant. In an ISPF split screen environment each File Manager module will be loaded only once by the first ISPF session that requires it. If the user starts another ISPF session and runs File Manager, any modules needed by File Manager will only be loaded if the module in not already loaded. This can result in unpredictable results - in terms of the module loaded - depending on whether File Manager Base or File Manager IMS™ is started before a File Manager Db2® session, or vice versa.

To make File Manager readily available from ISPF, configure your ISPF environment as described in Customizing the operating environment for File Manager on page 20.

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 Customizing File Manager to use library management system libraries on page 113.

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, make the following change 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 56.
- 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 58.
- Users can perform catalog actions that require APF-authorization, as described in *z/OS DFSMS Access Method Services for Catalogs*.

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 79 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.

Alternatives for controlling File Manager Base auditing

File Manager auditing is an optional facility. There is no requirement to implement it and File Manager works if File Manager auditing is not implemented. You should consider:

- Whether user access to data sets and other resources using File Manager requires auditing.
- The information that File Manager audit log records can provide.
- The information that File Manager audit log records cannot provide, and possible alternatives to obtaining that information.
- If you do decide to use File Manager auditing, how you will handle any issues associated with large audit log data sets, or additional SMF records.
- How you will use the information provided by File Manager audit log records.

If your site requires a record of a user's read access to data sets, an external security product such as RACF® can be configured to log access by some or all users, and may be a better alternative.

File Manager audit of read access to data sets does not write audit log records for every record processed, rather the name of the data set and how many records were processed are written to the audit log.

File Manager audit of changes to data sets typically writes two log records, a before and after image of the record that was changed. If you intend to log update changes to data sets that are subject to heavy update activity you need to consider the performance impact of writing many audit log records, also the size of any audit log data sets that may be produced.

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. This auditing only applies to changes made by means of the File Manager editor.

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 user IDs.
- 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 implement SAF-rule controlled auditing you need to decide how File Manager auditing will be enabled. This is described in more detail in Customizing the File Manager audit facility for Base component on page 81. There are two alternatives: one requires an enabling SAF rule and the presence of a member in SYS1.PARMLIB, the other requires an enabling SAF rule but has no requirement for a member in SYS1.PARMLIB. The use of a member in SYS1.PARMLIB provides additional facilities compared with the alternative that does not require the use of SYS1.PARMLIB. The additional facilities are documented in File Manager options specified in PARMLIB members on page 532.

When you have determined the appropriate type of auditing for your installation, follow the instructions in Customizing the File Manager audit facility for Base component on page 81.

The auditing method you choose also applies to your RDz and Eclipse plug-in users if you are using the ADFzCC 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.

Enable/Register File Manager

Product enablement in IFAPRDxx

If you purchased File Manager as part of product code 5655-AC6 IBM Application Delivery Foundation for z/OS®, include an entry in the IFAPRDxx parmlib member as follows:

```
PRODUCT OWNER('IBM CORP')
NAME('IBM APP DLIV FND')
ID(5655-AC6)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('FILE-MANAGER')
STATE(ENABLED)
```

Sample registration parmlib entry FMNWIFAG, which contains the above statements, is provided in your *hlq*.SFMNSAM1 data set.

Alternatively and optionally, if you purchased File Manager separately (NOT as part of a bundle), you can include an entry for File Manager in the IFAPRDxx parmlib member as follows:

```
PRODUCT OWNER(’IBM CORP’)
NAME(’IBM FILE MANAGER’)
ID(5655-Q42)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME(’IBM-FM’)
STATE(ENABLED)
```

Sample registration parmlib entry FMNWIFA, which contains the above statements, is provided in your *hlq*.SFMNSAM1 data set.

After the IFAPRDxx parmlib member is updated, it can be activated dynamically (until the next IPL) using the following console command:

```
SET PROD=xx
```

Additional File Manager IFAPRDxx processing

If a product above is not defined in IFAPRDxx, when File Manager is first invoked it will register during initialization as *“IBM File Manager for z/OS”*, product code 5655-Q42.

If a product above is defined with STATE(DISABLED) or STATE(NOTDEFINED), the product will not be selected for registration.

To prevent File Manager from running, use the following IFAPRDxx entry:

```
PRODUCT OWNER(’IBM CORP’)
NAME(’IBM FILE MANAGER’)
ID(5655-Q42)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME(’IBM-FM’)
STATE(DISABLED)
```

If you change how a product is purchased, for example if you upgrade from using a stand-alone version to the IBM Application Delivery Foundation for z/OS® product bundle, you must explicitly remove the existing product definition from the in-storage tables kept by z/OS® when activating the new definition. Follow this scenario to do this dynamically (without IPL):

1. In IFAPRDxx, define the new product as described above, and update the old product with STATE(DISABLED).
2. Activate the update using the following operator command:

   ```
   SET PROD=xx
   ```
3. You can now safely remove the old product definition from IFAPRDxx.

IBM® advises against defining IFAPRDxx entries that have NAME(*) or ID(*) fields, as this could result in unintended product registrations. If a match is found on an entry defined with NAME(*) and ID(*) with STATE(ENABLED), *"IBM File Manager for z/OS"* will register as *"IBM Application Delivery Foundation for z/OS®"* (product code 5655-AC6).

Modifying the ISPF environment

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

TSO region size

When using File Manager from ISPF, it is recommended you use at least a 4 MB logon region size (in conjunction with the 32 MB of "extended" storage that TSO generally provides) to provide enough 24-bit storage for various functions to operate effectively. A suggested minimum TSO region size is 40 MB.

This starting value might need to be increased when performing intensive virtual storage functions, such as:

- The use of templates with many fields or layouts.
- In-memory editing of large files.

Preparing File Manager to run with LIBDEFs

Copy the member FMNELIBD from *hlq*.SFMNSAM1 to a library that is available in the SYSPROC DD statement for your TSO/ISPF users. Modify the exec as follows.

LNG = 'ENU'

Specify JPN for Japanese, KOR for Korean.

hlq = 'FMN'

Specifying this will result in names in the form *hlq*.SFMN*xxxx* if `sfx = ''`

optl = ''

Specify a value if there is a separate installation options library.

sfx = "

Specifying this will result in names in the form *hlq*.SFMN*xxxx.sfx*

hlq2 = "

LIBDEF fixtest library first in the form *hlq2*.SFMN*xxxx*

sfx2 = "

LIBDEF fixtest library first in the form *hlq2*.SFMN*xxxx.sfx2*

Notes:

1. For a standard installation you will only need to set `hlq =` to the installation value. Specify `optl = data set name` if you have customized versions of the installation options module in a different library to *hlq*.SFMNMOD1.
2. *hlq2* and *sfx2* are there to support BETA library testing.

Invocation of File Manager using LIBDEFs

FMNELIBD is a multi-purpose REXX exec to provide LIBDEF invocation of File Manager features and functions. This eliminates the need for various user-coded EXECs that perform LIBDEFs before invoking File Manager features or functions. The exec is run as follows.

Value

Description

FM

Invoke the File Manager primary options menu.

F1

Invoke the FM/IMS primary options menu.

F2

Invoke the FM/Db2 primary options menu.

FND

Invoke File Manager enhanced search.

LST

Invoke File Manager enhanced reference lists.

FUN

Invoke supported File Manager functions with data set names to populate the entry panels. See FUN parameter syntax on page 23.

DEF

Perform LIBDEFs for File Manager libraries from a REXX that invokes File Manager functions in keyword mode.

DROP

Clear LIBDEFs setup from FMNELIBD DEF command.

```
FMNELIBD FUN function 'dsn1(mbr1)' 'dsn2(mbr2)' 'dsn3(mbr3)' 'dsn4(mbr4)'
```

FUN parameter syntax

If you specify FUN then the following syntax applies.

function must be one of the following supported functions.

Value

Description

CLM

Compare load module

DSB

Data set browse

DSC

Data set copy

DSE

Data set edit

DSG

Data set create

DSM

Data set compare

DSV

Data set view

PBK

Print and view copybook or template

SCS

Catalog services

TPIMP

Template import

TPEXP

Template export

TPED

Template edit

TPUP

Template update

VLM

View load module

'dsn1(mbr1)' is the input data set or path name.

'dsn2(mbr2)' is the input template (specify '-' to bypass).

'dsn3(mbr3)' is the output data set or path name.

'dsn4(mbr4)' is the output template.

Example 1. Invoking File Manager, FM/IMS, and FM/Db2 primary options from a selection panel

```
#)BODY CMD(ZCMD) ...
9       IBM                     Products IBM program development products
10      SCLM                    SW Configuration Library Manager
11      Workplace               ISPF Object/Action Workplace
FM      File Manager            File Manager for z/OS                Ⓝⓔⓦ
F1      FM/IMS                  File Manager/IMS for z/OS            Ⓝⓔⓦ
F2      FM/DB2                  File Manager/DB2 for z/OS            Ⓝⓔⓦ
#
)PROC
#
&ZSEL = TRANS( TRUNC (&ZCMD,'.')
#
9,'PANEL(ISRDIIS) ADDPOP'
10,'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
11,'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
FM,'CMD(FMNELIBD FM)'   /* File Manager */                            Ⓝⓔⓦ
F1,'CMD(FMNELIBD F1)'   /* FM/IMS       */                            Ⓝⓔⓦ
F2,'CMD(FMNELIBD F2)'   /* FM/DB2       */                            Ⓝⓔⓦ
```

Example 2. ISPF command table with LIBDEF invocations

```
     Verb      T  Action
____ FM        2  SELECT CMD(%FMNELIBD FM &ZPARM)       <= File Manager Base
____ F1        2  SELECT CMD(%FMNELIBD F1 &ZPARM)       <= FM/IMS
____ F2        2  SELECT CMD(%FMNELIBD F2 &ZPARM)       <= FM/DB2
____ ELIST     2  SELECT CMD(%FMNELIBD LST &ZPARM)      <= Enhanced reference list
____ EFIND     2  SELECT CMD(%FMNELIBD FND &ZPARM)      <= Enhanced search
```

Example 3. Using REXX to invoke the VLM function (view load module)

This example will run File Manager option 3.10.1 and populate the input data set field with 'FMN.LOAD'

```
/* REXX */
call FMNELIBD FUN VLM 'FMN.LOAD'
```

Example 4. Using REXX to invoke the DSE function (data set edit)

This example will run File Manager option 2 and populate the input data set field and template fields.

```
/* REXX */
call FMNELIBD FUN DSE 'FMN.FMDATA'  'FMN.TEMPLATE(SAMPLE)'
```

Example 5. REXX load module report using the DEF and DROP parameters

```
/* REXX - VIEW LOAD MODULE - XML REPORT    */
/*                                         */
CALL FMNELIBD DEF
ADDRESS TSO
  "ALLOC FI(SYSPRINT) DUMMY REUSE"
  'ALLOC FI(FMNXMLO) NEW SP(1,5) TRACKS,
        LRECL(1024) BLKSIZE(32720) RECFM( V B )'
ADDRESS ISPEXEC
  "SELECT CMD(FILEMGR $VLM DSNIN='FMN.LOAD'",
      "MEMBER=TURBO1,FUNCTION=PRINT,DATEFORM=YYMMDD,XML=YES)"
ADDRESS TSO
  'EXECIO * DISKR FMNXMLO (STEM XMLO. FINIS'
  IF RC = 0 THEN DO
    DO I = 1 TO XMLO.0
      SAY XMLO.I
    END
  END
  "FREE FI(FMNXMLO)"
  "FREE FI(SYSPRINT)"
CALL FMNELIBD DROP
EXIT 0;
```

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 this figure:

Figure 7. Adding File Manager to the ISPF Primary Option Menu panel

```
#
)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
#
```

To invoke File Manager with LIBDEFs, see Example 1. Invoking File Manager, FM/IMS, and FM/Db2 primary options from a selection panel on page 24.

For 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
- 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 entering FM on the command line.

Verb	FM
Action	SELECT PANEL(FMNSTASK) OPT(&ZPARM) SCRNAME(FILEMGR) SUSPEND NEWAPPL(FMN)
Description	File Manager for z/OS®

To invoke File Manager with LIBDEFs, see Example 2. ISPF command table with LIBDEF invocations on page 24.

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

Avoiding conflicts between FM commands and other applications

If you encounter conflicts between File Manager commands and the same commands defined in ISPF command tables for other products, one solution is create an application command table that is specific to File Manager. Add an entry for the command that is used in the other application to the File Manager application command table and specify the PASSTHRU option.

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: VSAM-related keywords and values on page 27:

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*.

Invoking File Manager from ISPF 3.4 or a data set list

You can invoke a cut-down File Manager primary options menu against any data set on an ISPF 3.4 or data set list by invoking the supplied exec FMNR as a line command. Any subsequent entry panel navigated to via the cut-down menu has the primary data set name populated with the corresponding data set name from the ISPF 3.4 or data set list.

Note: This assumes the hlq.SFMNEXEC library has been added to SYSEXEC for the ISPF session.

Implementing File Manager edit models in ISPF Edit

Configure your ISPF environment to enable access to File Manager edit models from an ISPF Edit session.

To implement the File Manager ISPF edit models, perform the following steps:

- Ensure that FMN.SFMNPENU is concatenated to ISPPLIB.
- Ensure that FMN.SFMNSLIB is concatenated to ISPSLIB.
- Modify the ISPF panel ISREMCLS to include a new model class named JCL.

  A sample of the updates required can be found in FMN.SFMNSAM1(FMNEMCLS).
- Modify the ISPF panels ISREMEXC and ISREMRXC to include an option for processing File Manager external REXX functions from the EXEC and REXX model panels.

  A sample of the updates required for ISPEMEXC can be found in FMN.SFMNSAM1(FMNEMEXC) and the updates required for ISPEMRXC can be found in FMN.SFMNSAM1(FMNEMRXC).

Setting up CICS® access for File Manager base component and Eclipse plug-in

To access a CICS® resource from the File Manager plug-in or from the File Manager Base functions, you must perform each of the steps shown here:

1. **Enable CICS® Interregion communications**

   Specify the system initialization parameter IRCSTRT=YES, or ensure IRC is started dynamically with an OPEN IRC command.
2. **Define connections to CICS®**
   Copy member FMNCCONN from FMN.SFMNSAM1 into your JCL library and customize the JCL as described in the member. Ensure the group is defined and included in the start up list for the CICS® regions to be connected to.

   Note: If you change the transaction name of FMNX, ensure you customize the File Manager base and server options modules to specify the new transaction name via the EXCITRAN parameter (see Step 5 on page 28).
3. **Install File Manager for CICS®**

   FM/CICS must be installed at the level where EXCI support was introduced in all relevant CICS® regions.
4. **Ensure the ADFzCC server is installed and configured for File Manager**

   ADFz Common Components must be installed at V1R8, and the ADFzCC server configured for File Manager and started.
5. **Option module changes**

   If the transaction ID FMNX was changed in Step 2 on page 28, then FMN0POPT must be modified to specify the new transaction ID by means of the EXCITRAN keyword.
6. **EXCI Load library**

   The CICS® external interface library *cicshlq*.SDFHEXCI must be made available to TSO, the batch job, or the ADFzCC server, either in the linklist, or as a part of the STEPLIB in the TSO procedure, or batch JCL, or as a part of the FMNLIB in the configuration file used by the ADFzCC server startup procedure. If you are using a STEPLIB, then you should customize the batch job submission skeleton FMNFTEXC to include the library.
7. **FMNCICS DD**

Copy member FMNCAPPL from FMN.SFMNSAM1 into your own PDS(E) data set that has the attributes of record format FB and logical record length 80. Edit this member to provide a list of CICS® regions that are accessible to File Manager. This is used to support generic queries from the File Manager ISPF interface and the File Manager Eclipse plug-in.

The layout of the member is:

1 - 8

CICS® VTAM® applid

10 - 72

CICS® region description

All users must have read access to the data set and the FMNCICS DD should be defined to the TSO procedure and the ADFzCC server File Manager configuration.

8. **Security**

   All CICS® communications are performed under the security defined for the user running the File Manager function. A user must have update access to the FACILITY class profile DFHAPPL.user ID to be authorized to use the EXCI Call interface. You must define the FACILITY class profiles shown here:

   ```
   RDEFINE FACILITY (DFHAPPL.userid) UACC(NONE)
   ```

   Where *userid* is either a generic name or a specific user name. We recommend you use an asterisk (*) and control access to the facility class using the permissions.
   Users requiring CICS® access should be given UPDATE access to the FACILITY class as shown here:

   ```
   PERMIT DFHAPPL.userid  CLASS(FACILITY) ID(USERID)   +
                 ACCESS(UPDATE)
   ```

For more information about setting up the CICS® external interface, refer to the *CICS® Transaction Server for z/OS® CICS® External Interfaces Guide*.

Enabling File Manager to work with certain products

You can enable File Manager to work with IBM® languages and 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

To access WebSphere® MQ queue managers and queue contents on the local z/OS® system where File Manager is running, the WebSphere® MQ libraries must be made available to File Manager and users must be granted read and update access.

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 using SYSTEM.COMMAND.REPLY.MODEL 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 user ID and a ".*" string. The MQ manager being used will then return the full dynamic name to be used. You can change this high level qualifier to suit your installation's MQ security rules.

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

File Manager users who must access IBM® MQ resources require the following access:

- Update access to the dynamic queues *MQREPHLQ.userid*.*
- Update access to the SYSTEM.COMMAND.INPUT queue
- Read access to the SYSTEM.COMMAND.REPLY.MODEL queue

Requesting users also require relevant permissions to access specific IBM® MQ resources. These resources might include these permissions:

- To connect to queue managers
- To read or alter queue manager or queue attributes
- To create and delete resources
- To browse or destructively get queue messages

To determine what permissions are required, see *“Setting up security on z/OS”* in the IBM® MQ security documentation in IBM Knowledge Center.

For FM/CICS using the ADFzCC server, they must be added to the configuration data for the FMNLIB concatenation in the ADFz Common Components CONFIG DD concatenation. For more information about the ADFzCC server and configuration data, see Customizing the ADFz Common Components server on page 558.

Customizing for processing COBOL copybooks

To use the 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 have 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, (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, or when 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 on page 31. For more information about using the File Manager internal COBOL compiler, see Using the File Manager COBOL compiler on page 33.

Use the SHOWCOB command to check whether File Manager is using its internal COBOL compiler or another compiler.

- In batch, submit a File Manager batch job with this control statement:

  ```
  $$FILEM SHOWCOB
  ```

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

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

  If File Manager is using its own COBOL compiler, the job output will show:

  ```
  FMN0023I The File Manager supplied COBOL Compiler will be used
  ```
- Under ISPF, enter SHOWCOB on any File Manager command line. If a licensed, supported COBOL compiler is available to File Manager, this message is displayed: `Using Licensed COBOL`. If File Manager is using its own COBOL compiler, this message is displayed: `Using FM COBOL Compiler`

Note that your COBOL compiler must be both licensed and supported. 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 attempt to compile copybooks will fail.

Using a supported COBOL compiler

If a supported COBOL compiler is available and will be used by File Manager, review the following sections for customization tasks you might need to perform.

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 43 and Tailoring the job control skeletons on page 277.

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        STGOPT
```

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 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

File Manager uses its internal COBOL compiler if one of these conditions applies:

- A licensed, supported COBOL compiler is not available.
- An `//FMNCOB DD DUMMY` has been defined (as described below).
- A user has a minimum of Read access to the SAF FACILITY class profile FILEM.COBOL.INTERNAL, as described in Controlling the use of the COBOL compiler on page 61.

If File Manager is to use its internal COBOL compiler, 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.

Forcing the use of the File Manager COBOL compiler

The File Manager COBOL compiler load modules are shipped in FMN.SFMNMOD1.

To force File Manager to use this compiler, you take one of these actions:

- Define a FACILITY class profile as described in Controlling the use of the COBOL compiler on page 61.
- Allocate a DD statement

  ```
  //FMNCOB DD DUMMY
  ```

  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 use the File Manager COBOL compiler in the File Manager batch JCL skeleton, FMNFTEXC, and the FM/IMS batch JCL skeleton, FMN1FTEX. 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 43 and Tailoring the job control skeletons on page 277.

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, FMN0DOPT, 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

File Manager provides an internal PL/I compiler that is used regardless of any external PL/I compiler previously installed.

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.

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.

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

File Manager can use DFSORT to improve the performance of the Data Set Copy and Data Set Print functions under ISPF and in batch.

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. File Manager accesses DFSORT using the aliases ICEDFSRT and ICEDFSRB.

File Manager will use DFSORT provided it can find DFSORT and verify that it 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.

The following topics describe how you can customize File Manager to use DFSORT 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 43.

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 43.

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

Use the SHOWSORT command to check if File Manager is using DFSORT.

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.

In batch, submit a File Manager batch job with this control statement:

```
$$FILEM SHOWSORT
```

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

```
SORT debugging is on
```

If DFSORT is not available, the job output will show:

```
DFSORT not available
```

Under ISPF, enter SHOWSORT on any File Manager command line. If DFSORT is available to File Manager, this message is displayed: `SORT debugging is on`. If DFSORT is not available, this message is displayed: `DFSORT not available`.

Summary

Once you have determined 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 the DFSORT COPY function because File Manager does this automatically. File Manager does not use the DFSORT 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 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. Plan 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

Disabling Websphere MQ feature by system name

You can disable the usage of File Manager Websphere MQ support by using the following FACILITY class profile if you have Security Server RACF® 1.9 (or later) or an equivalent security product.

FMNMQ.DISABLE.*system_name*

Where:

system_name

The value specified on the SYSNAME parameter in the IEASYMxx parmlib member. The value for any given system is displayed in the System ID field on the File Manager primary option menu or alternatively the ISPF primary option panel ISR@PRIM.

If the profile has been defined and a user has READ or higher access to the profile name then all Websphere MQ functionality within File Manager will be disabled for that user.

Example 1. Define a profile to disable all users on the system that is named FMPROD except *myuser*.

```
RDEF FACILITY FMNMQ.DISABLE.FMPROD AUDIT(NONE) +
   UACC(READ) OWNER(ownerid)
PE FMNMQ.DISABLE.FMPROD +
   CLASS(FACILITY) ID(myuser) ACC(NONE)
```

Example 2. Define a profile to enable all users on the system that is named FMPROD except *myuser*.

```
RDEF FACILITY FMNMQ.DISABLE.FMPROD AUDIT(NONE) +
   UACC(NONE) OWNER(ownerid)
PE FMNMQ.DISABLE.FMPROD +
   CLASS(FACILITY) ID(myuser) ACC(READ)
```

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.

Note:

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 115.

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 you must ensure that the module FMNSMF is APF-authorized. You also need to provide an SMF number, either in the appropriate option's module or parmlib member. See Customizing File Manager to write audit records to SMF on page 79.

The current File Manager function is terminated when auditing to SMF fails.

Message *“Unable to link to FMNSMF. TSOLNK RC=20 (Dec) REASON CODE=56 (Dec)”* is diagnostic of an APF authorization failure. The most common causes are:

- The library containing FMNSMF is not APF authorized.
- The DDNAME to which the library containing FMNSMF is allocated includes other libraries and at least one of those libraries is not APF-authorized.

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 by using the usermod FMNUMODP. For a description of the options and the values you can specify, see File Manager options on page 383.

The default macro statements are in FMN.SFMNSAM1(FMN0POPT).

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: If you do not want to use SMP/E you can use the sample job FMN0POPH to assemble FMN0POPT .

Setting the default national language

If you installed the Japanese or the Korean component (or you have provided locally translated messages and panels), you can change the default national language used in the File Manager ISPF and batch interfaces.

The language used by File Manager under ISPF depends on the language setting for your ISPF session.

The language used by File Manager for batch processing depends on the LANGUAGE option in FMN0POPT. See Table 16: Keyword values for the LANGUAGE option on page 102 for the value to specify for LANGUAGE. See LANGUAGE on page 405 for more information about the LANGUAGE option.

FM/CICS also uses the LANGUAGE option in FMN0POPT to determine the language for ISPF panels and messages. If you have installed the FM/CICS Japanese or Korean components, change the setting to that language. If you have provided FM/CICS messages and panels in another language, change the setting to that language.

Setting up File Manager to use a template repository

A template repository is a VSAM file that contains cross-references of data sets, CICS® files (including CICS® TS and CICS® TD queues), path names, and WebSphere MQ queue names to their respective templates.

There are three ways to identify the template repository to a File Manager session. The methods are evaluated in the following order and the first specification is used for the rest of the session.

- Specifying the repository name in the respective FMNxPARM PARMLIB member.
- Allocating the TPREPOS ddname to the data set name
- Customizing FMN4POPT. See Customizing miscellaneous options in FMN4POPT on page 383.

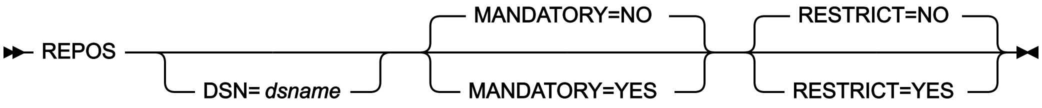
Specifying the repository data sets using PARMLIB options

You can specify one or more repository data sets in the PARMLIB member. Use this method to connect users to a repository data set name and to mandate templates or views for specified users.

- Use the <U *userid1*, *userid2*, … *useridn*> tag to connect specified users to a repository data set name.
- Use the <X *xfacilit.profile*> tag to apply the rule to all users who have read access or more to a specified XFACILIT profile.

Enter the repository definitions in order from the most specific to the most general. The first definition that matches the current user will be used.

Figure 8. Syntax



Where:

DSN

The name of the repository data set.

MANDATORY

Specify YES if the repository identifies resources for the user that have templates or views that are to be used to anonymize or scramble the data.

Note: To enable dynamic scrambling through the TESTMASK command the resource entry must be marked as mandatory.

RESTRICT

Specify YES to restrict usage to the resources specified in the repository.

Example 1

Enter this PARMLIB definition to set up a mandatory repository for three users and default repository for the rest.

```
<U FRED, BOBBYB, BIILBO>
REPOS DSN=MASK.CREDIT.CARDS.REPOS,MANDATORY=YES
</U>
REPOS DSN=DEFAULT.REPOS
```

Example 2

Enter this PARMLIB definition to set up a mandatory repository for any user having read access or more to the MY.MANDATED.USERS XFACILIT class profile.

```
<X MY.MANDATED.USERS>
REPOS DSN=MASK.CREDIT.CARDS.REPOS,MANDATORY=YES
</X>
REPOS DSN=DEFAULT.REPOS
```

Allocating TPREPOS

The ddname TPREPOS can be allocated to a TSO/ISPF session or specified as a DD card to a batch job.

JCL:

```
//DD TPREPOS DISP=SHR,DSN=hlq.TEMPLATE.REPOSTRY
```

TSO command:

```
'ALLOC FI(TPREPOS) DA('hlq.TEMPLATE.REPOSTRY') SHR REUSE'
```

You can also add a DD specification to the CONFIG=FM for File Manager tasks that require the ADFzCC server:

TPREPOS=FMN.TEMPLATE.REPOSTRY

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. These statements are commented out in the distributed version of the skeleton. They 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
```

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 this statement to the name of your File Manager load library.

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

File Manager can process WebSphere® MQ queues in both online and batch scenarios for the common utilities; create, copy, compare and print. To interface with WebSphere®, 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 the following statement, which appears as a comment in FMNFTEXC.

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

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 Using COBOL compiler options with File Manager on page 32 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 its 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
```

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 33.
If you want to produce a compile listing of copybooks processed by File Manager functions, in case of compile errors, remove the * to uncomment the fourth statement:

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

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 35.

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 Customizing File Manager to use library management system libraries on page 113.

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. 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's value to 16. The value 16 is reserved for use by File Manager. Figure 9: Default

FMN0RETC macro statements supplied with File Manager on page 47 shows the default set of FMN0RETC macro statements.

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.

Figure 9. Default FMN0RETC macro statements supplied with File Manager

```
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                      X
               INVDATA=8          Invalid data                         X
               SYNCERR=8          Key Synchronization error
         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
```

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.

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.

Note:

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. If you customize one return code for a given function, 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 setting the ABENDCC installation option to 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 384, and also the ABENDCC option of the SET command, described in section *"SET (Set Processing Options)"* in *File Manager User's Guide and Reference* .

Changing the print and display translation tables

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

File Manager provides default translation tables for English (FMNTRTBS), German (FMNTRDEU), Japanese (FMNTRJPN), and Korean (FMNTRKOR).

For other languages you will need to provide and maintain your own print translation table and display translation table. See Customizing File Manager for national languages on page 98 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, FMN.SFMNSAM1(FMNTRTBS).

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.

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 41.
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.

To change the tables in languages other than English, see Customizing File Manager for national languages on page 98.

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 sample member FMNASCII to change the ASCII translation tables.

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.

Chapter 4. Customizing the File Manager security environment

File Manager provides security for system-oriented functions through ®either RACF® (or an equivalent security product) or the FMNSECUR exit.

If RACF® 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 on page 51 for information on defining profiles. 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 RACF® 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 61.

Note: The FMNSECUR module will not be used (even if present) if RACF® 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).

The following sections contain examples of setting up facility classes for File Manager using RACF®. For more information about RACF® resource profiles, see *z/OS Security Server RACF Command Language Reference*.

Note: If you are using an equivalent security product, refer to that product's documentation for information on how to define and use 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 58

FILEM.DISK.FULLPACK

See Controlling fullpack access to DASD volumes on page 56

For more information about these groups, see Table 8: File Manager function to profile name cross-reference on page 72.

Controlling access

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 user ID 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.

To achieve this File Manager makes RACROUTE calls, with STATUS=ACCESS, to the CICS® SAF FACILITY profiles. 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 other security products such as ACF2 are used, an S047 abend may be issued in response to the above RACROUTE request. In this case you should consult the documentation for your security product and make changes accordingly.

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 Customizing File Manager CICS Component on page 323, and also the *File Manager User's Guide and Reference for CICS*.

Protecting update functions

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 IMS subsystems and FM/IMS functions access control facility on page 281.

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 419. For more information about the protected functions, see Unprotected functions and profile names for protected functions on page 69. For a list of functions that are protected by this method, see Table 6: File Manager unprotected functions on page 70, Customizing to protect update functions in FM/Db2 on page 180, and Customizing to protect update functions in FM/CICS on page 342.

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

Examples of giving or denying access

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 user ID *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 user ID *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: File Manager function to profile name cross-reference on page 72 (in the form FILEM.*group.name*) is used.
- If no profile name as shown in Table 8: File Manager function to profile name cross-reference on page 72 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: File Manager functions protected by FILEM.OTHER.ALL on page 71.

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 10: Access to File Manager functions on page 55.

Figure 10. Access to File Manager functions

```
┌──────────────────────────┐
│File Manager function (fc)│
└────────────┬─────────────┘
             ↓
┌────────────────────────────────────────────┐
│FACILITY(FILEM.FUNCTION.fc) access          │
├──────┬────────┬──────┬───────┬─────────────┤
│ALTER │ UPDATE │ READ │ NONE  │ not defined │
└──┬───┴────┬───┴──┬───┴───┬───┴──────┬──────┘
   │        │      │       │          │
   └────────┴──────┘  ┌────┘          │
   ↓                  ↓               ↓
┌────────┐  ┌────────┐  ┌──────────────────────────────┐
│Accepted│  │Rejected│  │FACILITY(FILEM.group.name) access│
└────────┘  └────────┘  ├──────┬────────┬──────┬───────┤
                        │ALTER │ UPDATE │ READ │ NONE  │
                        └──┬───┴────┬───┴──┬───┴───┬───┘
                           │        │      │       │
                           └────────┴──────┘       │
                                    ↓              ↓
                               ┌────────┐     ┌────────┐
                               │Accepted│     │Rejected│
                               └────────┘     └────────┘
```

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).

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: File Manager functions protected by FILEM.OTHER.ALL on page 71.

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 11: Fullpack processing for disk read functions on page 57 and Figure 12: Fullpack processing for disk update functions on page 58 show how this works for disk read and disk update functions.

Figure 11. Fullpack processing for disk read functions

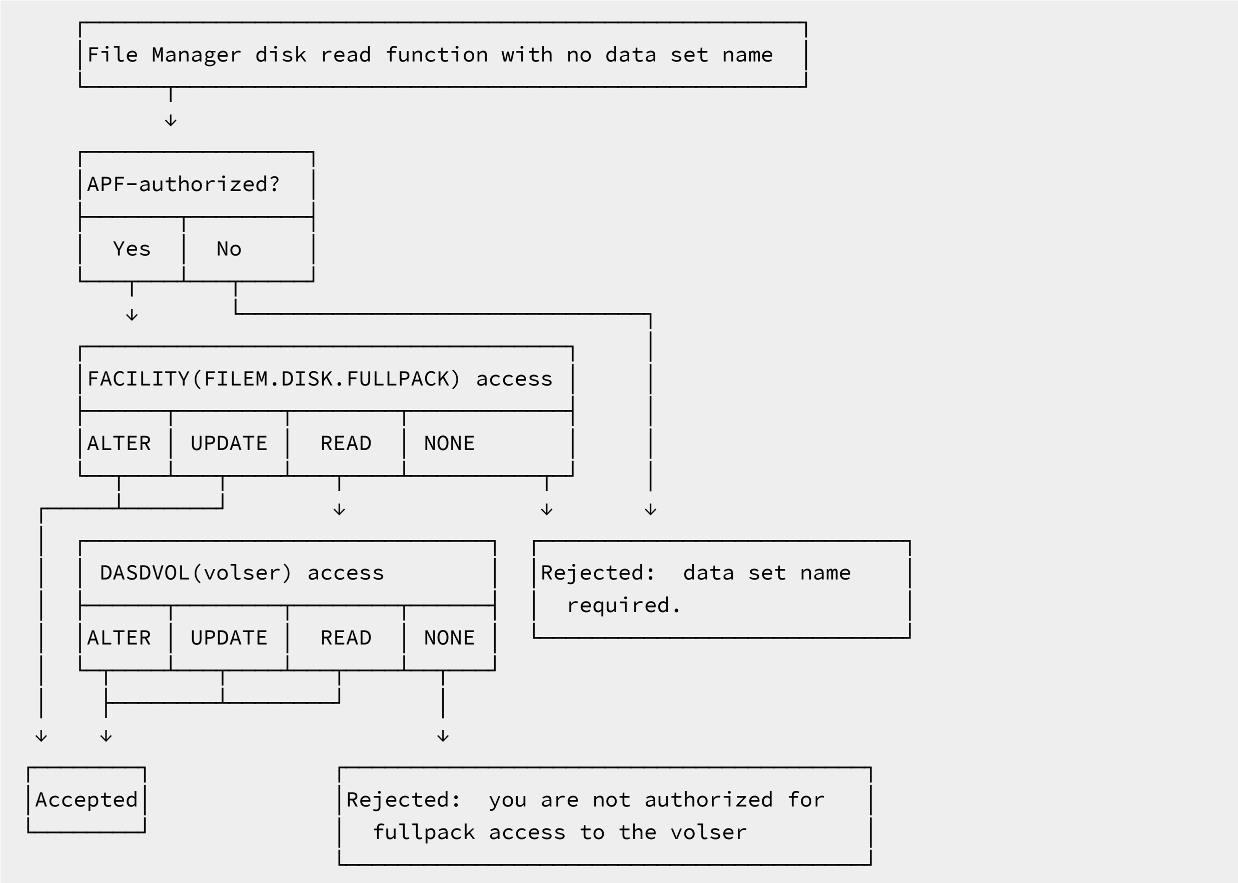
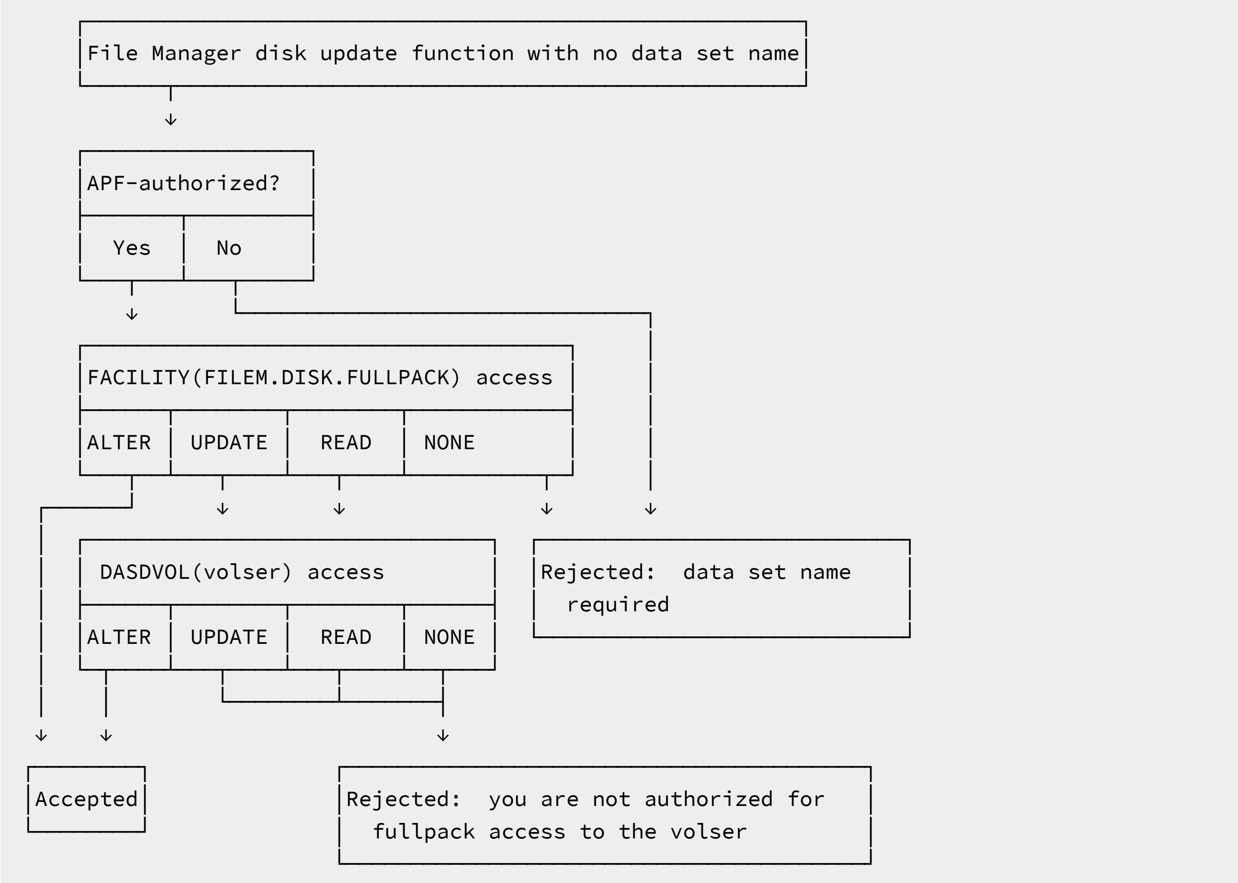

File Manager disk read function with no data set name
APF-authorized?
Yes
No
FACILITY(FILEM.DISK.FULLPACK) access
ALTER
UPDATE
READ
NONE
DASDVOL(volser) access
ALTER
UPDATE
READ
NONE
Rejected: data set name required.
Accepted
Rejected: you are not authorized for fullpack access to the volser

Figure 12. Fullpack processing for disk update functions


File Manager disk update function with no data set name
APF-authorized?
Yes
No
FACILITY(FILEM.DISK.FULLPACK) access
ALTER
UPDATE
READ
NONE
DASDVOL(volser) access
ALTER
UPDATE
READ
NONE
Rejected: data set name required
Accepted
Rejected: you are not authorized for fullpack access to the volser

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:

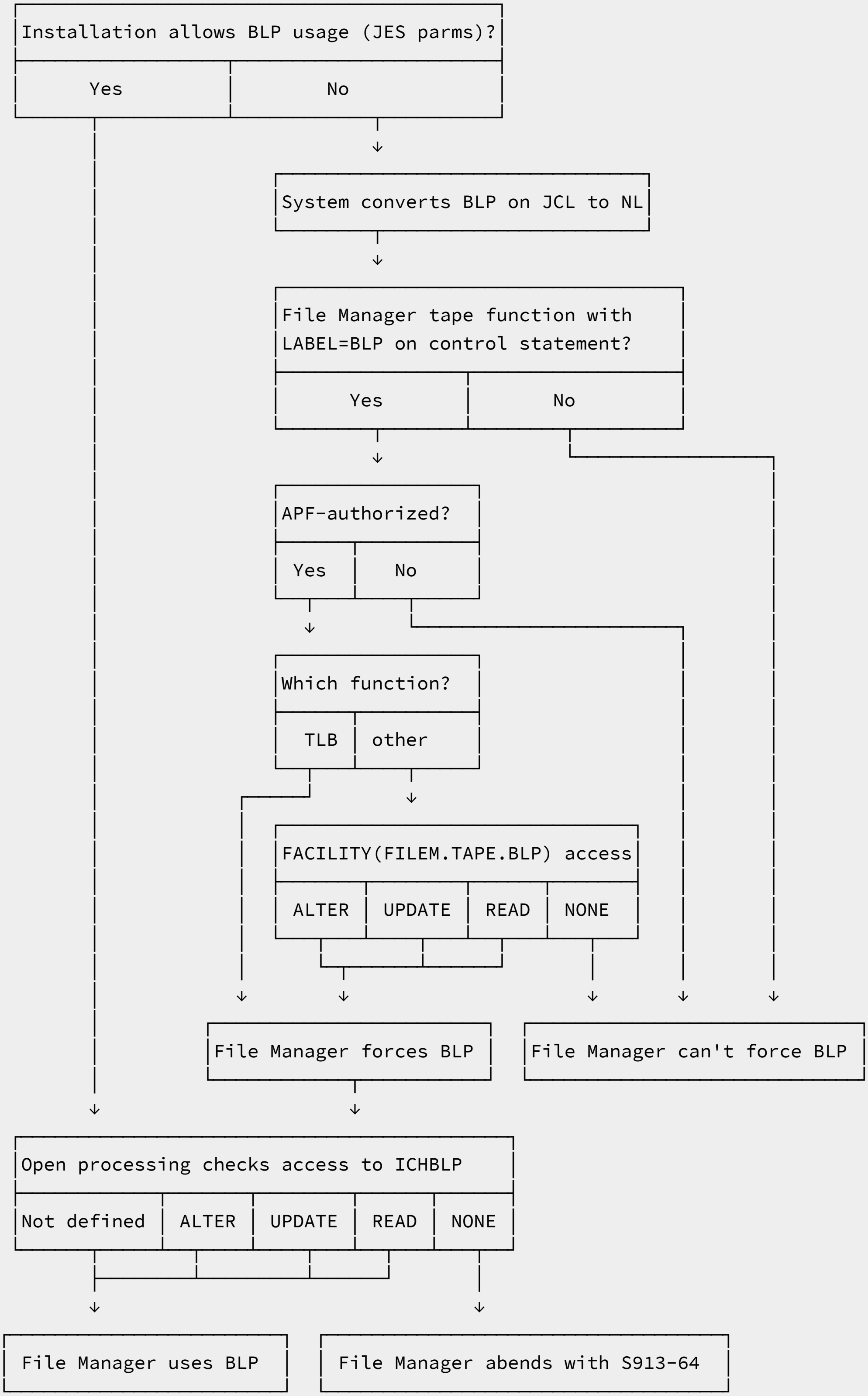
- File Manager must be running APF-authorized.
- For any function other than TLB, the user must have access to FILEM.TAPE.BLP.
- If ICHBLP is defined to your security product, the user must have READ access or greater to it.

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 13: BLP processing by File Manager on page 60.

Figure 13. BLP processing by File Manager


Installation allows BLP usage (JES parms)?
Yes
No
System converts BLP on JCL to NL
File Manager tape function with LABEL=BLP on control statement?
Yes
No
APF-authorized?
Yes
No
Which function?
TLB
other
FACILITY(FILEM.TAPE.BLP) access
ALTER
UPDATE
READ
NONE
File Manager forces BLP
File Manager can't force BLP
Open processing checks access to ICHBLP
Not defined
ALTER
UPDATE
READ
NONE
File Manager uses BLP
File Manager abends with S913-64

Controlling the use of the COBOL compiler

A FACILITY CLASS profile is provided to allow you to control access to the File Manager internal COBOL compiler:

```
FILEM.COBOL.INTERNAL
```

If a user ID running File Manager has READ access (or above) to this profile, then the File Manager internal compiler will always be used for compiling templates from COBOL copybooks. If this profile does not exist or if the user ID has ACCESS NONE to this profile, the usual rules determining which compiler is used are honored. See Using a supported COBOL compiler on page 31 and Using the File Manager COBOL compiler on page 33.

To achieve this, File Manager makes RACROUTE calls, with STATUS=ACCESS, to the FACILITY profiles.

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.

When a non-RACF security product such as ACF2 is used, an abend S047 may be issued in response to the above RACROUTE request. In this case consult the product documentation and make changes accordingly.

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 69.

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 58

FILEM.DISK.FULLPACK

See Controlling fullpack access to DASD volumes on page 56

For more information about these groups, see Table 8: File Manager function to profile name cross-reference on page 72.

- Give or deny some users (or all users) access to an individual File Manager function.

Controlling access

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 user ID 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 Customizing File Manager CICS Component on page 323, and also the *File Manager User's Guide and Reference for CICS*.

Protecting update functions

Three facility groups are 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 IMS subsystems and FM/IMS functions access control facility on page 281.)

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 419. For more information about the protected functions, see Unprotected functions and profile names for protected functions on page 69. For a list of functions that are protected by this method, see Table 6: File Manager unprotected functions on page 70, Customizing to protect update functions in FM/Db2 on page 180, and Customizing to protect update functions in FM/CICS on page 342.

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 65.

Examples of giving or denying access

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: File Manager function to profile name cross-reference on page 72 (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:

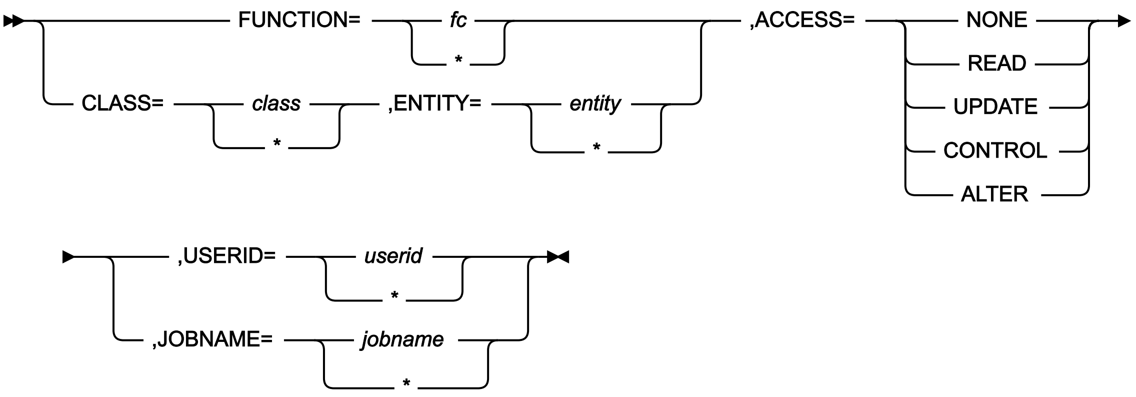
  ```
  FMNS CLASS=DASDVOL,
       ENTITY=volser,
       ACCESS=READ,
       USERID=userid
  ```

  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

Figure 14. 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: File Manager function to profile name cross-reference on page 72 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 64.

USERID

Specify either a user ID 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 user ID or job name. In batch mode, the job name is passed to FMNSECUR and the user ID is also passed to FMNSECUR if it is available. FMNSECUR reads through the list of FMNS macros until a match is found for the user ID 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: File Manager function to profile name cross-reference on page 72.
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 user ID.
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: File Manager unprotected functions on page 70, Table 7: File Manager functions protected by FILEM.OTHER.ALL on page 71, and Table 8: File Manager function to profile name cross-reference on page 72, show File Manager functions with their descriptions and (where applicable) the equivalent File Manager panel you can use under ISPF.

Table 6: File Manager unprotected functions on page 70 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: File Manager unprotected functions on page 70 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: File Manager unprotected functions on page 70 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 342.

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: File Manager unprotected functions on page 70 are protected, and Table 8: File Manager function to profile name cross-reference on page 72 shows the profile name used in the FACILITY class to protect a specific function.

All protected functions not listed in Table 8: File Manager function to profile name cross-reference on page 72 are protected by the FILEM.OTHER.ALL profile.

In particular, the File Manager functions listed in Table 7: File Manager functions protected by FILEM.OTHER.ALL on page 71 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 51. In particular, if you use a non-IBM security product, see the warning in Controlling access to File Manager functions with SAF on page 55.

Note: In Table 8: File Manager function to profile name cross-reference on page 72, 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	Equiva-lentpanel	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	

Table 6. File Manager unprotected functions (continued)

Function	Equiva-lentpanel	Description	Protect with SEC=YES
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	Equiva-lentpanel	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
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	Equiva-lentpanel	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
TLT	4.2.3	Tape to Labeled Tape	FILEM.TAPE.DUPLICATE

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

Function	Equiva-lentpanel	Description	Profile
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: File Manager unprotected functions on page 70 or this table.	FILEM.OTHER.ALL

Setting up the security environment for IBM® MQ

File Manager provides security features to secure access to MQ resources when using File Manager functions.

These features work in conjunction with IBM® MQ security, not as a replacement. If access is not restricted by File Manager, it may still be restricted by IBM® MQ security. Similarly, if access is not restricted by IBM® MQ security, it may still be restricted by File Manager security. File Manager security features for MQ are only applicable when accessing MQ resources using File Manager.

File Manager security for MQ is applicable to the user attempting to use a File Manager function that accesses an MQ resource. By default, File Manager does not secure access to MQ resources.

Activating security for a queue manager

To secure a queue manager, you must define a security resource indicating that security is required for a nominated queue manager on a given sysplex.

The security resource takes the form FMNMQ.SECURITY.*sysplex*.*qmgr* and must be defined to the FACILITY class. For example:

```
RDEFINE FACILITY FMNMQ.SECURITY.SYSPLEXD.CSQ1 UACC(READ)
```

Granting READ access to a user indicates that File Manager security is applicable to that user for the nominated queue manager on the nominated sysplex. If no access is granted, security is not active. When security is active, users must be granted further access to resources to access the queue manager’s attributes and queues. If security is not active for a queue manager, File Manager permissions related to the queue manager do not apply.

Securing queue manager resources

When security is active for a queue manager, a user cannot access any of a queue manager’s resources unless the user has at least READ access to a security resource of the form FMNMQ.*sysplex*.*qmgr* defined to the XFACILIT class.

For example:

```
RDEFINE XFACILIT FMNMQ.SYSPLEXD.CSQ1 UACC(NONE)
PERMIT FMNMQ.SYSPLEXD.CSQ1 CLASS(XFACILIT) ID(JOHND) ACCESS(READ)
```

Granting READ access to a user allows the user to list the queue manager’s attributes, its queues, and its queue’s attributes. To alter a queue manager’s attributes, a user must have ALTER authority. For example, the following authority also allows a user to modify the queue manager’s queue attributes, delete existing queues, and create new queues.

```
PERMIT FMNMQ.SYSPLEXD.CSQ1 CLASS(XFACILIT) ID(JOHND) ACCESS(ALTER) ALTER
```

Securing queue messages

When security is active for a queue manager, a user cannot access a queue’s messages unless the user has at least READ access to a security resource of the form FMNMQ.*sysplex*.*qmgr*.*queue* defined to the XFACILIT class.

For example:

```
RDEFINE XFACILIT FMNMQ.SYSPLEXD.CSQ1.* UACC(NONE)
PERMIT FMNMQ.SYSPLEXD.CSQ1.* CLASS(XFACILIT) ID(JOHND) ACCESS(READ)
```

Granting READ access to a user allows the user to browse messages on the queue. To edit, insert, delete, or destructively get messages on a queue, a user must have at least UPDATE authority. For example, the following authority also allows a user to reset or clear a queue’s messages:

```
PERMIT FMNMQ.SYSPLEXD.CSQ1.* CLASS(XFACILIT) ID(JOHND) ACCESS(UPDATE)
```

UPDATE authority also allows a user to reset or clear a queue’s messages.

Securing File Manager commands

When security is active for a queue manager, a user must have appropriate access to target MQ resources pertinent to the File Manager command being executed.

There is a range of commands that can affect IBM® MQ resources. In each case the following security resource definitions and permissions are required:

- When a command reads a queue manager's attributes or its queue's attributes, the requesting user must have READ authority to resource FMNMQ.*sysplex*.*qmgr* in the XFACILIT class for the queue manager being read.
- When a command alters a queue manager's attributes or defines a queue, the requesting user must have ALTER authority to resource FMNMQ.*sysplex*.*qmgr* in the XFACILIT class for the queue manager being modified.
- When a command reads message data, the requesting user must have READ authority to resource FMNMQ.*sysplex*.*qmgr*.*queue* in the XFACILIT class for the queue being read.
- When a command updates message data, the requesting user must have UPDATE authority to resource FMNMQ.*sysplex*.*qmgr*.*queue* in the XFACILIT class for the queue being updated.

Securing message context

File Manager uses MQ security controls to edit messages and message context.

When a File Manager session is started, File Manager checks if the user has CONTROL access to the *qmgr*.CONTEXT.*queue* resource in the MQADMIN class. For some External Security Managers, this check might require certain permissions. When a message is updated, the MQ message context is preserved if the user has CONTROL access to the *qmgr*.CONTEXT.*queue* resource in the MQADMIN class. If not, the message context is replaced with a default message context in accordance with the MQ normal operation.

MQ security examples

These examples illustrate techniques for controlling access to MQ resources in File Manager.

Activate security for all queue managers on a sysplex

To activate security for all users and for all queue managers and their resources on a sysplex, define the following resource:

```
RDEFINE FACILITY FMNMQ.SECURITY.sysplex.* UACC(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue managers. Without further permissions, this will disable all File Manager MQ function for all queue managers on the sysplex.

Activate security for a specific queue manager on a sysplex

To activate security for all users of a specific queue manager and its resources, define the following resource:

```
RDEFINE FACILITY FMNMQ.SECURITY.sysplex.qmgr UACC(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, and *qmgr* is the name of the relevant queue manager on the sysplex. Without further permissions, this will disable all File Manager MQ function for the nominated queue manager.

Permit all users read access to a queue manager's resources

When security is active, to permit all users read access to all resources of a specific queue manager, define the following resources:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(READ)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(READ)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, and *qmgr* is the name of the relevant queue manager on the sysplex. This will allow all users to list the queue manager’s attributes, its queues, and its queue’s attributes, as well as read messages from its queues.

Permit all but a single user read access to a queue manager’s resources

To permit all but a single user read access to all resources of a specific queue manager, define the following resources and permissions:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(READ)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(READ)
PERMIT FMNMQ.sysplex.qmgr CLASS(XFACILIT) id(userid) ACCESS(NONE)
PERMIT FMNMQ.sysplex.qmgr.* CLASS(XFACILIT) id(userid) ACCESS(NONE)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, *qmgr* is the name of the relevant queue manager on the sysplex, and *userid* represents the restricted user. This will allow all but the nominated user to list the queue manager's attributes, its queues, and its queue’s attributes, as well as read messages from its queues.

Permit a single user read access to a queue manager's resources

To permit only a single user read access to all resources of a specific queue manager, define the following resources and permissions:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(NONE)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(NONE)
PERMIT FMNMQ.sysplex.qmgr CLASS(XFACILIT) id(userid) ACCESS(READ)
PERMIT FMNMQ.sysplex.qmgr.* CLASS(XFACILIT) id(userid) ACCESS(READ)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, *qmgr* is the name of the relevant queue manager on the sysplex, and *userid* represents the permitted user. This will allow only the nominated user to list the queue manager’s attributes, its queues, and its queue’s attributes, as well as read messages from its queues.

Permit all users update access to all message data

When security is active, to permit all users update access to all queue's message data of a specific queue manager, define the following resources:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(READ)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(UPDATE)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, and *qmgr* is the name of the relevant queue manager on the sysplex. This will allow all users to list the queue manager’s attributes, its queues, and its queue’s attributes, as well as create, read, update, and delete messages from its queues.

Permit all but a single user update access to all message data

To permit all but a single user update access to all message data of a specific queue manager, define the following resources and permissions:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(READ)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(UPDATE)
PERMIT FMNMQ.sysplex.qmgr CLASS(XFACILIT) id(userid) ACCESS(NONE|READ)
PERMIT FMNMQ.sysplex.qmgr.* CLASS(XFACILIT) id(userid) ACCESS(NONE|READ)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, *qmgr* is the name of the relevant queue manager on the sysplex, and *userid* represents the restricted user. This will allow all but the nominated user to list the queue manager’s attributes, its queues, and its queue's attributes, as well as create, read, update, and delete messages from its queues.

Permit a single user update access to all message data

To permit only a single user update access to all message data of a specific queue manager, define the following resources and permissions:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(NONE)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(NONE)
PERMIT FMNMQ.sysplex.qmgr CLASS(XFACILIT) id(userid) ACCESS(READ)
PERMIT FMNMQ.sysplex.qmgr.* CLASS(XFACILIT) id(userid) ACCESS(UPDATE)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, *qmgr* is the name of the relevant queue manager on the sysplex, and *userid* represents the permitted user. This will allow only the nominated user to list the queue manager’s attributes, its queues, and its queue’s attributes, as well as create, read, update, and delete messages from its queues.

Permit all users alter access to a queue manager’s resources

When security is active, to permit all users alter access to all resources of a specific queue manager, define the following resources:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(ALTER)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(UPDATE)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, and *qmgr* is the name of the relevant queue manager on the sysplex. This will allow all users to list and update the queue manager’s attributes, its queues, and its queue’s attributes, as well as create, read, update, and delete messages from its queues.

Permit all but a single user alter access to a queue manager's resources

To permit all but a single user alter access to all resources of a specific queue manager, define the following resources and permissions:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(ALTER)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(UPDATE)
```

```
PERMIT FMNMQ.sysplex.qmgr CLASS(XFACILIT) id(userid) ACCESS(NONE|READ)
PERMIT FMNMQ.sysplex.qmgr.* CLASS(XFACILIT) id(userid) ACCESS(NONE|READ)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, *qmgr* is the name of the relevant queue manager on the sysplex, and *userid* represents the restricted user. This will allow all but the nominated user to list and update the queue manager’s attributes, its queues, and its queue’s attributes, as well as create, read, update, and delete messages from its queues.

Permit a single user alter access to a queue manager's resources

To permit only a single user alter access to all resources of a specific queue manager, define the following resources and permissions:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr UACC(NONE)
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.* UACC(NONE)
PERMIT FMNMQ.sysplex.qmgr CLASS(XFACILIT) id(userid) ACCESS(ALTER)
PERMIT FMNMQ.sysplex.qmgr.* CLASS(XFACILIT) id(userid) ACCESS(UPDATE)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, *qmgr* is the name of the relevant queue manager on the sysplex, and *userid* represents the permitted user. This will allow only the nominated user to list and update the queue manager's attributes, its queues, and its queue’s attributes, as well as create, read, update, and delete messages from its queues.

Permit a single user access to a subset of queues

To permit a single user access to a subset of resources of a queue manager, define the following resources and permissions:

```
RDEFINE XFACILIT FMNMQ.sysplex.qmgr.DEV.* UACC(NONE)
PERMIT FMNMQ.sysplex.qmgr.DEV.* CLASS(XFACILIT) id(userid) ACCESS(READ|UPDATE)
SETROPTS RACLIST(XFACILIT) REFRESH
```

Where *sysplex* is the name of the sysplex hosting the queue manager, *qmgr* is the name of the relevant queue manager on the sysplex, and *userid* represents the permitted user. This will allow the nominated user relevant access to queues prefixed "DEV.".

Chapter 5. Customizing File Manager to write audit records to SMF

You can use the audit logging facilities of File Manager to create an audit trail of editing and other activities against data sets, Db2® objects, IMS™ databases and CICS® resources. For more information, refer to:

- Customizing the File Manager audit facility for Base component on page 81
- Customizing the audit facility for FM/Db2 on page 186
- Customizing the File Manager audit facility for IMS component on page 295
- Customizing the File Manager audit facility for CICS component on page 349

Audit records can be written to an audit log data set, or to SMF, or both (dual logging). See the topics referred to above for information about restrictions that apply to dual logging.

If you intend to write audit log records to SMF for any component of File Manager, complete the steps below.

Using System Management Facilities (SMF) for audit logging

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

- Select an SMF record number between 128 and 255 for the audit log records, and include it in your SMF parmlib member SMFPRMxx.
- Specify this SMF record number in one of these locations:
    - The FMN0POPI macro for the appropriate FMNxPOPT module. (See Note 1).
    - The FMNxPARM member in SYS1.PARMLIB, or other library in the logical PARMLIB concatenation. (See Note 2).
- 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 39.
- Add the load module FMNSMF to the AUTHTSF list in member IKJTSOxx 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 is done.

Note:

1. Each File Manager component has a customization module:

    FMN0POPT

    For File Manager Base component

FMN1POPT

For FM/IMS

FMN2POPT

For FM/Db2

FMN3POPT

For FM/CICS

All the customization modules include an FMN0POPI macro specification, which is described in File Manager options on page 383. The SMF record number is specified using the SMFNO parameter of the FMN0POPI macro. See SMFNO on page 421. You should specify the SMF record number in the FMNxPOPT member when you are using FMNxPOPT controlled auditing, or SAF-controlled auditing without the use of a member in SYS1.PARMLIB.

2. Auditing for each File Manager component can be controlled using a member in SYS1.PARMLIB, or other library in the logical PARMLIB concatenation. The member names for each component are:

FMN0PARM	For File Manager Base component
FMN1PARM	For FM/IMS
FMN2PARM	For FM/Db2
FMN3PARM	For FM/CICS

Specify the SMF record number in the FMNxPARM member when you are using SAF-controlled auditing and a member in SYS1.PARMLIB.

If File Manager was previously customized to use the BPX1SMF service, consider removing access to SAF FACILITY class profile BPX.SMF from all users to be audited.

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 the *“Audit trail”* option from the Utilities 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. The sample job can be used to extract audit log records for all File Manager components (Base, FM/Db2, FM/IMS, and FM/CICS). The logon ID used to run the sample job must have read access to the SYS1.MANx data sets to run successfully.

Chapter 6. Customizing the File Manager audit facility for Base component

File Manager can write audit log records to either SMF or an audit log data set.

If auditing is not required:

- Set AUDITLOG=NO in the FMN0POPI macro specification of the FMN0POPT module. See AUDITLOG on page 388 for more information.
- Set SMFNO=0 in the FMN0POPI macro specification of the FMN0POPT module. See SMFNO on page 421 for more information.
- Skip the customization described in the rest of this chapter.

File Manager provides two different methods (FMN0POPT-controlled and SAF-controlled auditing) for controlling whether audit records are written for File Manager Base component. These are described in detail in Alternatives for controlling File Manager Base auditing on page 18.

You should determine which of the two methods is appropriate for your site's requirements.

Use the checklist to determine the customization required for the File Manager audit facility.

Table 9. Checklist for audit customization, File Manager Base component. This table lists choices and decisions.

Audit customization choice	Decision (Yes|No|Not applicable)
1. Control auditing using the FMN0POPT options module	
2. Control auditing using SAF rules and a member in SYS1.PARMLIB	
3. Control auditing using SAF rules, without any changes to SYS1.PARMLIB	
4. Audit records are to be written to a data set	
5. Audit records are to be written to SMF	

For choices 1 to 3, you should answer YES for one choice only. Mark the other two choices as 'Not applicable'.

If you answer YES for choice 1, you can answer YES for one of choices 4 and 5. Mark the other choice as 'Not applicable'.

If you answer YES for choices 2 or 3, you can answer YES for one or both of choices 4 and 5. If you answer YES for both choices you are implementing dual-logging, which is only available with SAF-controlled auditing.

Table 10. Customization steps for audit customization choices. This table lists choices and related actions.

Customization choice	Sections to complete
1. Control auditing using the FMN0POPT options module	• FMN0POPT-controlled auditing on page 82
2. Control auditing using SAF rules and a member in SYS1.PARMLIB	• SAF-controlled auditing for File Manager Base component on page 84 • Implementing SAF-rule controlled auditing on page 87
3. Control auditing using SAF rules, without any changes to SYS1.PARMLIB	• SAF-controlled auditing for File Manager Base component on page 84 • Implementing SAF-rule controlled auditing on page 87
4. Audit records are to be written to a data set	• Audit data set configuration on page 82
5. Audit records are to be written to SMF	• Customizing File Manager to write audit records to SMF on page 79

FMN0POPT-controlled auditing

To implement FMN0POPT-controlled auditing you need to set the AUDITLOG option in the FMN0POPI macro of the FMN0POPT module.

For information about changing the options in FMN0POPT see Changing the default options on page 41.

For information about the options see File Manager options on page 383.

- If you want to produce an audit trail, specify AUDITLOG=YES in FMN0POPI macro.
- If you want to produce an audit trail and report on the changes made at the 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*.

Audit data set configuration

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 386 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

Any 1-8 character constant that is valid in the context of a data set name.

userid

The user ID creating the data set.

Dyymmdd

The date of the activity.

Thhmmss

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.

Set the AUDITHLQ parameter in the FMN0POPI macro for the FMN0POPT to the required value, based on the above information and your site's requirements.

You can print the information in a File Manager Base audit data set using the File Manager Print Audit Trail utility. To do this select option 3.12 from the File Manager Primary Option Menu.

SAF-controlled auditing for File Manager Base component

There are two methods for implementing SAF-controlled auditing for File Manager Base component. These are:

1. Control auditing for File Manager Base component using an enabling SAF Facility class rule and a member in SYS1.PARMLIB.

   To use this method complete the customization described in SAF-controlled auditing using SYS1.PARMLIB on page 84.
2. Control auditing for File Manager Base component using an enabling SAF Facility class rule, without any changes to SYS1.PARMLIB.

   To use this method complete the customization described in SAF-controlled auditing without SYS1.PARMLIB on page 86.

SAF-controlled auditing using SYS1.PARMLIB

You need to define an enabling SAF facility profile as described below:

Define SAF facility profile

```
FILEM.PARMLIB.BASE
```

and ensure all File Manager Base users to be audited have at least read access to that facility. See the example below:

Example

User PROD1 to have SAF-rule controlled auditing using SYS1.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)
```

Add member FMN0PARM to SYS1.PARMLIB (or any other library in the logical parmlib concatenation). See Defining the FMN0PARM member on page 85.

Once the above SAF rule is defined and activated, auditing for File Manager Base component users is controlled by the FMAUDIT parameter in the FMN0PARM member. See File Manager options specified in FMN0PARM on page 532 for more information. If audit log records are to be written to SMF, the SMF record number is specified as an FMAUDIT parameter option. See FMAUDIT on page 532, and SMF_NO on page 533.

Note: File Manager does not start if a user has read access to the above facility and the FMN0PARM member does not exist in the logical parmlib concatenation.

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.

Defining the FMN0PARM member

If auditing is to be controlled from parmlib, then member FMN0PARM must be defined in SYS1.PARMLIB (or any other library in the logical parmlib concatenation).

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 a FMSECRTY statement. This option is not used at present, and can be either omitted, or commented out. It has no effect.

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).

Note:

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.
   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
2. Add member FMN0PARM to the new library, specifying the appropriate FMAUDIT parameter.
3. 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 on page 86. If the FMN0PARM member specifies any include statements (see Facilities for customizing the FMN0PARM definitions on page 534), all of the included members must also reside in the same library.

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 File Manager options specified in FMN0PARM on page 532.

SAF-controlled auditing without SYS1.PARMLIB

You need to define an enabling SAF facility profile as described below:

Define SAF facility profile

```
FILEM.SAFAUDIT.BASE
```

and ensure that all File Manager Base users to be audited have at least read access to that facility. See the example below.

Example

User PROD2 to have SAF-rule controlled auditing without using SYS1.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)
```

If you use this method and intend to write audit records to SMF, the required SMF number is specified in the FMN0POPT module. See Customizing File Manager to write audit records to SMF on page 79 for more information.

When a security product other than RACF® is in use

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® 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, an S047 abend may be issued in response to the above RACROUTE request. These users should consult the product documentation and make changes accordingly.

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 87 for more information.
2. Write the relevant SAF rules. See the examples in SAF rule examples on page 92.
3. Activate SAF auditing for a specific logon using the chosen method of activating SAF-controlled auditing. See SAF-controlled auditing for File Manager Base component on page 84.
4. Using the selected logon, test the configuration to ensure that auditing is occurring as required.
5. When testing is complete, activate SAF-controlled auditing for all File Manager users.

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 13: File Manager auditing FACILITY class resource names on page 95 and Table 14: File Manager auditing XFACILIT class resource names on page 95).

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 13: File Manager auditing FACILITY class resource names on page 95). 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 (online 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:

1. Step 1.
   - If auditing is being controlled by means of parmlib, the FMAUDIT specification of the FMN0PARM member is used as follows.

     The FMAUDIT specification setting in the FMN0PARM member (in SYS1.PARMLIB or any other library in the logical parmlib concatenation) 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 532 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 Customizing the File Manager audit facility for Base component on page 81.

     SAF_CTRL=YES

     SAF-rule controlled auditing is in effect. Processing continues to Step 2 on page 89.
   - If auditing is being controlled using the method which does not access the parmlib concatenation, the TSO logonid has READ access to the SAF FACILITY rule FILEM.SAFAUDIT.BASE for processing to continue to Step 2 on page 89.
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 13: File Manager auditing FACILITY class resource names on page 95; the various outcomes are summarized in Table 11: Determination of a user's ability to write audit log records on page 90.

   Table 11. 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

Notes:

a. Refers to the level of access the user has to SAF FACILITY rule 1 in Table 13: File Manager auditing FACILITY class resource names on page 95.
b. Refers to the level of access the user has to SAF FACILITY rule 2 in Table 13: File Manager auditing FACILITY class resource names on page 95.
c. Refers to the level of access the user has to SAF FACILITY rule 3 in Table 13: File Manager auditing FACILITY class resource names on page 95.
d. 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.

3. Step 3.

   Does the user have access to write audit records for the current function and data set?

   The XFACILIT resource names that are 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 12: File Manager function codes that can be audited using SAF on page 91 shows the function codes which are supported.

Table 12. 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.

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 13: File Manager auditing FACILITY class resource names on page 95 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 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 writes audit log records for functions which access resources. Table 12: File Manager function codes that can be audited using SAF on page 91 shows File Manager function 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 user IDs 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 user IDs 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 user IDs 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 user IDs 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 FACILITY and XFACILIT class resource names and details.

Table 13. File Manager auditing FACILITY class resource names

Rule Number	Resource Name	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 14. File Manager auditing XFACILIT class resource names

Resource Name[3]	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 (re-

Table 14. File Manager auditing XFACILIT class resource names

(continued)

Resource Name[3]	Purpose
	source) 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).

Note:

1. Use this option with caution. The size of the data set being accessed and the editing technique used will influence the number of read records logged, and this may affect the performance of File Manager.
2. When specifying the resource:
   - 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.
   - For CICS® resources other than files, the resource name is in the format:

     ```
     queuetype:cicsapplid:resourcename
     ```

     Where:

     queuetype

     Queue type. Can be one of these values:

     TS

     For temporary storage queues.

     TD

     For internal transient data queues.

     External transient data queues are controlled using the associated data set name.

     cicsapplid

     The CICS® applid where the resource is defined.

     resourcename

     The name of the temporary storage queue or internal transient data queue.
   - For CICS® files and external transient data queues, specify only the data set name.
   - If accessing CICS® resources from File Manager base running under FM/CICS, omit the *cicsapplid* parameter from the resource name:

     ```
     queuetype:resourcename
     ```

Table 14. File Manager auditing XFACILIT class resource names

(continued)

Resource Name[3]	Purpose
3. You cannot define a SAF XFACILIT rule containing any of these: • Lowercase characters • Embedded spaces • Unprintable characters • Most special characters	

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, you will need to perform all of the steps listed in Table 15: Summary of steps for customizing File Manager for a national language on page 98.

Table 15. Summary of steps for customizing File Manager for a national language

Step	Description
__ 1	Set the LANGUAGE option for batch processing, if required. See Setting the default national language on page 41.
__ 2	Change the TERMTYPE option in FMN0POPT, if you are using a DBCS language. See TERMTYPE on page 423.
__ 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. See Changing the print and display translation tables for languages other than English on page 98.
__ 5	Translate the File Manager message text to your language. See Translating the message text on page 100.
__ 6	Provide a version of FMN0MENU for your language. See Providing a multicultural version of FM-N0MENU on page 100.
__ 7	Translate the File Manager ISPF messages to your language. See Translating the ISPF messages text on page 101.
__ 8	Translate the File Manager panels to your language. See Translating the panel text on page 101.

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

If you plan to use File Manager with a national language other than English, you might need to provide 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, which is distributed in FMN.SFMNMOD1. You can modify this table, if necessary, using the usermod FMNUMODG, which is distributed in FMN.SFMNSAM1.

A sample translation table is provided for Thai. This is FMNTRTHT, which is distributed in FMN.SFMNSAM1. To build the Thai translation table, copy this sample member to FMNTRTBS and use the usermod FMNUMODX.

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 41 for information on how to do this.
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 102.

To provide translated versions of the messages, you must provide a version of FMN0MENU for your language, as described in Providing a multicultural version of FMN0MENU on page 100, **and** provide translated versions of the appropriate members in FMN.SFMNMENU, as described in Translating the ISPF messages text on page 101.

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 multicultural 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 step 4.a on page 99 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 step 4.a on page 99.) 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 102.

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.)

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 4.a on page 99.) 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 102.

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 16: Keyword values for the LANGUAGE option on page 102. See Changing the default options on page 41 for information on how to do this.

Table 16. 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.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

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 Service Provider for z/OS® Connect

File Manager Service Provider for IBM® z/OS® Connect Enterprise Edition enables client applications to access IBM® Z data sources using z/OS® Connect.

Using z/OS® Connect tooling, customers can create their own RESTful APIs and access IBM® Z data sources through the service provider. Clients can read data sequentially, by relative record position, or by key.

Installation and customization of the service provider is optional. It is only necessary for customers who plan to use this feature to access IBM® Z data sources through z/OS® Connect.

Data sources

The supported data sources include:

- MVS data sets (including VSAM)
- USS files
- CICS® files (including VSAM and CICS® TS and CICS® TD queues)
- Websphere MQ queues
- Db2®
- IMS™

Prerequisites

File Manager Service Provider has the following prerequisites:

- File Manager for z/OS V14 (with APAR PI97118, and APAR PH08775 for IMS™ and Db2® support).

  The service provider is only available with File Manager for z/OS V14.1.2 and later.
- Application Delivery Foundation for z/OS Common Components V1.8 (with APAR PI97749, and APAR PH04064 for IMS™ and Db2® support).

  ADFz Common Components is provided with the File Manager product and must be installed and configured as a prerequisite. The ADFzCC server must be running for the service provider to perform its function.

  Refer to *ADFz Common Components Customization Guide and User Guide* for more information.
- IBM® z/OS® Connect Enterprise Edition V3.0 (5655-CE3) or later

  z/OS® Connect Enterprise Edition V3.0 or later must be installed and configured before installing and configuring the File Manager Service Provider. Refer to z/OS® Connect EE documentation in IBM® Knowledge Center for installation and configuration information.

Installing File Manager Service Provider for z/OS® Connect

The File Manager Service Provider feature is extracted to UNIX® System Services, then installed into z/OS® Connect and configured in ADFzCC server.

The File Manager Service Provider for z/OS® Connect feature is shipped as a binary archive in the SFMNSAM1 data set as member FMNZCEE.

Generic or specific mapping of data sources

z/OS® Connect Enterprise Edition provides tools to create APIs that use the File Manager Service Provider. During API creation, a service archive (SAR file) is associated with the API which identifies the target service provider.

File Manager service archives (fm*.sar) and the File Manager z/OS Connect Build Toolkit plug-in (com.hcl.zosconnect.buildtoolkit.filemanager.jar) are provided for this purpose.

fmService.sar, fmIMSService.sar, and fmDB2Service.sar are ready-to-use SAR files that can be associated with z/OS® Connect APIs during API creation. They provide a *generic* mapping of file data to a prescribed response format, that is, one format for all data sources. When using generic mappings, you do not need to create SAR files.

Alternatively, customers who require a response mapping *specific* to a data resource can create their own SAR files. To create and manage your own SAR files you must install the File Manager z/OS Connect Build Toolkit plug-in.

To understand if you will use the *specific* or *generic* response mapping methods, and therefore whether you need to install the Build Toolkit plug-in or the provided SAR files or both, refer to *"Methods for accessing IBM Z data sources"* in the *File Manager User's Guide and Reference*.

Installing File Manager Service Provider on z/OS®

The File Manager Service Provider feature is extracted to USS, then installed into z/OS Connect within WLP, and then configured in ADFzCC server.

The installation procedure is conducted in the UNIX® System Services (USS) environment and has the following environmental prerequisites:

- The shell environment must have a z/OS® Connect EE-compatible version of Java™ available on the environment PATH variable. You can check the PATH variable by using the UNIX® System Services (USS) echo $PATH command.
- The z/OS® Connect installation directory `<ZCON_INST_DIR>/v3r0/wlp/bin` must be in the PATH environment variable.

The File Manager Service Provider for z/OS® Connect feature is shipped as a binary archive in the SFMNSAM1 data set as member FMNZCEE. The archive contains the following files:

com.hcl.zosconnect.fm.provider.feature_n.n.n.n.esa	Feature archive
fmService.sar	Base service archive

fmIMSService.sar	IMS™ service archive
fmDB2Service.sar	Db2® service archive
filemanager.properties	Properties file
com.hcl.zosconnect.buildtoolkit.filemanager.jar	Build Toolkit plug-in

1. Extract the File Manager Service Provider for z/OS® Connect archive from the SFMNSAM1 data set using the USS pax command.

   ```
   pax -rf "//'<hlq>.SFMNSAM1(FMNZCEE)'"
   ```

   This command extracts the FMNZCEE archive into the current working directory.
2. (Optional) Edit the productInstall option in the properties file if you want to change the installation directory.

   ```
   com.ibm.websphere.productInstall=</your/preferred/directory>
   ```

   This is the location where the File Manager Service Provider is installed. The directory must be fully qualified. That is, it must begin with a forward slash (/). The default installation directory is /var/filemanager. If you elect to change the default directory, it is recommended that you choose one that does not already contain existing files or sub-directories. The productInstall directory is created if it does not already exist.
3. Copy the properties file to the z/OS® Connect `<WLP_USER_DIR>/v3r0/extensions` directory using the USS cp command.
   For example:

   ```
   cp filemanager.properties /var/zosconnect/v3r0/extensions
   ```
4. Copy the service archive files to the z/OS® Connect service directory.
   For example:

   ```
   cp fm*.sar
      /var/zosconnect/servers/<yourWLPServer>/resources/zosconnect/services
   ```

   Ensure that the service archive files are readable by the z/OS® Connect WLP started task user.
5. Install the File Manager Service Provider feature into z/OS® Connect using the installUtility command. For example:

   ```
   installUtility install
   com.hcl.zosconnect.fm.provider.feature_2.0.0.0.esa --to=filemanager
   ```

   Ensure that the ESA file version number matches the file name in the archive.
6. Edit the z/OS Connect WLP `server.xml` configuration.

   The server.xml requires the following changes:

   a. Add the File Manager feature to the featureManager stanza:

   ```
   <featureManager>
    <feature>filemanager:fmProvider-2.0</feature>
   </featureManager>
   ```

   b. Add the runport and maximum timeout configuration for the feature:

```
<FileManager_Connection id="default" runport="2800" max_timeout="1800" />
```

where runport represents the ADFzCC server port that is used to establish File Manager sessions, and max_timeout represents the maximum number of seconds for which a client can hold a File Manager session open without activity.

7. Change the ADFzCC server JCL to support z/OS Connect REST requests.

   In the IPVSRV CONFIG DD statement, add REST=YES after the CONFIG=FM stanza. For example:

   ```
   //CONFIG DD *
    CONFIG=FM
    REST=YES
    …
   /*
   ```
8. Restart or refresh the z/OS Connect WLP task and ADFzCC server task.

To verify a successful installation of the File Manager Service Provider, do the following steps.

1. Check the JOBLOG of the z/OS Connect WLP started task and verify that the CWWKF0012I message lists `fmProvider-2.0` as one of the installed features. For example:

   ```
   CWWKF0012I: The server installed the following features: ssl-1.0,
   jdbc-4.1, filemanager:fmProvider-2.0, …
   ```
2. Invoke the File Manager Service Provider manually, using an HTTP PUT request to `https://zceehost:zceeport/zosConnect/services/fmService?action=invoke` with the following payload:

   ```
   {
       "action" : "verify"
   }
   ```

   This can be done from a browser or a REST API tool such as POSTMAN. The HTTP verb must be PUT or POST, and a `Content-Type: application/json` header is required. A HTTP status code of 200 indicates that the File Manager Service Provider has been installed correctly.

If installation verification fails, check that all installation steps have been completed in the designated order. The *File Manager User's Guide and Reference* also provides a troubleshooting topic for the File Manager Service Provider for z/OS Connect.

Installing the service archive files

z/OS® Connect API creation tools run outside z/OS® in an Eclipse environment. If you want to use the generic File Manager service archive files (fmService.sar, fmIMSService.sar, or fmDB2Service.sar) you must make them available to the Eclipse environments in which they will run. The SAR files must be copied to the relevant Eclipse environments, or to a network folder where the relevant Eclipse environments can access them.

The service archive files reside in the USS directory where the pax archive was expanded. They must be copied in binary mode.

Installing the Build Toolkit plug-in

This topic explains how to install the File Manager Build Toolkit plug-in for use with the zconbt command.

z/OS® Connect provides a Build Toolkit (zconbt) for creating service archive (SAR) files. SAR files are used during z/OS® Connect API creation to associate APIs with service providers such as File Manager Service Provider.

The z/OS® Connect EE Build Toolkit is bundled with the z/OS® Connect host product. It provides the zconbt command, which runs in Windows, Linux, and z/OS® environments.

1. If you have not already done so, extract the z/OS® Connect Build Toolkit to the environment (Windows, Linux or z/OS®) where you intend to build your SAR files.
2. Copy the File Manager Build Toolkit plug-in to the Build Toolkit `/plugins` directory.
   For example:

   ```
   cp com.hcl.zosconnect.buildtoolkit.filemanager.jar /var/zconbtv3/plugins
   ```
3. Edit the `plugin.properties` file in the z/OS® Connect Build Toolkit `/lib` directory, and add an entry for the File Manager plug-in.
   For example, edit `/var/zconbtv3/lib/plugin.properties` and then add the following entry:

   ```
   filemanager=com.hcl.zosconnect.buildtoolkit.filemanager.FileManagerSarGenerator
   ```
4. You can now use the zconbt command to create SAR files for File Manager APIs.
   For example:

   ```
   cd /var/zconbtv3/bin
   zconbt.zos -p=yourSAR.properties -f=./yourSAR.sar
   ```
5. All service archives created using the File Manager Build Toolkit plug-in must be copied to the z/OS Connect `/zosconnect/services` directory.
   For example:

   ```
   cp yourService.sar /var/zosconnect/servers/<yourWLPServer>/resources/zosconnect/services
   ```

   Copied archives must be readable by the z/OS Connect WLP STC user.

Related information

The z/OS Connect EE build toolkit

Creating services with the build toolkit

Using the Build Toolkit plug-in

TLS for the Build Toolkit plug-in

During SAR creation, the File Manager Build Toolkit plug-in needs to connect to an instance of the ADFzCC server (IPVSRV).

If your IPVSRV server is configured to require clients to connect using transport layer security (TLS), you will need to install the trust chain of your server's digital certificate in the environment where the Build Toolkit runs.

The File Manager Build Toolkit plug-in uses the default trust store of the JVM environment in which it runs, for example: `$JAVA_HOME/jre/lib/security/cacerts`

If your default trust store does not already contain the Certificate Authority (CA) trust chain of your server certificate, you will need to export your CA certificate chain and import it into the default trust store. For example:

```
RACDCERT SITE EXPORT(LABEL('ADFzCC Server Certificate')) DSN(ADFZCC.CERT)
   FORMAT(CERTDER)
```

Note that there may be multiple certificates in the trust chain.

Once you have the CA certificates of your server certificate, you can import them into the JVM default trust store using your preferred tooling. For example, using Java™ keytool:

```
keytool -import -trustcacerts -keystore jre\lib\security\cacerts -storepass changeit
 -noprompt -alias ADFZCC -file C:\certs\adfzcc.cert
```

With the CA trust chain in place, the File Manager Build Toolkit plug-in will be able to establish a TLS connection with your IPVSRV server.

Maintaining File Manager Service Provider

If service is applied to the FMNZCEE member of the SFMNSAM1 data set, some maintenance steps are necessary.

Note: If the service provider is not already installed into the z/OS Connect environment, then maintenance is not applicable, and the installation steps should be followed instead.

The maintenance procedure is conducted in the UNIX® System Services (USS) environment and has the following environmental prerequisites:

- The shell environment must have a z/OS® Connect EE-compatible version of Java™ available on the environment PATH variable. You can check the PATH variable by using the UNIX® System Services (USS) echo $PATH command.
- The z/OS® Connect installation directory `<`*`ZCON_INST_DIR`*`>/v3r0/wlp/bin` must be in the PATH environment variable.

The File Manager Service Provider for z/OS® Connect feature is shipped as a binary archive in the SFMNSAM1 data set as member FMNZCEE. The archive contains the following files:

com.hcl.zosconnect.fm.provider.feature_n.n.n.n.esa	Feature archive
fmService.sar	Base service archive
fmIMSService.sar	IMS™ service archive
fmDB2Service.sar	Db2® service archive
filemanager.properties	Properties file
com.hcl.zosconnect.buildtoolkit.filemanager.jar	Build Toolkit plug-in

The following procedure includes stopping and restarting the z/OS® Connect WLP server. In high-availability environments this may be undesirable. For alternative procedures refer to the following topics:

- Stopping and starting z/OS Connect EE servers in an HA environment
- Management of z/OS Connect EE APIs, services, and API requesters in an HA environment

1. Extract the File Manager Service Provider for z/OS® Connect archive from the SFMNSAM1 data set using the USS pax command.

   ```
   pax -rf "//'<hlq>.SFMNSAM1(FMNZCEE)'"
   ```

   This command extracts the FMNZCEE archive into the current working directory.
2. Copy the service archive files to the z/OS® Connect service directory.

   For example:

   ```
   cp fm*.sar
      /var/zosconnect/servers/<yourWLPServer>/resources/zosconnect/services
   ```

   Ensure that the service archive files are readable by the z/OS® Connect WLP started task user.
3. (Optional) If you use the zconbt command to create SAR files, copy the File Manager Build Toolkit plug-in to the Build Toolkit `/plugins` directory.

   For example:

   ```
   cp com.hcl.zosconnect.buildtoolkit.filemanager.jar /var/zconbtv3/plugins
   ```
4. Check APAR HOLDDATA before continuing to the next step in case there are additional steps that need to occur at this point.
5. Stop the z/OS Connect WLP server.
6. Uninstall the current File Manager Service Provider using the installUtility command.

   For example:

   ```
   installUtility uninstall filemanager:fmProvider-2.0
   ```
7. Reinstall the File Manager Service Provider feature into z/OS Connect using the installUtility command.

   For example:

   ```
   installUtility install
   com.hcl.zosconnect.fm.provider.feature_2.0.0.0.esa --to=filemanager
   ```

   Note that the .esa filename may vary by version.
8. Restart the z/OS Connect WLP server.

Uninstalling File Manager Service Provider

Uninstall File Manager Service Provider from z/OS® Connect using the installUtility command.

The uninstall procedure is conducted in the UNIX® System Services (USS) environment and has the following environmental prerequisites:

- The shell environment must have a z/OS® Connect EE-compatible version of Java™ available on the environment PATH variable. You can check the PATH variable by using the UNIX® System Services (USS) echo $PATH command.
- The z/OS® Connect installation directory `<ZCON_INST_DIR>/v3r0/wlp/bin` must be in the PATH environment variable.

The following procedure includes stopping and restarting the z/OS® Connect WLP server. In high-availability environments this may be undesirable. For alternative procedures refer to the following topics:

- Stopping and starting z/OS Connect EE servers in an HA environment
- Management of z/OS Connect EE APIs, services, and API requesters in an HA environment

1. Stop the z/OS Connect WLP server.
2. Uninstall the File Manager Service Provider feature using the installUtility command.

   For example:

   ```
   installUtility uninstall filemanager:fmProvider-2.0
   ```
3. Remove the properties file from the z/OS® Connect `<WLP_USER_DIR>/v3r0/extensions` directory using the USS rm command.

   For example:

   ```
   rm /var/zosconnect/v3r0/extensions/filemanager.properties
   ```
4. Remove the service archive files from the z/OS® Connect service directory.

   For example:

   ```
   rm /var/zosconnect/servers/<yourWLPServer>/resources/zosconnect/services/fm*.sar
   ```

   Tip: You might also decide to remove any other File Manager SAR files from this directory.
5. Edit the z/OS Connect WLP `server.xml` configuration.

   The server.xml requires the following changes:

   a. Remove the entry for the File Manager feature from the `<featureManager>` stanza.
   b. Remove the `FileManager_Connection` configuration stanza.
   c. If you have added any other `FileManager_Connection` configuration stanzas for IDs other than *"default"*, remove these also.
6. Restart the z/OS Connect WLP server.

Chapter 9. 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 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 17: Summary of steps for customizing File Manager to use LMS libraries on page 113.

Table 17. Summary of steps for customizing File Manager to use LMS libraries

Step	Description
__ 1	Determine if your LMS provides a SUBSYS interface.
__ 2	Set the LMS and LMSUBSYS options, as necessary. See LMS on page 406.
__ 3	Ensure that the correct release of Language Environment® is available to File Manager. See the third bullet point in Accessing source code in an LMS with SUBSYS interface on page 114, and the third bullet point in Accessing source code in an LMS without SUBSYS interface on page 115.
__ 4	Determine if your LMS requires your exit to run below the 16MB line. See Note on page 118.
__ 5	Write your own File Manager exit and link edit it into a load library available to File Manager, as load module FMNCRAEX. See Writing your own exit on page 117, and The library management system exit on page 560.

Accessing source code in CA-Panvalet libraries

The CA-Panvalet interface enables File Manager to work with COBOL copybooks, PL/I include books, and HLASM copybooks stored in CA-Panvalet libraries.

Before you can use 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 114.

For information about the LMS option, see LMS on page 406. For more information on how to change the default options, see Changing the default options on page 41.

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.

Note:

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 17: Summary of steps for customizing File Manager to use LMS libraries on page 113, 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 17: Summary of steps for customizing File Manager to use LMS libraries on page 113, 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.SFMNSAM1. 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. The library management system exit on page 560 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 569.

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 LMS sample exit

Sample code for the LMS user exit is provided in FMN.SFMNSAM1. There are two sample members: FMNCRAEX and FMNCRACJ.

FMNCRAEX contains the working COBOL code for the sample exit. Although the function performed by FMNCRAEX is not very useful (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 TEST. TEST is the main program and it calls the sample COBOL exit, FMNCRAEX. You can use 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 sample members FMNCRAEX and FMNCRACJ. 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 The library management system exit on page 560.

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 43.

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
```

Note:

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. See Writing the exit in HLASM on page 569 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 10. Customizing File Manager to use the Optim™ Data Privacy Provider API

To enable File Manager to specify and run ODPP commands, you must first allocate or define a DD name FMNODPP. This name points to a data set that contains the environment variables that are required by the ODPP to run as a TSO or batch application.

Example

```
//FMNODPP DD DISP=SHR,DSN=OPTIM.ODPP.SAMPLIB(ENVVARS)
```

Where the data set member ENVVARS contains the following lines:

```
LIBPATH=/usr/local/odpp/odppbin
ODPPLL=/usr/local/odpp/licensefiles
ODPPERRL=/usr//local/odpp/odppbin
ODPPTRCL=/usr/local/odpp/tracefiles
ODPPTRC=N
```

Customize the names to match the path names that are specified in the installation of the ODDP API. Refer to the relevant ODPP User's Guide and ODPP Installation Guide for more details on the environment variables required for your installation.

If the FMNODPP DD is allocated by JCL or TSO allocation statement, File Manager attempts to initialize the framework that is required for running ODPP commands. If the framework initialization is successful then you are able to define the ODPP commands as a scrambling attribute for any given field in a File Manager template. The commands are run when you run a copy operation with an input and output template and the receiving field has the scrambling option of ODPP with the associated ODPP command. For more information, see the *File Manager for z/OS® User's Guide and Reference*.

Note:

1. For TSO/ISPF users. Only one instance of File Manager can have the framework initialized. Therefore, when running split sessions under ISPF, only one session can have an active File Manager session that can support specifying and running ODPP commands.
2. You might need to increase your region sizes when running File Manager to accommodate the ODPP API. You receive a Language Environment® message that indicates *“not enough storage”* when you start File Manager if your region size is insufficient. Run with a minimum of 100 MB.
3. The IBM® Optim™ Data Privacy Solution on z/OS® needs to be Version 11 or above.

Chapter 11. 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 Customizing File Manager to use library management system libraries on page 113 and The library management system exit on page 560.

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 428. For information about changing the options in FMN0POPT, see Changing the default options on page 41. This exit name can be overridden by the user interactively or in batch. *File Manager for z/OS User's Guide and Reference* describes how to use the I/O exit.

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 Information that should always be returned from the I/O Exit to File Manager on page 131 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 428.

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 on page 123. See Using the I/O exit control block on page 130 for information on how to write your exit using the copybooks, and the processing your exit should perform.

Note:

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 section *“Using UNIX™ System Services and the Hierarchical File System”* in the *File Manager for z/OS 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 18. 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
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

Table 18. Exit data names (continued)

Description	Data type	Set by:	Value
			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

Table 18. Exit data names (continued)

Description	Data type	Set by:	Value
			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
DSORG	2X	FM open	Data set organization: 1st byte: **1** Unmovable **2** PDS(E)

Table 18. Exit data names (continued)

Description	Data type	Set by:	Value
			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

Table 18. Exit data names (continued)

Description	Data type	Set by:	Value
			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

Table 18. Exit data names (continued)

Description	Data type	Set by:	Value
			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
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.

Table 18. Exit data names (continued)

Description	Data type	Set by:	Value
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

Table 18. Exit data names (continued)

Description	Data type	Set by:	Value
			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.
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

Note:

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 132 for more information, and examples of a compression ratio.
3. See Information that should always be returned from the I/O Exit to File Manager on page 131 for more information about message suppression.

Using the I/O exit control block

The I/O exit control block is a single parameter that is passed with each call from File Manager to the I/O exit and returns 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 121. 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. 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 131.)
- 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 installs a basic exit written in HLASM. FMNUMODH installs 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 41. For information about the USEIOX option, see USEIOX on page 428.
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 123 and Using the I/O exit control block on page 130. 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 123 and Using the I/O exit control block on page 130. 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 12. Customizing File Manager to use a scrambling exit

You can use File Manager to create test data based upon production data with the option to change the values of certain fields that may contain sensitive information. This facility is known as field scrambling.

You can also write your own scrambling exit, to perform scrambling operations that are not provided in File Manager.

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 for z/OS 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:

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 on page 137. See Using the scrambling exit control block on page 143 for information on how to write your exit using the copybooks.

Note:

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 19: Exit control block on page 137

Exit control block description

Table 19. Exit control block

Description	Data type	Set by:	Value
Eye catcher	CL8	FM initialization	'UFLDXCB'
Version	F	FM initialization	**1** Initial value

Table 19. Exit control block (continued)

Description	Data type	Set by:	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
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.

Table 19. Exit control block (continued)

Description	Data type	Set by:	Value
			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:

Table 19. Exit control block (continued)

Description	Data type	Set by:	Value
			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.
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.

Table 19. Exit control block (continued)

Description	Data type	Set by:	Value
			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.

Table 19. Exit control block (continued)

Description	Data type	Set by:	Value
			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).
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 column out values have been provided on the field attributes panel for this field.

Table 19. Exit control block (continued)

Description	Data type	Set by:	Value
			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 in 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. See the information in Exit control block description on page 137 and Using the scrambling exit control block on page 143. 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

See the instructions in the jobs for changes you need to make. These jobs are distributed in FMN.SFMNSAM1.

Chapter 13. Verifying the customization of File Manager

After you have completed the initial installation and customization of File Manager base function, you can perform the following steps to verify your customization.

You might need to refer to the *File Manager for z/OS User's Guide and Reference for IMS Data*.

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 user ID 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 25), type the option value you have assigned to File Manager and press Enter. For example, if you have assigned F to File Manager, type 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 26), 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 14 Release 1 Modification 4

(not APF authorized)

Service Levels of installed components

             Base       IMS        Db2        CICS
English      -NONE-     -NONE-     -NONE-     -NONE-
```

   Note:

   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 its functions. They are:

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. See the sample jobs for information about the changes you need to make.

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 for z/OS User's Guide and Reference for IMS Data*.

Note: The File Manager for z/OS® Japanese and Korean components both require the File Manager base function product to have already been installed.

1. Log on to a TSO user ID 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 25), 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 26), 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 12 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 12 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.

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.

Part II. Customizing File Manager Db2® Component

Chapter 14. Preparing to customize FM/Db2

Before you can install and customize FM/Db2 you must have installed the File Manager base function. These topics 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. Customizing File Manager on page 12 describes the customization of File Manager base function.

⚠ **Attention:** Granting access to Db2® resources is a Db2® task, not a File Manager task. Consult your site's Db2® administrator or security administrator before proceeding. If, after careful consideration of your site's security requirements and the requirement for FM/Db2 to access the Db2® catalog, you are unable to provide FM/Db2 users with the appropriate level of access, do not attempt to install and use FM/Db2.

To function correctly, FM/Db2 requires access to various Db2® catalog tables. The configuration of Db2® security/authorization is described in the *Db2® Administration Guide*. Only general information is provided on how you might allow FM/Db2 users SELECT access to the Db2® catalog tables. 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.

There are three customization tasks you **must** perform for FM/Db2, otherwise you will not be able to use FM/Db2.

- Granting access to the Db2® catalogs, described in Granting access to the Db2 catalog (required) on page 159.
- Binding the FM/Db2 packages and plans for Db2®, described in Binding Db2 (required) on page 170.
- Customizing the **Db2 Subsystem Selection** panel, described in Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.

Table 20: Summary of steps for customizing FM/Db2 and the operating environment on page 151 lists the customization tasks that you can perform for FM/Db2. Read the referenced sections to see if you need to perform the customization described.

Checklist for installing and customizing FM/Db2

Table 20. Summary of steps for customizing FM/Db2 and the operating environment

	Description
__ 1	Concatenate FM/Db2 libraries to the LINKLIST. See Concatenating libraries to the LINKLIST on page 152.
__ 2	Modify the TSO logon procedure. See Modifying the TSO logon procedure on page 153.
__ 3	Customize to improve FM/Db2 performance. See Improving FM/Db2 performance on page 154.

Table 20. Summary of steps for customizing FM/Db2 and the operating environment (continued)

	Description
__ 4	Add FM/Db2 to the ISPF menu. See Adding FM/Db2 to the ISPF menu on page 157.
__ 5	Define FM/Db2 in an ISPF command table. See Defining FM/Db2 in an ISPF command table on page 157.
__ 6	Customize the FM/Db2 Primary Option Menu. See Customizing the FM/Db2 Primary Option Menu on page 158.
__ 7	Grant access to the Db2® catalogs - **required.** See Granting access to the Db2 catalog (required) on page 159.
__ 8	Add FM/Db2 to the Db2® Admin Launchpad See Adding FM/Db2 to the Db2 Administration Launchpad on page 163.
__ 9	Add commands to start FM/Db2 functions from Db2® Admin Tool See Adding commands to start FM/Db2 functions from the Db2 Administration Tool on page 164.
__ 10	Bind Db2® - **required**. See Binding Db2 (required) on page 170.
__ 11	Set up to run multiple versions of FM/Db2. See Running multiple versions of FM/Db2 on page 171.
__ 12	Identify all the Db2® systems that FM/Db2 will access - define these in the FMN2POPT module - **required**. See Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.
__ 13	Customize FM/Db2 options. See Customizing the FM/Db2 options on page 176.
__ 14	Change the default options. See Changing the default options on page 178.
__ 15	Change the batch JCL skeletons. See Changing the JCL skeletons for batch mode on page 179.
__ 16	Customize to protect update functions. See Customizing to protect update functions in FM/Db2 on page 180.
__ 17	Decide how to customize the FM/Db2 audit facility. See Alternatives for controlling FM/Db2 auditing on page 154.
__ 18	Customize FM/Db2 for national languages. See Customizing FM/Db2 for national languages on page 210.

Alternatives for making FM/Db2 available

You can make FM/Db2 available to 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 Customizing the operating environment for File Manager on page 20.

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 438 and DB2RLIB on page 440.

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 17.

Using LIBDEFs to allocate the FM/Db2 Libraries

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 178.
- 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 433.

Planning for running FM/Db2 with APF-authorization

If you intend to use SMF for recording audit trail information for FM/Db2, you must ensure that the File Manager load module, FMNSMF, is APF-authorized.

If you have not made File Manager base function APF-authorized, use one of the following methods to ensure that FMNSMF is authorized.

- Add FMN.SFMNMOD1 to your list of authorized libraries.
- Copy FMNSMF into an authorized library.

Note that library FMN.SFMNMODA contains File Manager modules that may need to run authorized.

Improving FM/Db2 performance

FM/Db2 performance can be improved by creating additional indexes on various Db2® catalog tables. A sample job FMN2CIX0 is provided in FMN.SFMNSAM1 to help you do this. This job contains additional indexes that are useful in improving performance when connected to Db2® version 10 or later systems.

Take a copy of the job and make the changes described in the job. You should review the indexes to be created against the current indexes defined for the Db2® catalog tables. It is possible that some of the indexes to be defined already exist, having been created as part of the installation of other products, or during the installation of previous versions of File Manager Db2®.

After the indexes have been successfully created, run the RUNSTATS utility against database DSNDB06. See your site Db2® System Administrator for additional information on how to do this.

The File Manager Db2® editor can operate in two modes: "normal" and "large". The mode of operation is determined by the value entered in the **row count** field for those FM/Db2 functions that use the FM/Db2 editor to display data. The characteristics of the two editor modes are documented in the *File Manager User's Guide and Reference for Db2® Data*.

The use of "large" editor mode may have negative Db2® performance implications. When the FM/Db2 editor operates in "large" mode, it uses a Db2® scrollable cursor for access to Db2® data. This minimizes the memory usage in the FM/Db2 user's TSO address space, but might require Db2® to build a temporary copy of the entire result table in a Db2® temporary database. For large tables, this can lead to SQLCODE-904 (unavailable resource) on table spaces defined within the Db2® temporary database. For these reasons, providing access to the FM/Db2 editor in production Db2® environments, where there are large Db2® tables, should be carefully considered.

The product installer can disable the use of "large" editor mode, by Db2® subsystem. This is achieved by setting the FMN2SSDM macro parameter EDIT_MAX_ROWS to a non-zero value. See EDIT_MAX_ROWS on page 442. The FMN2SSDM macro is discussed further in Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.

Alternatives for controlling FM/Db2 auditing

FM/Db2 auditing is an optional facility. FM/Db2 works if auditing is not implemented. You should consider:

- Whether user access to Db2® data and other resources using File Manager Db2® component requires auditing.
- The information that File Manager audit log records can provide.
- The information that File Manager audit log records cannot provide, and possible alternatives to obtaining that information.
- If you do decide to use File Manager auditing, how you will handle any issues associated with large audit log data sets, or additional SMF records.
- How you will use the information provided by File Manager audit log records.

If your site requires a record of a user's read access to Db2® data, an external security product such as RACF® can be configured to log access by some or all users, and may be a better alternative. Db2® also provides an audit facility which may be an alternative to File Manager auditing.

File Manager audit of read access to Db2® data does not write audit log records for every row processed, rather the name of the Db2® object and how many rows were processed are written to the audit log. File Manager audit of changes to Db2® data typically writes two log records, a before and after image of the row that was changed. If you intend to log update changes to Db2® tables that are subject to heavy update activity, consider the performance impact of writing many audit log records as well the size of any audit log data sets that might be produced.

You have two choices with respect to auditing of FM/Db2 audit activities. These are:

1. Use FMN2POPT controlled auditing.

   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 and copy). Other FM/Db2 functions are not subject to audit (for example print, import, 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 user IDs.
- 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 implement SAF-rule controlled auditing you need to decide how File Manager auditing will be enabled. This is described in more detail in Customizing the audit facility for FM/Db2 on page 186. There are two alternatives. One requires an enabling SAF rule and the presence of a member in SYS1.PARMLIB, the other requires an enabling SAF rule but has no requirement for a member in SYS1.PARMLIB. The use of a member in SYS1.PARMLIB provides additional facilities compared with the alternative that does not require the use of SYS1.PARMLIB. The additional facilities are documented in File Manager options specified in PARMLIB members on page 532.

When you have determined the appropriate type of auditing for your installation, follow the instructions in Customizing the audit facility for FM/Db2 on page 186.

Chapter 15. 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 15: on page 157. You could add FM/Db2 in your Primary Option Menu after the File Manager base function.

Figure 15.

```
#
)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        FM/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

#
```

To invoke FM/Db2 with LIBDEFs, see Example 1. Invoking File Manager, FM/IMS, and FM/Db2 primary options from a selection panel on page 24.

For 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, 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 entering FD on the command line.

Verb	FD
Action	SELECT PANEL(FMN2ST00) OPT(&ZPARM) SCRNAME(FMDB2) SUSPEND NEWAPPL(FMN2)
Description	File Manager for z/OS® Db2® Component

To invoke FM/Db2 with LIBDEFs, see Example 2. ISPF command table with LIBDEF invocations on page 24.

For information about ISPF command tables, see z/OS ISPF Planning and Customizing and *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. If the DB2I option is selected, 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, you may need to modify the FMN2ST00 panel or the FMN2DB2I exec.

Owing to the wide range of possible customizations the information provided here is not specific to any particular method.

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.

Note:

1. FM/Db2 provides two methods for allocating the File Manager libraries. See Modifying the TSO logon procedure on page 153 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 is a *"full-function"* Db2® application, intended to provide access to every Db2® catalog table, including those catalog tables that may contain sensitive information. 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.

FM/Db2 includes an option (Db2® Privileges utility) that enables information in the various *AUTH Db2® catalog tables to be viewed, and possibly changed. If your installation restricts access to the *AUTH tables, you can disable the Db2® Privileges utility entirely, and remove the need to grant SELECT access on the *AUTH tables at install time. You can disable the FM/Db2 Privileges utility for some Db2® systems, but not for others as required.

To disable access to the Db2® Privileges utility you need to:

- Code `AUTH_ACCESS=N` in the FMN2SSDM macro entry for each Db2® system where no SELECT access to the *AUTH tables is allowed. See the parameter description AUTH_ACCESS on page 436 for more information.
- Remove or comment out references to the *AUTH tables in the sample job used to grant SELECT access on the Db2® catalog tables to FM/Db2 users. See Sample jobs to grant SELECT access on the Db2 catalog tables on page 161 for additional information.

You can also further restrict access to some Db2® catalog tables - see Sample jobs to grant SELECT access on the Db2 catalog tables on page 161 for additional information. FM/Db2 is designed to tolerate incomplete access to most Db2® catalog tables, although the functionality of the product is reduced when access to some Db2® catalog tables is reduced. Restricting access to certain key Db2® catalog tables renders the product inoperative - for an indicative list of the key tables and columns see the *“minimal subset”* sample members described below.

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 170 for further information. The effect of DYNAMICRULES(RUN) behavior is that FM/Db2 uses the *run behavior* for dynamic SQL statements, as summarized here:

- Db2® uses the authid of the application process and the SQL authid (the value of the CURRENT SQLID special register) for authorization checking of dynamic SQL statements.
- Db2® uses the authid of the application process and the SQL authid (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 authid 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 logon ID is used as the Db2® authid, so SELECT access on the appropriate Db2® catalog tables needs to be granted to a list of Db2® authids which would be the same IDs as the list of the TSO logon IDs 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 complex 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 TSO logon ID 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 approach to granting FM/Db2 users SELECT access to Db2® catalog tables is 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 on page 161) 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 logon ID 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 TSO logon ID.
- As an additional security measure, grant EXECUTE on the FM/Db2 plans to the generic Db2® authid, rather than PUBLIC.

When implemented, this 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 logon ID - to the Db2® catalog tables.
5. Only those users with access to the generic Db2® authid can run 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. FMN2GSC0 (Db2® V10)
2. FMN2GSC1 (Db2® V11)
3. FMN2GSC2 (Db2® V12)
4. FMN2GVW0 (Db2® V10)
5. FMN2GVW1 (Db2® V11)
6. FMN2GVW2 (Db2® V12)
7. FMN2GV20 (Db2® V10)
8. FMN2GV21 (Db2® V11)
9. FMN2GV22 (Db2® V12)

You should review the GRANT statements in these members whenever you install a later version of Db2®. New versions of Db2® may include new or updated catalog tables, and FM/Db2 may require access to these new or updated tables.

Sample jobs FMN2GSC*n* 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 FMN2GVW*n*

- 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 455. For information about changing the options in FMN2POPT, see Changing the default options on page 178.

Sample jobs FMN2GV2*n*

- 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 some cases 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 455. For information about changing the options in FMN2POPT, see Changing the default options on page 178.

Note:

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. For this 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 run the SQL statement.
    - If this attempt fails, 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 tables.
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 user IDs.
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 that do this are provided in FMN.SFMNSAM1:
    - **FMN2GV20** for Db2® version 10 subsystems.
    - **FMN2GV21** for Db2® version 11 subsystems.
    - **FMN2GV22** for Db2® version 12 subsystems.
4. 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 will reflect the columns of the view and not necessarily all columns of the catalog table, reducing the usefulness of the product.

5. If your installation security standards prevent access to the SYSIBM.*AUTH tables, you can 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).
6. You can define different views for different Db2® subsystems. For example, you could define the views in FMN2GVW0 against a development system, but the views in FMN2GV20 against a production system. You must ensure, however, that the owner (CATOWNER value) used is consistent across all Db2® subsystems accessed by FM/Db2.
7. 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 21: Values for Db2 Admin Launchpad definition on page 164.
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. In your copy of FMN2ADIN, change the names for the File Manager libraries in the following statements to the appropriate values, if you are not using the default values.

```
sfmnexec = "'FMN.SFMNEXEC'"
sfmnllib = "'FMN.SFMNMOD1'"
sfmnmlib = "'FMN.SFMNMENU'"
sfmnplib = "'FMN.SFMNPENU'"
sfmnslib = "'FMN.SFMNSLIB'"
sfmntlib = "'FMN.SFMNTENU'"
```

Note:

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 21. Values for Db2® Admin Launchpad definition

Admin Tool Field	Value
PID	5655-Q42
REL	1410
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. In your copy of FMN2ADIE, change the names for the File Manager libraries in the following statements to the appropriate values, if you are not using the default values.

```
sfmnexec = "'FMN.SFMNEXEC'"
sfmnllib = "'FMN.SFMNMOD1'"
sfmnmlib = "'FMN.SFMNMENU'"
sfmnplib = "'FMN.SFMNPENU'"
sfmnslib = "'FMN.SFMNSLIB'"
sfmntlib = "'FMN.SFMNTENU'"
```

Note:

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.
- The SQLID used when FM/Db2 is started is the same as the users TSO logon ID, unless modified by a Db2® authorization exit.
- 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 before 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 22: Parameters for FMN2INEX on page 166.

The function codes that can be specified are shown in Table 23: Function codes for FMN2INEX on page 166.

For example, referring to Table 22: Parameters for FMN2INEX on page 166 and Table 23: Function codes for FMN2INEX on page 166, you might build the following parameter string:

```
SSID(DFE2) OW(DSN81210) NM(EMP) F(4.5)
```

In this case your select statement would be:

```
"SELECT MODE(FSCR) CMD(%FMN2INEX SSID(DFE2) OW(DSN81210) NM(EMP) F(4.5)),
NEWAPPL(FMN2) PASSLIB SUSPEND"
```

Table 22. 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 23: Function codes for FMN2INEX on page 166 for additional exceptions.
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 23: Function codes for FMN2INEX on page 166 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 23. 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.

Table 23. Function codes for FMN2INEX (continued)

Function code	Description
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.
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.

Table 23. Function codes for FMN2INEX (continued)

Function code	Description
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.
- The SQLID used when FM/Db2 is started is the same as the users TSO logon ID, unless modified by a Db2® authorization exit.
- 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 23: Function codes for FMN2INEX on page 166 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 16. 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. The sample jobs shown here are provided in FMN.SFMNSAM1:

- **FMN2BN0P** Db2® V10 binds the FM/Db2 plans (run once).
- **FMN2BN0K** Db2® V10 binds the FM/Db2 packages.
- **FMN2BN1P** Db2® V11 binds the FM/Db2 plans (run once).
- **FMN2BN1K** Db2® V11 binds the FM/Db2 packages.
- **FMN2BN2P** Db2® V12 binds the FM/Db2 plans (run once).
- **FMN2BN2K** Db2® V12 binds the FM/Db2 packages.

The FMN2BN*n*K jobs must be run once and then rerun when maintenance changes the Db2®-specific parts of the product.

Note: File Manager provides sample bind jobs only for supported versions of Db2® running in new-function mode (NFM).

Attention: The packages used in File Manager version 14 are different to previous File Manager versions. Re-running the bind jobs derived from an earlier version of File Manager will not work.

It is a requirement that FM/Db2 is bound using the DYNAMICRULES(RUN) bind option (the default). The use of DYNAMICRULES(BIND) prevents FM/Db2 from being able 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 FMN2BN*n*K 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.

Select the appropriate bind jobs, depending on the version of the target Db2® system. Run both the "bind plan" and "bind package" jobs, 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.

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 the appropriate bind plan and bind package jobs for each Db2® subsystem you want FM/Db2 to use. 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 bind jobs against **every Db2® system that FM/Db2 will access**, either directly or remotely. The FMN2BN*n*K jobs require 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 FMN2BN*n*K jobs when these are 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 bind jobs (FMN2BN*n*P and FMN2BN*n*K as appropriate) complete 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.
- 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.

You can change the plan names for the FM/Db2 plan and the FM/Db2 reverse engineering plan. To do this modify FMN2BN*n*P as appropriate. 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 Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172. For information about the FMN2SSDM macro, see FMN2SSDM on page 433.

Running multiple versions of FM/Db2

This topic explains how you can install multiple different versions of FM/Db2.

⚠ **Attention:** You must ensure that each FM/Db2 installation is completely independent of any others.

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 are to be run concurrently while acceptance testing is completed for the newer version.

The default plan names and package names are used for the existing version of FM/Db2 that is already installed:

- FMN2PLAN
- FMN2GEN
- FMN2PLPK
- FMN2GNPK

For the new version of FM/Db2, select new plan and package names, for example:

- FMN2PLND (instead of FMN2PLAN)
- FMN2GEND (instead of FMN2GEN)
- FMN2PLPD (instead of FMN2PLPK)
- FMN2GNPD (instead of FMN2GNPK)

Any new names can be selected as long as they are different to all other FM/Db2 plan and package names that are already in use.

For the new version of FM/Db2, modify the sample bind jobs FMN2BN*n*P and FMN2BN*n*K as appropriate.

Run the modified bind jobs, both the bind plan and bind package, on every Db2® system. 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=FMN2PLND,PLAN2=FMN2GEND
```

Assemble the new FMN2POPT module, and ensure that this is available when a user runs the new version of Db2®.

For information about modifying FMN2POPT, see Changing the default options on page 178. For information about FMN2SSDM, see FMN2SSDM on page 433.

Defining all Db2® systems that FM/Db2 will access in FMN2POPT (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.

If your site uses Db2® data sharing groups, review the explanatory information for the LIST parameter of the FMN2POPI macro in LIST on page 462. This information explains how FM/Db2 determines the Db2® systems that are candidates for connection on the local z/OS® server.

You must then install FMN2POPT on your FM/Db2 system, using the usermod FMN2UMDP. FMN2SSDM on page 433 describes the syntax of the options in FMN2SSDM. See Changing the default options on page 178 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 455 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 433. 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 on page 173 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.

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 9, and data sharing groups for Db2® version 10. 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=DF52,PLAN=FMN2PLNE,                              +  DF52
               TYPE=SUBSYS,                                             +
               AUDIT=(REQUIRED,SMF),                                    +
               EDIT_MAX_ROWS=0,                                         +
               DB2LLIB=('DB2VA10.DF52.SDSNEXIT','DB2.VA10.SDSNLOAD'),  +
               DB2RLIB=('DB2VA10.DF52.RUNLIB.LOAD'),                    +
               DB2CLIB=('DB2.VA10.SDSNCLST'),                           +
               DB2MLIB=('DB2.VA10.SDSNSPFM'),                           +
               DB2PLIB=('DB2.VA10.SDSNSPFP','DB2.VA10.SDSNPFPE'),       +
               DB2SLIB=('DB2.VA10.SDSNSPFS'),                           +
               DB2TLIB=('DB2.VA10.SDSNSPFT'),                           +
               DB2PROC=('DB2VA10.DF52.PROCLIB'),                        +
               DESC='DB2 Version 10.1 - Production'
*
         FMN2SSDM SSID=DF61,PLAN=FMN2PLNE,                              +  DF61
               DISPLAY=HIDDEN,                                          +
               AUDIT=OPTIONAL,                                          +
               DB2LLIB=('DB2VB10.DF61.SDSNEXIT','DB2.VB10.SDSNLOAD'),  +
               DB2RLIB=('DB2VB10.DF61.RUNLIB.LOAD'),                    +
               DB2CLIB=('DB2.VB10.SDSNCLST'),                           +
               DB2MLIB=('DB2.VB10.SDSNSPFM'),                           +
               DB2PLIB=('DB2.VB10.SDSNSPFP','DB2.VB10.SDSNPFPE'),       +
               DB2SLIB=('DB2.VB10.SDSNSPFS'),                           +
               DB2TLIB=('DB2.VB10.SDSNSPFT'),                          +
               DB2PROC=('DB2VB10.DF61.PROCLIB'),                       +
               DESC='DB2 Version 11.1 – Development'
*
         FMN2SSDM SSID=DF13,PLAN=FMN2PLNE,                             +  DF13
               TYPE=SUBSYS,                                            +
               AUDIT=REQUIRED,                                         +
               EDIT_MAX_ROWS=0,                                        +
               FORCE_WITH_UR=Y,                                        +
               TABLE_LOCKING=NO,                                       +
               DB2LLIB=('DB2VC10.DF13.SDSNEXIT','DB2.VC10.SDSNLOAD'),  +
               DB2RLIB=('DB2VC10.DF13.RUNLIB.LOAD'),                   +
               DB2CLIB=('DB2.VC10.SDSNCLST'),                          +
               DB2MLIB=('DB2.VC10.SDSNSPFM'),                          +
               DB2PLIB=('DB2.VC10.SDSNSPFP','DB2.VC10.SDSNPFPE'),      +
               DB2SLIB=('DB2.VC10.SDSNSPFS'),                          +
               DB2TLIB=('DB2.VC10.SDSNSPFT'),                          +
               DB2PROC=('DB2VC10.DF13.PROCLIB'),                       +
               TMPDDLN=DF13TMDD,                                       +
               DB2SLIB=('DB2.VC10.SDSNSPFS'),                          +
               DB2TLIB=('DB2.VC10.SDSNSPFT'),                          +
               DB2PROC=('DB2VC10.DF13.PROCLIB'),                       +
               TMPDDLN=DF13TMDD,                                       +
               STMJCL1='DISP=SHR,DSN=FIRST.TEMPLATE.LIBRARY,',         +
               STMJCL2='DISP=SHR,DSN=SECOND.TEMPLATE.LIBRARY',         +
               OPTEVT1='ITEMERROR,HALT',                               +
               CPYCPYN=DF13CPYC,                                       +
               LODINDN=DF13LDIN,                                       +
               DESC='DB2 Version 12.1 – Production'
*
         FMN2SSDM SSID=DG03,PLAN=FMN2PLNE,                             +  DG03
               TYPE=GROUP,                                             +
               DISPLAY=YES,                                            +
               DB2LLIB=('DB2VB10.DF61.SDSNEXIT','DB2.VB10.SDSNLOAD'),  +
               DB2RLIB=('DB2VB10.DF61.RUNLIB.LOAD'),                   +
               DB2CLIB=('DB2.VB10.SDSNCLST'),                          +
               DB2MLIB=('DB2.VB10.SDSNSPFM'),                          +
               DB2PLIB=('DB2.VB10.SDSNSPFP','DB2.VB10.SDSNPFPE'),      +
               DB2SLIB=('DB2.VB10.SDSNSPFS'),                          +
               DB2TLIB=('DB2.VB10.SDSNSPFT'),                          +
               DB2PROC=('DB2VB10.DF61.PROCLIB'),                       +
               DESC='DB2 Group Attach ID number 1'
*
         FMN2SSDM SSID=DEFAULT                                      DEFAULT
```

Note: In this example the + characters are in column 72.

DF52

1. These statements define a Db2® 10 subsystem, called `DB2 Version 10.1 – Production`.
2. The SSID is DF52.
3. Auditing is required. Audit records are written as SMF records.
4. EDIT_MAX_ROWS is set to 0. This disables *“large”* editor mode for all users.
5. The Db2® library names are specified using DB2LLIB, DB2RLIB, and so on.

DF61

1. These statements define a Db2® 11 subsystem, called `DB2 Version 11.1 – Development`.
2. The SSID is DF61.
3. Auditing is optional.
4. DISPLAY=HIDDEN is specified. This Db2® subsystem will not appear on the File Manager Db2® subsystem selection list.
5. The Db2® library names are specified using DB2LLIB, DB2RLIB, and so on.

DF13

1. These statements define a Db2® 12 subsystem, called `DB2 Version 12.1 – Production`.
2. The SSID is DF13.
3. Auditing is required. Audit records are written to data sets.
4. EDIT_MAX_ROWS is set to 0. This disables *"large"* editor mode for all users.
5. FORCE_WITH_UR is set to Y. This adds the `WITH UR` clause to any SQL statements executed by the FM/Db2 editor.
6. TABLE_LOCKING is set to NO. This prevents any user specifying the *"Lock table"* option when using the FM/Db2 editor.
7. The TMPDDLN keyword specifies a ddname for the site-specific template library of DF13TMDD. 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:

   ```
   //DF13TMDD DD DISP=SHR,DSN=FIRST.TEMPLATE.LIBRARY
   //                    DD DSN=SECOND.TEMPLATE.LIBRARY
   ```
8. The OPTEVT1 keyword will generate this OPTIONS EVENT statement:

   ```
   OPTIONS EVENT(ITEMERROR,HALT)
   ```
9. The CPYCPYN keyword specifies a template name of DF91CPYC to be used in a COPYDDN clause in a COPY utility statement.
10. The LODINDD keyword specifies a template name of DF13LDIN to be used in an INDDN clause in the LOAD utility statement.
11. The Db2 library names are specified using DB2LLIB, DB2RLIB, and so on.

DG03

1. These statements define a Db2® 11 data sharing group called `DB2 group attach ID number 1`.
2. The SSID is DG03.
3. TYPE=GROUP specifies that the entry is for a Db2® group, not a Db2® subsystem.
4. The Db2® library names are specified using DB2LLIB, DB2RLIB, and so on.

DEFAULT

This statement is required. Its only parameter is DEFAULT. The last (or only if there is only one) FMN2SSDM macro invocation must specify DEFAULT.

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 455. 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 178 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.

Usage tips

The three most important FMN2POPI and FMN2SSDM parameters for controlling the FM/Db2 subsystem selection list are:

- CONNECT (FMN2POPI)
- LIST (FMN2POPI)
- DISPLAY (FMN2SSDM)

The default for the CONNECT parameter is DEFINED. Consider changing this to ANY. With CONNECT=DEFINED, FM/Db2 will only attempt connection to Db2 systems that are defined in the FMN2POPT.

The default for the LIST parameter is ALL. This can be changed to show any combination of active, inactive, or unavailable subsystems or groups.

The default for the DISPLAY parameter is YES. A Db2® subsystem or group will be displayed if it is eligible to be displayed based on how the FMN2POPI LIST parameter is set.

You can prevent inactive or unavailable Db2® systems or groups being displayed on the FM/Db2 subsystem selection list by specifying LIST=ACTIVE.

You can prevent an *individual* Db2® system or group being displayed on the FM/Db2 subsystem selection list by specifying DISPLAY=HIDDEN. A typical use of this feature is to hide one or more Db2® subsystems that are also part of a Db2® data sharing group, leaving the Db2® data sharing group visible.

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 File Manager options on page 383.
3. Include your FMN2SSDM macros, (as described in Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172, and FMN2SSDM on page 433), in the FMN2POPT member in your library.

4. Include your FMN2POPI macros (if required), (as described in Customizing the FM/Db2 options on page 176, and FMN2POPI on page 455), 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

You can change the default national language for your installation that is used by FM/Db2 for 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 405 for more information about the LANGUAGE option.

For other customization you can do for FM/Db2 for national languages, refer to Customizing FM/Db2 for national languages on page 210.

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 job control skeletons. 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 433.)

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, 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.

Modify FMN2FTEX using the usermod FMN2UMDE, which 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 24. FM/Db2 update functions

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

To protect update functions, specify SEC=YES in the FMN2POPT module (see the SEC parameter in File Manager options on page 383).

The facility class rules that are required are:

FILEM.DB2.UPDATE
FILEM.FUNCTION.*function_code*

Example 1: To protect all Db2® update functions

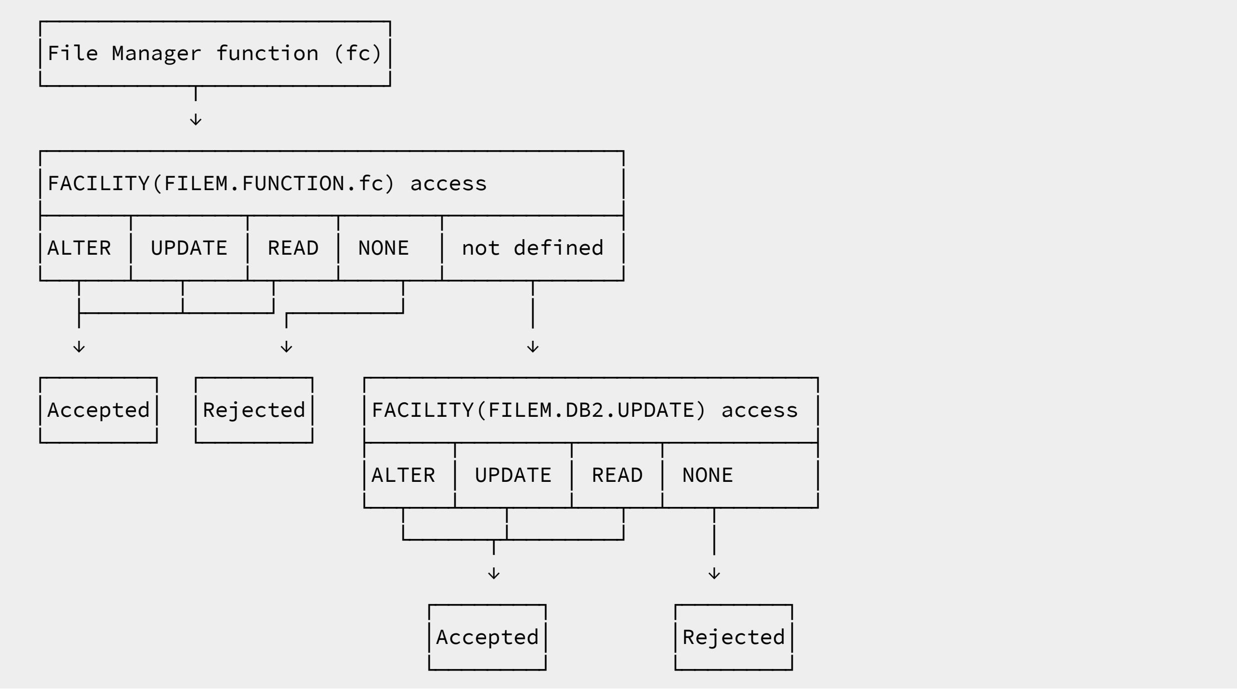
- Specify SEC=YES in the FMN2POPT
- Write a facility class rule for FILEM.DB2.UPDATE

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 following diagram shows the processing that is used when function rules, update facility class rules, or both are specified.

Figure 16. Security system validation for update functions


File Manager function (fc)
FACILITY(FILEM.FUNCTION.fc) access
ALTER
UPDATE
READ
NONE
not defined
Accepted
Rejected
FACILITY(FILEM.DB2.UPDATE) access
ALTER
UPDATE
READ
NONE
Accepted
Rejected

Note: In most cases it is preferable to use Db2® security, with or without an external security server, to control update access to Db2® objects. Db2® security allows access to be specified for individual Db2® objects at various levels of access (SELECT, INSERT, UPDATE, DELETE), and to individual Db2® authids. The same level of control is not possible using FM/Db2.

Customizing FM/Db2 for use in production environments

If you intend to use FM/Db2 to access Db2® systems that support a high throughput online transaction processing environment, you should consider the potential for adverse performance and concurrency problems arising from the use of FM/Db2, particularly the FM/Db2 editor.

This section outlines concurrency and performance issues that might arise:

- The use of *"large"* mode in the FM/Db2 editor.

  The File Manager Db2® editor can operate in two modes: *"normal"* and *"large"*. The mode of operation is determined by the value entered in the **row count** field for those FM/Db2 functions that use the FM/Db2 editor to display data. The characteristics of the two editor modes are documented in the *File Manager User's Guide and Reference for DB2 Data*.

  The use of *"large"* editor mode might have negative Db2® performance implications. When the FM/Db2 editor operates in *"large"* mode, it uses a Db2® scrollable cursor for access to Db2® data. This minimizes the memory usage in the FM/Db2 user's TSO address space, but might require Db2® to build a temporary copy of the entire result table in a Db2® temporary database. For large tables, this can lead to SQLCODE-904 (unavailable resource) on table spaces defined within the Db2® temporary database. For these reasons, providing access to the FM/Db2 editor in production Db2® environments, where there are large Db2® tables, should be carefully considered.

The product installer can disable the use of *"large"* editor mode, by Db2® subsystem. This is achieved by setting the FMN2SSDM macro parameter

  - EDIT_MAX_ROWS to a non-zero value. See EDIT_MAX_ROWS on page 442.
  - PROD_EDIT to YES. See PROD_EDIT on page 448. Note that setting PROD_EDIT=YES over-rides EDIT_MAX_ROWS=0, if specified, replacing it with EDIT_MAX_ROWS=1000. The FMN2SSDM macro is discussed further in Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.

- The use of table locking options

  The File Manager Db2® editor has an option that allows a table to be locked for the duration of the editor session. The table can be locked in *"SHARED"* or *"EXCLUSIVE"* mode, see the description of the 'LOCK TABLE' statement in the SQL Reference manual for the appropriate version of Db2® for a description of these options.

  The use of either the *"SHARED"* or *"EXCLUSIVE"* options is inconsistent with the requirements for a high throughput online transaction processing environment. You can disable the editor table locking option, by Db2® subsystem. This is achieved by setting the FMN2SSDM macro parameters:

  - TABLE_LOCKING to NO. See TABLE_LOCKING on page 452.
  - PROD_EDIT to YES. See PROD_EDIT on page 448. Note that setting PROD_EDIT=YES will over-ride TABLE_LOCKING=YES, if specified, replacing it with TABLE_LOCKING=NO. The FMN2SSDM macro is discussed further in Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.

- User specified SELECT statements and FM/Db2 editor choice

  Various FM/Db2 functions allow users to create and execute SELECT statements. The FM/Db2 editor is used to display the result table for these statements. There is a system option controlling whether the result table is displayed in browse, view or edit modes.

  FM/Db2 functions that process SELECT statements include the following:

  Prototype SELECT statements (basic)
  Prototype SELECT statements (advanced)
  Enter, execute and explain SQL statements
  Edit and execute SQL statements from a data set

  You can disable the use of the FM/Db2 editor, in edit mode, when processing the result table for user specified SELECT statements, by Db2® system. This is achieved by setting the FMN2SSDM macro parameter USER_SELECT_EDIT to NO. See USER_SELECT_EDIT on page 454.

  The FMN2SSDM macro is discussed further in Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.

- The use of Cursor concurrency options

The FM/Db2 editor has an option that allows a *"WITH"* clause to be added to the declaration for the cursor used to access Db2® data. See the description for *"isolation-clause"* in the SQL Reference manual for the appropriate version of Db2®.

The default for this option is to not add any WITH clause, which results in Db2® taking locks based on system options, and options specified for the table spaces that contain the Db2® object. The following editor options are also relevant:

Read-only access
Commit after data fetch

The *"Uncommitted read"* concurrency option can be specified when you need to avoid any locks being taken by FM/Db2 users when they read data prior to display in an editor session. You can force the use of *"uncommitted read"*, by Db2® system. This is achieved by setting the FMN2SSDM macro parameter FORCE_WITH_UR to Y. See FORCE_WITH_UR on page 444.

The FMN2SSDM macro is discussed further in Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.

- The PROD_EDIT parameter

  The FMN2SSDM macro parameter PROD_EDIT has multiple effects on the behavior of the File Manager Db2® editor, as described below:

  - An inactivity timer applies to all FM/Db2 editor sessions. If EDITOR_TIMEOUT=0 is specified in the same FMN2SSDM macro, this value is replaced with EDITOR_TIMEOUT=60.
  - Table locking is disabled. If TABLE_LOCKING=YES is specified in the same FMN2SSDM macro, this value is replaced with TABLE_LOCKING=NO. When a FM/Db2 user displays the Editor Locking option, the value is set to '1' (None) and cannot be altered. This prevents the user changing the option to either SHARED or EXCLUSIVE.
  - *"Large"* edit mode is disabled. If EDIT_MAX_ROWS=0 is specified in the same FMN2SSDM macro, this value is replaced with EDIT_MAX_ROWS=1000. When a FM/Db2 user enters 0 in the row count field on an editor-related function entry panel, the value is ignored and replaced with the value specified for EDIT_MAX_ROWS. This prevents the user accessing the FM/Db2 editor in *"large"* mode.
  - The *"Commit after data fetch"* option is fixed and is selected.
  - The *"Commit when save issued"* option is fixed and is selected.
  - The *"Commit when no save errors"* option is fixed and is not selected.

  The *"Commit after data fetch"* option ensures that any user of the FM/Db2 editor has no locks against Db2® when the data in the editor session is initially displayed, and the user cannot change the option. The other two options ensure that a Db2® commit is always issued when a user attempts to save changes made in the current editor session.

  The commit is always issued, whether there are save errors or not. When considered in combination, the PROD_EDIT behavior reduces locking against the Db2® object being edited to a minimum. The only time locks are held is when:

  - The data is initially accessed, prior to the display of the data.
  - Whenever changes made in the editor session are presented to Db2® for validation.
- Restricting the use of the FM/Db2 editor and educating users

  Setting the FMN2SSDM parameters as described in above reduces the potential for unwanted locking or resource constraints arising from the use of the FM/Db2 editor, it does not eliminate it.

  The FM/Db2 editor is a powerful tool. You should consider carefully which users, if any, have access to the FM/Db2 editor in production Db2® environments.

  You should consider educating any users of the FM/Db2 editor in production environments. The main issues to be considered are:

  - Minimizing the number of rows of data retrieved for any editor session. The fewer rows that are retrieved, the smaller the number of locks that might be taken against production data bases.
  - Minimizing the duration of any editor sessions against production data. The Db2® *"unit of work"* concept is relevant here, it is recommended that any users of the FM/Db2 editor against production data are familiar with the concept, and potential problems that might arise when the elapsed time for a unit of work becomes long.

  The following example demonstrates the potential problem.

  The user starts an FM/Db2 editor session of a production table. They have the editor option *"uncommitted read"* on and use row selection criteria to limit the number of rows in the editor session to 1000.

  The user issues a change command that affects 800 rows. The user issues the SAVE command to validate the changes made. The editor option *"Commit when save issued"* is turned off. The changes are successful: all of the 800 rows that were changed are updated in Db2®.

  The user is called away, goes to lunch or finishes work for the day, leaving the editor session active.

  In this scenario the FM/Db2 editor session starts with no locks against the production database. When the SAVE command is issued, the 800 rows that are changed are marked in Db2® with update locks. The editor option *"Commit when save issued"* option is off, so there is no COMMIT issued and the Db2® unit of work remains active. The Db2® locks against the 800 rows remain active until the Db2® unit of work ends, which occurs when the user cancels the editor session, the user ends the editor session, or the user's TSO session is cancelled. This might be hours, even days, in the scenario outlined above. During that time other users of the production database may not be able to access those rows.

  The issues outlined in this scenario can be avoided if PROD_EDIT=YES is specified.

  Users of the FM/Db2 editor in production environments should be educated to minimize the elapsed time for any editor sessions that make changes to production data. This minimizes the duration of any locks taken against production data.

Chapter 17. Customizing the audit facility for FM/Db2

FM/Db2 can write audit log records to either SMF or an audit log data set.

If auditing is not required, you should:

- Set AUDIT=NONE in the FMN2SSDM macro specification of the FMN2POPT module. There is one FMN2SSDM specification for each Db2® system accessed by FM/Db2. Specify AUDIT=NONE for each FMN2SSDM specification where auditing is not required.

  See AUDIT on page 434 for more information.
- Set SMFNO=0 in the FMN0POPI macro specification of the FMN2POPT module. See SMFNO on page 421 for more information.

You can skip the customization described in the rest of this chapter.

File Manager provides two different methods (FMN2POPT-controlled and SAF-controlled audit) for controlling whether audit records are written for FM/Db2. These are described in detail in Alternatives for controlling FM/Db2 auditing on page 154.

Use the checklist to determine the customization required for File Manager Db2® audit facility.

Table 25. Checklist for customizing FM/Db2 auditing. This table lists choices and decisions.

Audit customization choice	Decision (Yes|No|Not applicable)
1. Control auditing using the FMN2POPT options module	
2. Control auditing using SAF rules and a member in SYS1.PARMLIB	
3. Control auditing using SAF rules, without any changes to SYS1.PARMLIB	
4. Audit records are to be written to a data set	
5. Audit records are to be written to SMF	

For choices 1 to 3, you should answer YES for one choice only. Mark the other two choices as 'Not applicable'.

If you answer YES for choice 1, you can answer YES for one of choices 4 and 5. Mark the other choice as 'Not applicable'.

If you answer YES for choices 2 or 3, you can answer YES for one or both of choices 4 and 5. If you answer YES for both choices you are implementing dual-logging, which is only available with SAF-controlled auditing.

Once you have completed the checklist, use the table below to identify the customization that is required for each customization choice. Only complete the customization if your decision was *"YES"* in the checklist.

Table 26. Customization steps for audit customization choices. This table lists choices and related actions.

Customization choice	Sections to complete
1. Control auditing using the FMN2POPT options module	• FMN2POPT-controlled auditing on page 187
2. Control auditing using SAF rules and a member in SYS1.PARMLIB	• SAF-controlled auditing for File Manager Db2 component on page 189 • Implementing SAF-rule controlled auditing on page 193
3. Control auditing using SAF rules, without any changes to SYS1.PARMLIB	• SAF-controlled auditing for File Manager Db2 component on page 189 • Implementing SAF-rule controlled auditing on page 193
4. Audit records are to be written to a data set	• Audit data set configuration on page 188
5. Audit records are to be written to SMF	• Customizing File Manager to write audit records to SMF on page 79

FMN2POPT-controlled auditing

Use the AUDIT option in FMN2SSDM to customize how FM/Db2 records an audit trail.

- 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, but the audit option is reset to "not selected" when 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 is produced for each user, regardless of whether the user wants to record audit information. Specify AUDIT=(REQUIRED,....).

- Demand. An audit trail is produced for each user, regardless of whether the user wants to record audit information. An audit report job is submitted when the edit function ends or when you change SSIDs. Specify AUDIT=(DEMAND,....).

  The job submitted is determined by the skeleton member FMN.SFMNSLIB(FMN2FTAD). Customize the job card and JCL to specify the reporting options you require. To change the audit report options in the skeleton see "AUD (Print Audit Trail Report)" in the *File Manager User's Guide and Reference*.

When recording of audit information is optional, this is determined by the individual setting of Create an audit trail in the user's 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 is 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 435 for information about AUDITBROWSE.

Audit data set configuration

The format of the audit log data set name is determined by the setting of the AUDITHLQ parameter in the FMN0POPI definition in FMN2POPT.

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

Any 1-8 character constant that is valid in the context of a data set name.

ssid

The name of the currently connected (local) Db2® system.

userid

The user ID creating the data set.

Dyymmdd

The date of the activity.

Thhmmss

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 is null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix is also 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.

Set the AUDITHLQ parameter in the FMN0POPI macro to the required value, based on your site's requirements. See AUDITHLQ on page 386 for more information and examples.

You can print the information in an audit data set using the FM/Db2 Print Audit Trail utility. To do this select option 3.10 from the FM/Db2 Primary Option menu.

SAF-controlled auditing for File Manager Db2® component

There are two methods for implementing SAF-controlled auditing for FM/Db2. These are:

1. Using an enabling SAF Facility class rule and a member in SYS1.PARMLIB.

   To use this method complete the customization described in SAF-controlled auditing using SYS1.PARMLIB on page 190.
2. Using an enabling SAF Facility class rule, without any changes to SYS1.PARMLIB.

   To use this method complete the customization described in SAF-controlled auditing without SYS1.PARMLIB on page 192.

! **Important:** If you use a security product other than RACF®, review the information in When a security product other than RACF is in use on page 87 to avoid one possible cause of S047 abends.

SAF-controlled auditing using SYS1.PARMLIB

You need to define an enabling SAF facility profile as described below:

Define SAF facility profile

```
FILEM.PARMLIB.DB2
```

and ensure all File Manager Db2® users to be audited have at least read access to that facility. See the example below:

Example

User PROD1 to have SAF-rule controlled auditing using SYS1.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)
```

Add member FMN2PARM to SYS1.PARMLIB (or any other library in the logical parmlib concatenation). Defining the FMN2PARM member on page 190.

Once the above SAF rule is defined and activated, auditing for File Manager Db2® component users is controlled by the FMAUDIT parameter in the FMN2PARM member. See FM/Db2 options specified in FMN2PARM on page 544 for more information. If audit log records are to be written to SMF, the SMF record number is specified as an FMAUDIT parameter option. See FMAUDIT on page 545, and SMF_NO on page 545.

Note: File Manager Db2® component does not start if a user has read access to the above facility and the FMN2PARM member does not exist in the logical parmlib concatenation.

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.

Defining the FMN2PARM member

If auditing is to be controlled from parmlib (user has read access to FILEM.PARMLIB.DB2, see SAF-controlled auditing for File Manager Db2 component on page 189) 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 a FMSECRTY statement. This option is not used in FM/Db2 and can be either omitted or commented out. It has no effect.

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: FMN2POPT 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 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 parmlib concatenation.

This will not work, the attempt by an FM/Db2 user to access the logical 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.
   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
2. Add member FMN2PARM to the new library, specifying the appropriate FMAUDIT parameter.
3. 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 step 1 on page 192. If the FMN2PARM member specifies any include statements (see Facilities for customizing the FMN2PARM definitions on page 546), all of the included members must also reside in the same library.

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 FM/Db2 options specified in FMN2PARM on page 544 for more information.

SAF-controlled auditing without SYS1.PARMLIB

You need to define an enabling SAF facility profile as described below:

Define SAF facility profile

```
FILEM.SAFAUDIT.DB2
```

and ensure that all FM/Db2 users to be audited have at least read access to that facility. See the example below.

Example

User PROD2 to have SAF-rule controlled auditing without using SYS1.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)
```

If you use this method and intend to write audit records to SMF, the required SMF number is specified in the FMN2POPT module. See Customizing File Manager to write audit records to SMF on page 79 for more information.

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 193 for more information.
2. Write the relevant SAF rules. See the examples in SAF rule examples on page 200.
3. Activate SAF auditing for a specific logon using the chosen method of activating SAF-controlled auditing. See SAF-controlled auditing for File Manager Db2 component on page 189.
4. Using the selected logon, test the configuration to ensure that auditing is occurring as required.
5. When testing is complete, activate SAF-controlled auditing for all FM/Db2 users.

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 31: FM/Db2 auditing FACILITY class resource names on page 205 and Table 32: FM/Db2 auditing XFACILIT class resource names on page 205).

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 31: FM/Db2 auditing FACILITY class resource names on page 205). 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 (online 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 32: FM/Db2 auditing XFACILIT class resource names on page 205. 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 31: FM/Db2 auditing FACILITY class resource names on page 205).

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:

1. Step 1.
    - If auditing is to be controlled by means of parmlib, the FMNAUDIT specification of the FMN2PARM member is used as follows.

      The FMAUDIT specification setting in the FMN2PARM member (in SYS1.PARMLIB or any other library in the logical parmlib concatenation) 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 File Manager options specified in PARMLIB members on page 532 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 Customizing the audit facility for FM/Db2 on page 186.

      SAF_CTRL=YES

      SAF-rule controlled auditing is in effect. Processing continues to Step 2.
    - If auditing is being controlled using the method which does not access the logical parmlib concatenation, the TSO logonid has READ access to the SAF FACILITY rule FILEM.SAFAUDIT.DB2 for processing to continue to Step 2 on page 195.
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 31: FM/Db2 auditing FACILITY class resource names on page 205, the various outcomes are summarized in Table 27: Determination of a user's ability to write audit log records on page 195.

Table 27. 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

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

Note:

a. Refers to the level of access the user has to SAF FACILITY rule 1 in Table 31: FM/Db2 auditing FACILITY class resource names on page 205.

b. Refers to the level of access the user has to SAF FACILITY rule 2 in Table 31: FM/Db2 auditing FACILITY class resource names on page 205.

c. Refers to the level of access the user has to SAF FACILITY rule 3 in Table 31: FM/Db2 auditing FACILITY class resource names on page 205.

d. 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.

3. Step 3.

   Does the user have access to write audit records for the current function?

   The XFACILIT resource names 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 28: Types of SQL (and Db2) statements issued by FM/Db2 functions on page 196.

   The relationship between various SQL statements or Db2® commands and XFACILIT resource names is shown in Table 29: Relationship between SQL statement type and SAF resource names on page 198.

Table 28. 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

Table 28. Types of SQL (and Db2®) statements issued by FM/Db2 functions (continued)

FM/Db2 function	Option number	SQL/DB2 statements
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
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

Table 28. Types of SQL (and Db2®) statements issued by FM/Db2 functions (continued)

FM/Db2 function	Option number	SQL/DB2 statements
		-STOP -TERM

Table 29. 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]
ROLLBACK	OTHER.ADHOCSQL

SELECT	READ.<object>[2]
	UPDATE.<object>[3]
	READ.ADHOCSQL[4,6]
	UPDATE.ADHOCSQL[5,6]

SET	OTHER.ADHOCSQL
TRUNCATE	UPDATE.ADHOCSQL

Table 29. Relationship between SQL statement type and SAF resource names

(continued)

SQL Statement Type	Audit resource name suffix (see note 1)
UPDATE	UPDATE.ADHOCSQL

Note:

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 33: Resource name suffixes for Db2 object types (DDL SQL statements) on page 207.
8. See Table 34: Resource name suffixes for Db2 privileges (GRANT and REVOKE SQL statements) on page 208.

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.

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 30. 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 on page 200.

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 31: FM/Db2 auditing FACILITY class resource names on page 205 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 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 32: FM/Db2 auditing XFACILIT class resource names on page 205 shows the SAF XFACILIT 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 user IDs 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 user IDs 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 32: FM/Db2 auditing XFACILIT class resource names on page 205 shows the SAF XFACILIT 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 user IDs 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 user IDs 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. Activate SAF auditing for logonid TEST1 using the chosen method of activating SAF-controlled auditing. See SAF-controlled auditing for File Manager Db2 component on page 189.
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, activate auditing for all users using your chosen method.

FM/Db2 auditing FACILITY and XFACILIT class resource names

These two tables (and associated tables) list FACILITY and XFACILIT class resource names and details.

Table 31. 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.
Note: 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 32. 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).

Table 32. FM/Db2 auditing XFACILIT class resource names

(continued)

Rule Number	Resource Name suffix[1]	Purpose
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).
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.

Table 32. FM/Db2 auditing XFACILIT class resource names

(continued)

Rule Number	Resource Name suffix[1]	Purpose
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).

Note:

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 33: Resource name suffixes for Db2 object types (DDL SQL statements) on page 207.
3. See Table 34: Resource name suffixes for Db2 privileges (GRANT and REVOKE SQL statements) on page 208.
4. See Table 35: Resource name suffixes for Db2 commands on page 208.

Table 33. 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
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

Table 33. Resource name suffixes for Db2® object types (DDL SQL statements) (continued)

Db2® Object Type	Resource Rule Name suffix
TYPE	TYPE
VIEW	VIEW

Table 34. 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 35. 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
MODIFY	MODIFY

Table 35. Resource name suffixes for Db2® commands (continued)

Db2® Command	Resource Rule Name suffix
REBIND	REBIND
RECOVER	RECOVER
REFRESH	REFRESH
RESET	RESET
RUN	RUN
SET	SET
START	START
STOP	STOP
TERM	TERM

Chapter 18. 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, you will need to perform the customization tasks listed in Table 36: Summary of steps for customizing FM/Db2 for a national language on page 210. Perform **ALL** these steps (except Step 9).

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 36. Summary of steps for customizing FM/Db2 for a national language

Step	Description
__ 1	Set the LANGUAGE option for batch processing, if required. See Setting the default national language on page 179.
__ 2	Change the TERMTYPE option in FMN2POPT, if you are using a DBCS language. See TERMTYPE on page 423.
__ 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. See Changing the print and display translation tables for languages other than English on page 210.
__ 5	Translate the File Manager message text to your language. See Translating the message text on page 211.
__ 6	Provide a version of FMN2MENU for your language. See Providing a multicultural version of FMN2MENU on page 211.
__ 7	Translate the FM/Db2 ISPF messages to your language. See Translating the ISPF messages text on page 212.
__ 8	Translate the FM/Db2 panels to your language. See Translating the panel text on page 213.
__ 9	Translate the FM/Db2 object list utility headings and statements (FMN2DENU) to your language. See Providing a multicultural version of FMN2DENU on page 214.

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

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 98.

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.

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

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 213.

To provide translated versions of the messages, you must provide a version of FMN2MENU for your language, as described in Providing a multicultural version of FMN2MENU on page 211, **and** provide translated versions of the appropriate members in FMN.SFMNMENU, as described in Translating the ISPF messages text on page 212.

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 multicultural 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 on page 213. 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 ISPF 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 on page 213.

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 37: Keyword values for the LANGUAGE option on page 214. See Changing the default options on page 178 for information on how to do this.

Table 37. 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 multicultural 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'FMN2Dyyy'
     ```

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 FMNUMDD 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 FMNUMDD.

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 103.

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 104.

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 19. Verifying the customization of FM/Db2

After you have completed all the necessary steps to install and customize FM/Db2, you can perform the following steps to verify your customization.

1. Use the FM/Db2 *“Edit/Execute SQL (Data Set)”* function to execute sample DDL. This creates a small database that is used in subsequent IVP steps.
2. Run a batch job to execute the Db2® CHECK DATA utility against the Db2® objects in the FM/Db2 IVP database.
3. Use the FM/Db2 *“Db2® Utilities”* function to generate and run a job to take an image copy of the Db2® objects in the FM/Db2 IVP database.
4. Use the FM/Db2 editor to display and modify data in the FM/Db2 IVP tables.
5. Use the FM/Db2 *“Basic SELECT Prototyping”* function to create and execute a simple join of the tables in the FM/Db2 IVP database.
6. Use the FM/Db2 *“Copy utility”* function to copy data from one IVP table to another.
7. Use the FM/Db2 *“Export utility”* function to copy the data in one of the FM/Db2 IVP tables to a sequential data set.
8. Use the FM/Db2 *“Import utility”* function to copy the sequential data set (created in step 7) into an FM/Db2 IVP table.
9. Use the FM/Db2 *“Object list utility”* function to display information about the FM/Db2 IVP database and objects.

To review the customization and verification steps, refer to:

- Checklist for installing and customizing FM/Db2 on page 151
- *File Manager User’s Guide and Reference for DB2 Data*.

Step 1. Define Db2® objects to be used during verification

The first step in the FM/Db2 verification is to define the Db2® objects to be used.

Take a copy of the IVP member from the sample library. Select the sample member that matches the Db2® version of the Db2® system where the IVP database is to be created. For example:

```
FMN2VERA  (for Db2 version 10 systems)
```

Note: The figures in this chapter are from a Db2® version 10 system. If you run the IVP against a later Db2® version there might be minor differences in the names of the Db2® objects.

Follow the instructions at the top of the sample. You need to review, and possibly change, the following values:

SG name

Change this to the name of a Db2® storage group of your choice. You can use an existing Db2® storage group name. To do this, change *SG_name* to the existing storage group name, and comment out the

CREATE STOGROUP statement. If you use an existing Db2® storage group name, you can skip the customization for *Vol_list* and *DB2_VCAT_name*.

VOL_list

Change this, if required, to a list of valid disk volume names.

DB2_VCAT_name

Change this to the VCAT name used for Db2® user data sets, for the appropriate Db2® system.

To execute the DDL statements:

1. Log on to a TSO user ID that is enabled to access and use FM/Db2.
2. 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 157), type this option value, and press Enter. For example, if you have assigned FD to FM/Db2, type FD and then press Enter.
    - If you defined FM/Db2 in an ISPF command table (see Defining FM/Db2 in an ISPF command table on page 157), verify that FM/Db2 can be started by entering the command FD on any ISPF command line.
3. If you have previously selected a Db2® subsystem, you immediately see the Primary Option Menu.

    If this is the first time you have used FM/Db2:

    - The Copyright panel appears. After reading the panel text, press the Cancel key, PF12. In subsequent sessions this panel does not automatically appear.
    - The **Db2 Subsystem Selection** menu is displayed, showing the Db2® subsystems you customized in your FMN2SSDM macro. Select the Db2® subsystem you want to use, or run the FM/Db2 IVP against, and press Enter to go to the FM/Db2 Primary Option Menu.
4. Enter VER on the command line to display the release level and PTF level of FM/Db2. A panel is displayed that gives you, as example, the following information:

    ```
    IBM File Manager for z/OS Version 14 Release 1 Modification 4
    Db2 Component
    (not APF authorized)

    Service Levels of installed components

                 Base       IMS        Db2        CICS
    English      -NONE-     -NONE-     -NONE-     -NONE-
    ```

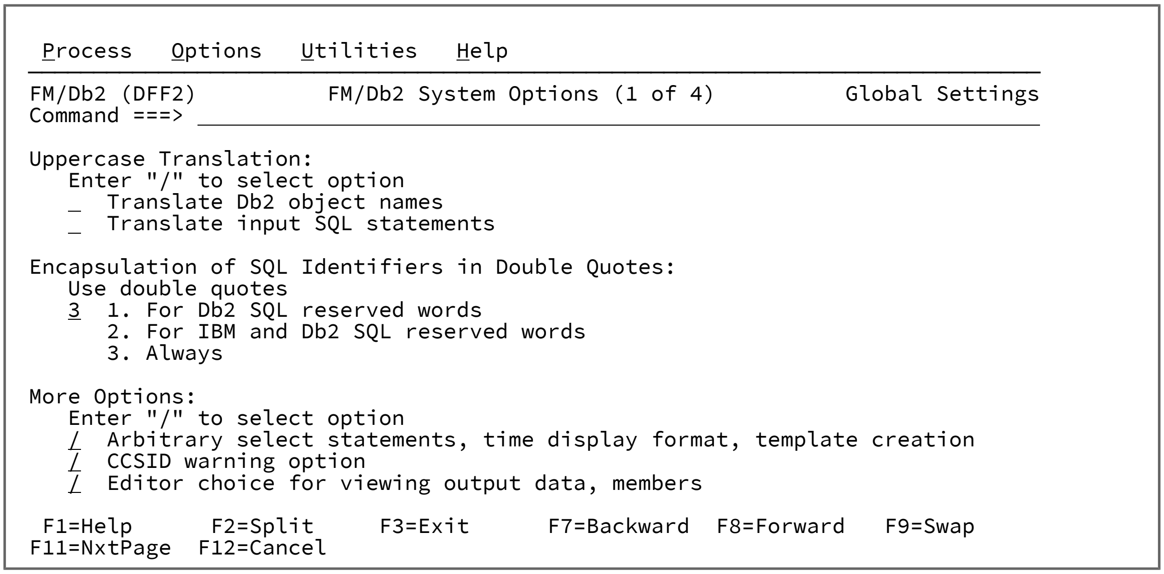
    - FM/Db2 is always shown as `APF not authorized`, even if you have made File Manager APF-authorized, as File Manager cannot run as APF-authorized under ISPF.
    - When you first install File Manager, `-NONE-` is shown against each component. Subsequently, when you have applied service to File Manager, a PTF number is shown, indicating the PTF level of each component you have installed. If you have not installed a component, that component is not be shown at all.

        If you have installed the Japanese, or Korean, language components of FM/Db2, another line is displayed indicating the service level of that component.
5. Select the FM/Db2 system options by typing '0.0.2' on the FM/Db2 main menu panel and pressing Enter. See Figure 17: FM/Db2 System Options panel on page 220. Ensure that:

- The *“Translate Db2® object names”* option is **not selected**.
- The *“Translate input SQL statements”* option is **not selected**.

6. Press PF3 to return to the FM/Db2 main menu.

   Figure 17. FM/Db2 System Options panel



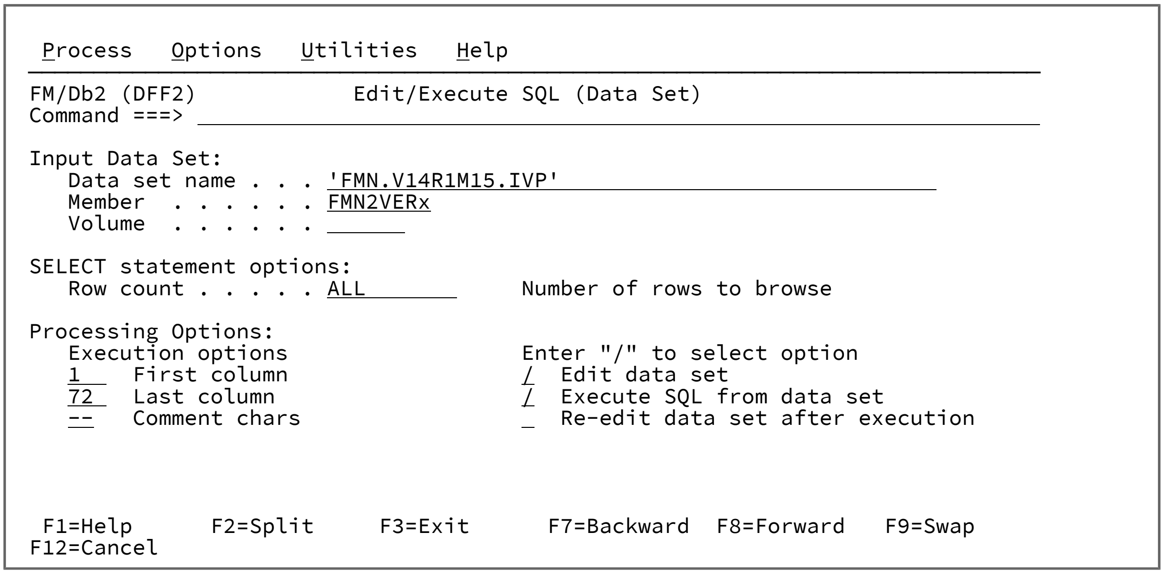
```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)          FM/Db2 System Options (1 of 4)          Global Settings
Command ===> _________________________________________________________________

Uppercase Translation:
   Enter "/" to select option
   _  Translate Db2 object names
   _  Translate input SQL statements

Encapsulation of SQL Identifiers in Double Quotes:
   Use double quotes
   3  1. For Db2 SQL reserved words
      2. For IBM and Db2 SQL reserved words
      3. Always

More Options:
   Enter "/" to select option
   /  Arbitrary select statements, time display format, template creation
   /  CCSID warning option
   /  Editor choice for viewing output data, members

 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F11=NxtPage  F12=Cancel
```

7. Select the *“Edit/Execute SQL (Data Set)”* function by typing '4.4' on the FM/Db2 main menu panel and pressing Enter.

   See Figure 18: Edit/Execute SQL (Data Set) panel on page 220.

   Enter a data set name, member name, and execution options, as shown in Figure 18: Edit/Execute SQL (Data Set) panel on page 220. Modify the data set and member names to reflect the data set and member name containing your modified copy of the FMN2VERx sample.

   Figure 18. Edit/Execute SQL (Data Set) panel

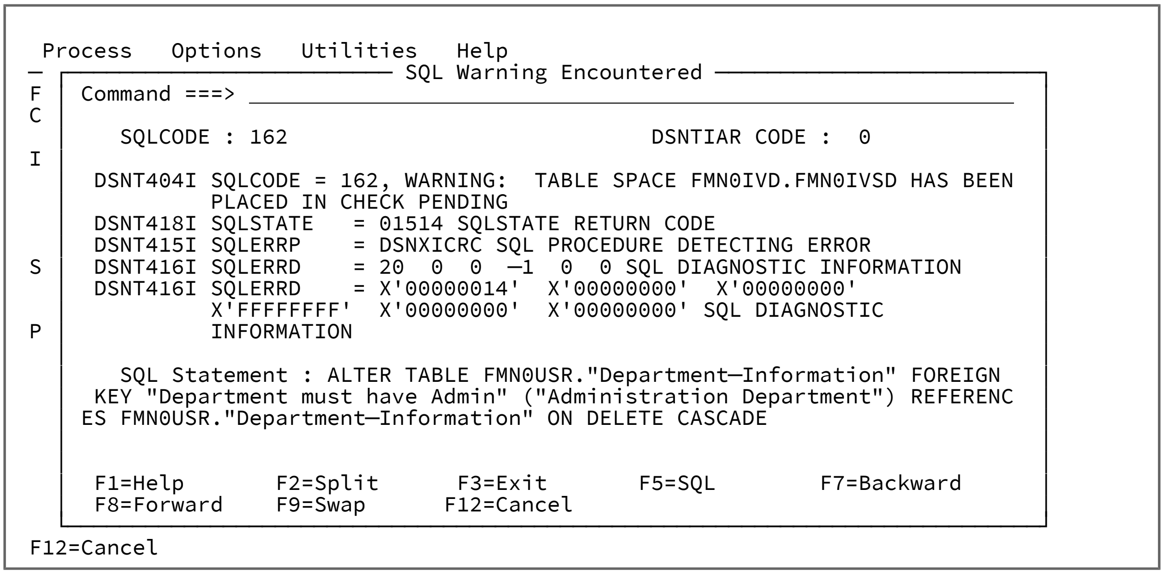


```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)            Edit/Execute SQL (Data Set)
Command ===> _________________________________________________________________

Input Data Set:
   Data set name . . . 'FMN.V14R1M15.IVP'_______________________________
   Member  . . . . . . FMN2VERx
   Volume  . . . . . . ______

SELECT statement options:
   Row count . . . . . ALL______      Number of rows to browse

Processing Options:
   Execution options                  Enter "/" to select option
   1__  First column                  /  Edit data set
   72_  Last column                   /  Execute SQL from data set
   --   Comment chars                 _  Re-edit data set after execution



 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

8. Press Enter to edit the sample DDL. When you have made any required changes, press PF3 to run the DDL statements.

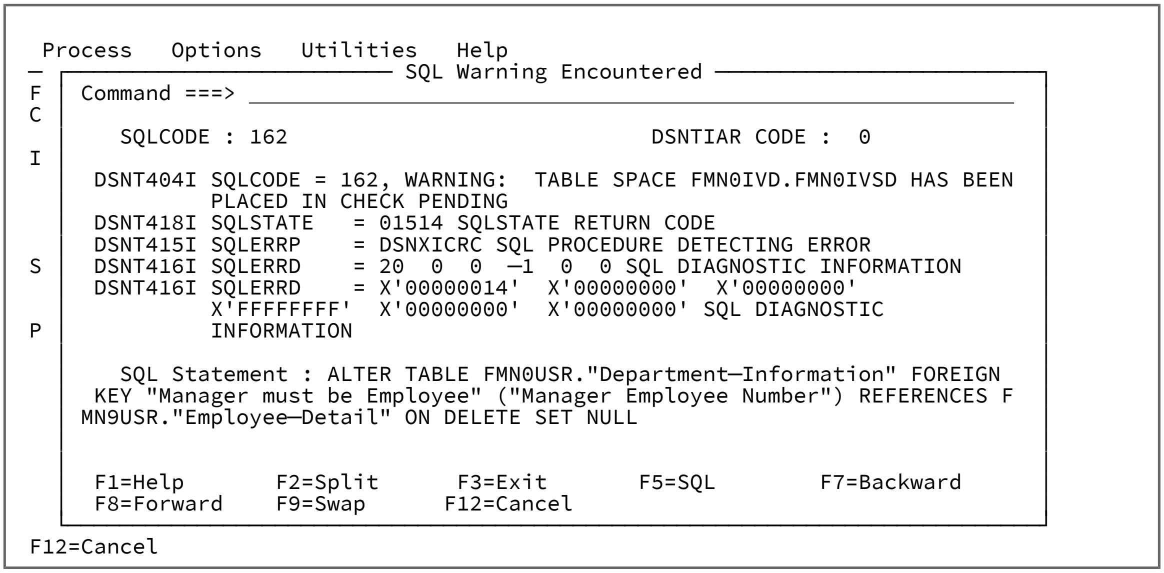
When the DDL samples run successfully, two SQL warning messages are displayed. See Figure 19: SQL Warning (1) panel on page 221 and Figure 20: SQL Warning (2) panel on page 221. These warnings are expected. To continue running, press Enter.

Figure 19. SQL Warning (1) panel



```
 Process   Options   Utilities   Help
—                          SQL Warning Encountered
F   Command ===> ____________________________________________________________
C
       SQLCODE : 162                            DSNTIAR CODE :  0
I
     DSNT404I SQLCODE = 162, WARNING:  TABLE SPACE FMN0IVD.FMN0IVSD HAS BEEN
              PLACED IN CHECK PENDING
     DSNT418I SQLSTATE   = 01514 SQLSTATE RETURN CODE
     DSNT415I SQLERRP    = DSNXICRC SQL PROCEDURE DETECTING ERROR
S    DSNT416I SQLERRD    = 20  0  0  —1  0  0 SQL DIAGNOSTIC INFORMATION
     DSNT416I SQLERRD    = X'00000014'  X'00000000'  X'00000000'
              X'FFFFFFFF'  X'00000000'  X'00000000' SQL DIAGNOSTIC
P             INFORMATION

       SQL Statement : ALTER TABLE FMN0USR."Department—Information" FOREIGN
     KEY "Department must have Admin" ("Administration Department") REFERENC
    ES FMN0USR."Department—Information" ON DELETE CASCADE


     F1=Help       F2=Split      F3=Exit       F5=SQL        F7=Backward
     F8=Forward    F9=Swap      F12=Cancel

F12=Cancel
```

Figure 20. SQL Warning (2) panel



```
 Process   Options   Utilities   Help
—                          SQL Warning Encountered
F   Command ===> ____________________________________________________________
C
       SQLCODE : 162                            DSNTIAR CODE :  0
I
     DSNT404I SQLCODE = 162, WARNING:  TABLE SPACE FMN0IVD.FMN0IVSD HAS BEEN
              PLACED IN CHECK PENDING
     DSNT418I SQLSTATE   = 01514 SQLSTATE RETURN CODE
     DSNT415I SQLERRP    = DSNXICRC SQL PROCEDURE DETECTING ERROR
S    DSNT416I SQLERRD    = 20  0  0  —1  0  0 SQL DIAGNOSTIC INFORMATION
     DSNT416I SQLERRD    = X'00000014'  X'00000000'  X'00000000'
              X'FFFFFFFF'  X'00000000'  X'00000000' SQL DIAGNOSTIC
P             INFORMATION

       SQL Statement : ALTER TABLE FMN0USR."Department—Information" FOREIGN
     KEY "Manager must be Employee" ("Manager Employee Number") REFERENCES F
    MN9USR."Employee—Detail" ON DELETE SET NULL


     F1=Help       F2=Split      F3=Exit       F5=SQL        F7=Backward
     F8=Forward    F9=Swap      F12=Cancel

F12=Cancel
```

When the SQL has run successfully, a box appears at the bottom of the screen with *"173 statements executed"*.

Move onto Step 2. Run the FMN2CHKx sample job on page 222 of the IVP.

Step 2. Run the FMN2CHKx sample job

The Db2® objects created in Step 1. Define Db2 objects to be used during verification on page 218 have referential constraints defined. Before these objects can be accessed, you need to run the Db2® CHECK DATA utility against the objects to ensure that the data does not violate any data constraints.

Take a copy of the *"check data"* member from the sample library. Select the sample member that matches the Db2® version of the Db2® system where the IVP database is to be created. For example:

```
FMN2CHKA  (for Db2 version 10 systems)
```

Follow the instructions at the top of the sample. You need to review and possibly change the following values:

Job card

Specify a valid job card.

sss

Change *ssss* to the appropriate value. This will be the same value you selected when you connected to FM/Db2.

Db2 library name

Change the *Db2 library name* to the name of the Db2® load library for the Db2® system.

SORTLIB

Review, and possibly change, the name of the library used for DFSORT, or equivalent product.

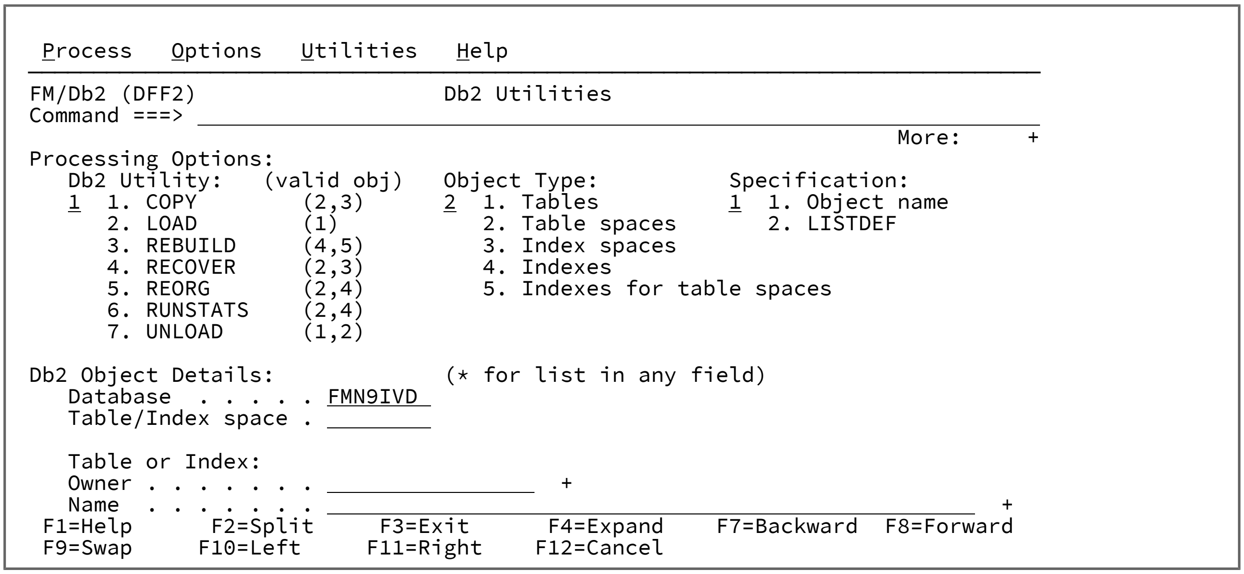
Submit the job. The expected return code is 0.

Step 3. Take a copy of the FM/Db2 IVP tables

Log on to FM/Db2 and connect to the same Db2® system specified in Step 1. Define Db2 objects to be used during verification on page 218. Select the *"Db2® Utilities"* function by typing '3.9' on the FM/Db2 main menu and press Enter. See Figure 21: Db2 Utilities panel on page 223.

Enter the name of the FM/Db2 IVP database in the Database field, select the COPY utility and Table spaces. Press Enter.

Figure 21. Db2® Utilities panel

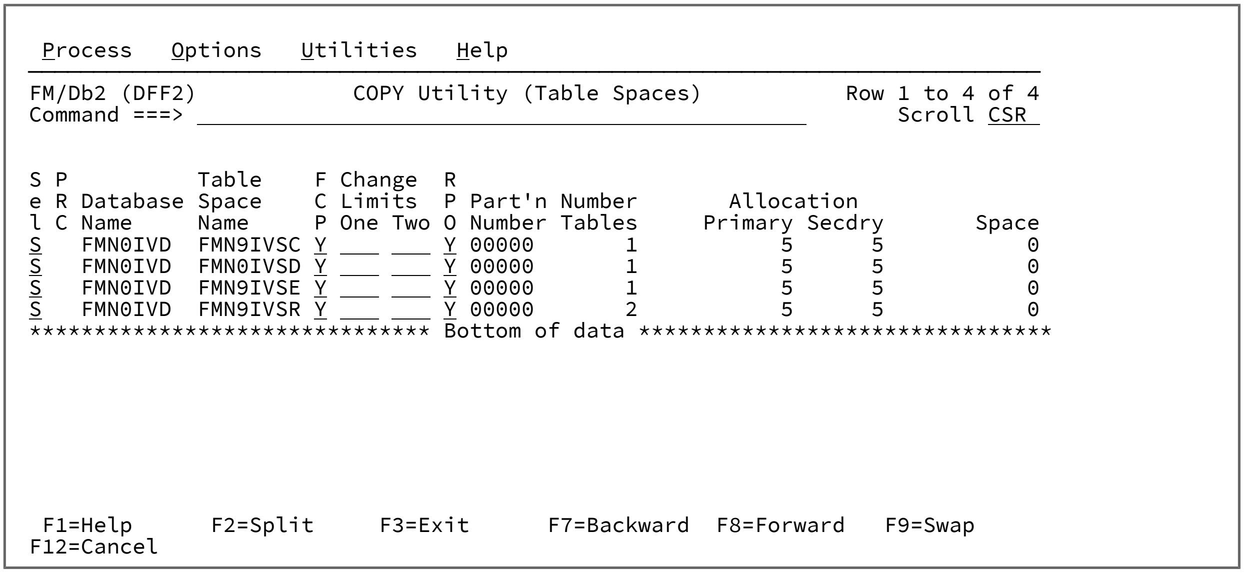


```
 Process   Options   Utilities   Help
 ──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                  Db2 Utilities
Command ===> _________________________________________________________________
                                                                  More:     +
Processing Options:
   Db2 Utility:   (valid obj)  Object Type:          Specification:
   1  1. COPY        (2,3)     2  1. Tables           1  1. Object name
      2. LOAD        (1)          2. Table spaces        2. LISTDEF
      3. REBUILD     (4,5)        3. Index spaces
      4. RECOVER     (2,3)        4. Indexes
      5. REORG       (2,4)        5. Indexes for table spaces
      6. RUNSTATS    (2,4)
      7. UNLOAD      (1,2)

Db2 Object Details:            (* for list in any field)
   Database  . . . . . FMN9IVD
   Table/Index space . ________

   Table or Index:
   Owner . . . . . . . ________________  +
   Name  . . . . . . . __________________________________________________  +
 F1=Help      F2=Split     F3=Exit      F4=Expand     F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

See Figure 22: COPY Utility panel (Table Spaces) on page 223.

This display shows the table spaces defined in the FM/Db2 IVP data base. Type *"s"* against every entry and press Enter.

Figure 22. COPY Utility panel (Table Spaces)



```
 Process   Options   Utilities   Help
 ──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)           COPY Utility (Table Spaces)           Row 1 to 4 of 4
Command ===> _______________________________________________       Scroll CSR


S P          Table     F Change   R
e R Database Space     C Limits   P Part'n Number        Allocation
l C Name     Name      P One Two  O Number Tables      Primary Secdry        Space
S   FMN0IVD  FMN9IVSC  Y ___ ___  Y 00000       1            5      5            0
S   FMN0IVD  FMN0IVSD  Y ___ ___  Y 00000       1            5      5            0
S   FMN0IVD  FMN9IVSE  Y ___ ___  Y 00000       1            5      5            0
S   FMN0IVD  FMN9IVSR  Y ___ ___  Y 00000       2            5      5            0
******************************* Bottom of data ********************************







 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

FM/Db2 displays an ISPF edit session with a job that takes an image copy of the selected table spaces. Check the following carefully:

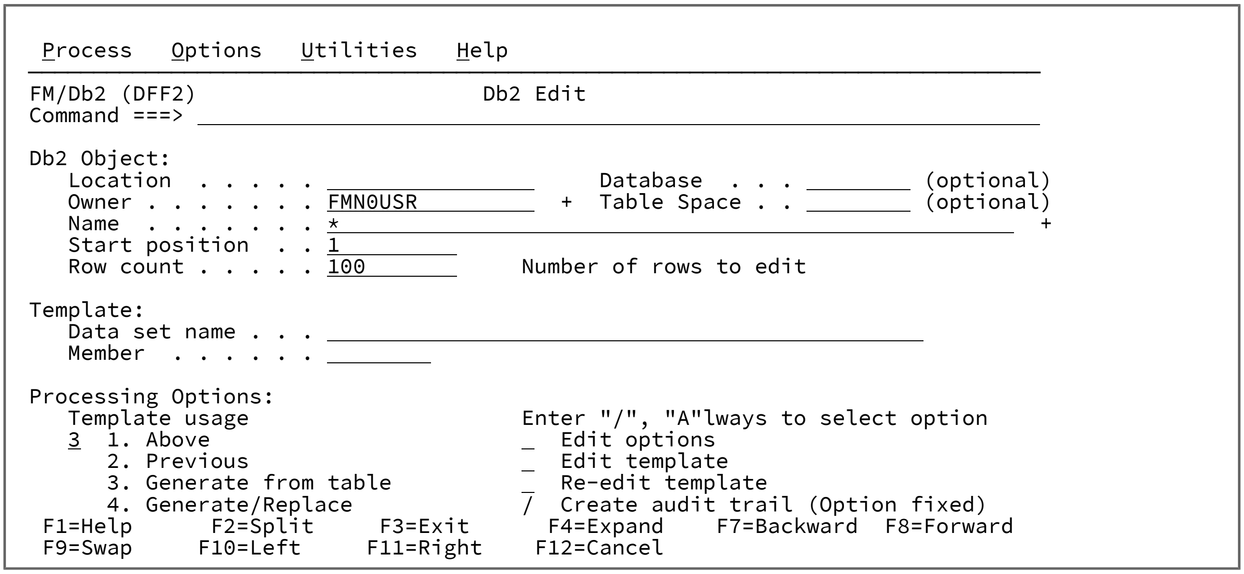
- Job card.
- DSNUPROC.STEPLIB libraries. The values for these libraries are those specified in the FMN2SSDM macro entry for the connected Db2® system in the FMN2POPT. For more information, see Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.

Submit the job. The expected return code is RC=0.

Step 4. Use the FM/Db2 editor (normal and related edit)

Log on to FM/Db2, and connect to the same Db2® system specified in Step 1. Define Db2 objects to be used during verification on page 218. Select the FM/Db2 editor by typing '2' on the FM/Db2 main menu. Press Enter.

Figure 23. Db2® Edit panel

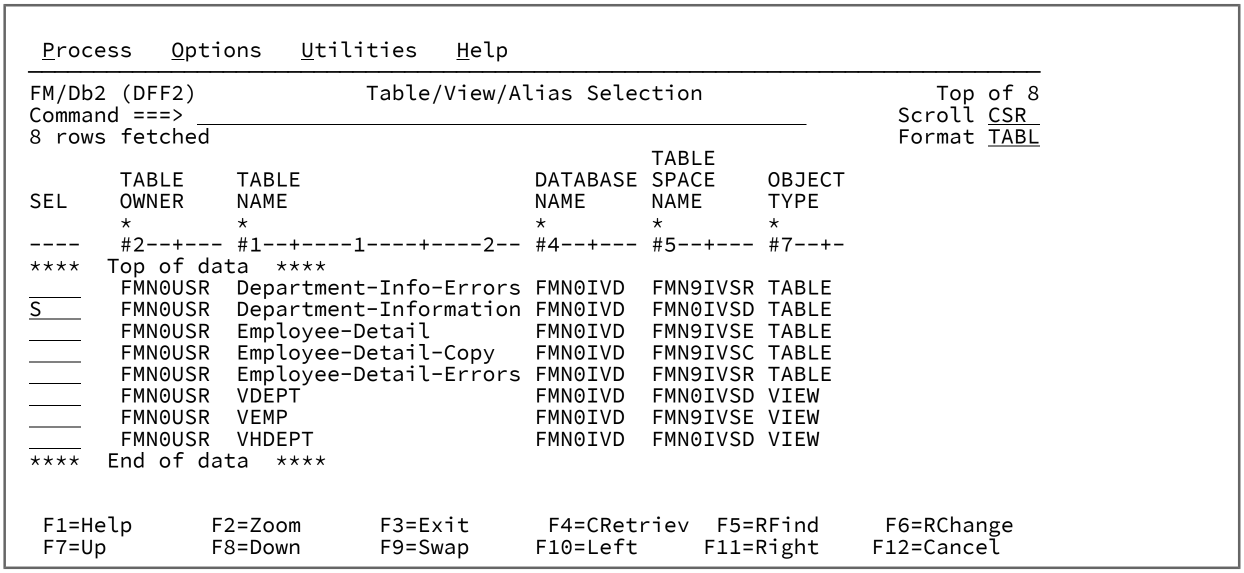


```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                      Db2 Edit
Command ===> _________________________________________________________________

Db2 Object:
   Location  . . . . . ________________     Database  . . . ________ (optional)
   Owner . . . . . . . FMN0USR_________  +  Table Space . . ________ (optional)
   Name  . . . . . . . *____________________________________________________ +
   Start position  . . 1_________
   Row count . . . . . 100_______      Number of rows to edit

Template:
   Data set name . . . ______________________________________________
   Member  . . . . . . ________

Processing Options:
   Template usage                      Enter "/", "A"lways to select option
   3  1. Above                         _  Edit options
      2. Previous                      _  Edit template
      3. Generate from table           _  Re-edit template
      4. Generate/Replace              /  Create audit trail (Option fixed)
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

See Figure 23: Db2 Edit panel on page 224. Enter the values shown in the figure, FMN0USR in the *“Owner”* field and '*' in the *“Name”* field.

Press Enter to display the *“Table/View/Alias Selection”* panel. See Figure 24: Table/View/Alias Selection panel on page 224.

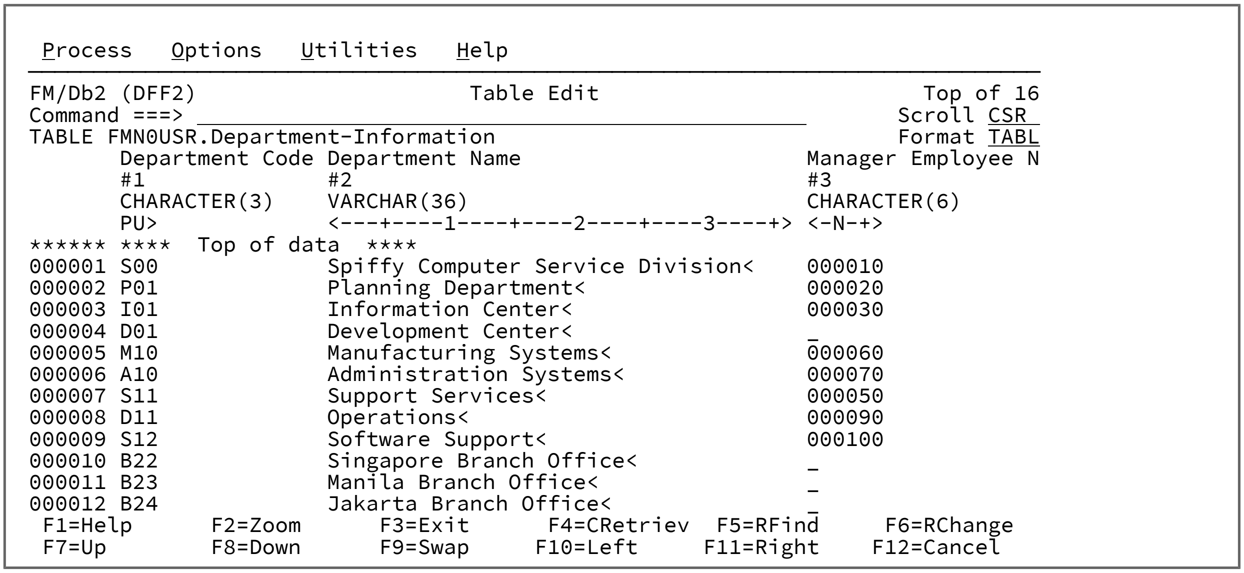
Figure 24. Table/View/Alias Selection panel



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)           Table/View/Alias Selection                   Top of 8
Command ===> ______________________________________________         Scroll CSR
8 rows fetched                                                      Format TABL
                                                  TABLE
       TABLE    TABLE                    DATABASE SPACE    OBJECT
SEL    OWNER    NAME                     NAME     NAME     TYPE
       *        *                        *        *        *
----   #2--+--- #1--+----1----+----2-- #4--+--- #5--+--- #7--+-
****  Top of data  ****
____   FMN0USR  Department-Info-Errors FMN0IVD  FMN9IVSR TABLE
S___   FMN0USR  Department-Information FMN0IVD  FMN0IVSD TABLE
____   FMN0USR  Employee-Detail        FMN0IVD  FMN9IVSE TABLE
____   FMN0USR  Employee-Detail-Copy   FMN0IVD  FMN9IVSC TABLE
____   FMN0USR  Employee-Detail-Errors FMN0IVD  FMN9IVSR TABLE
____   FMN0USR  VDEPT                  FMN0IVD  FMN0IVSD VIEW
____   FMN0USR  VEMP                   FMN0IVD  FMN9IVSE VIEW
____   FMN0USR  VHDEPT                 FMN0IVD  FMN0IVSD VIEW
****  End of data  ****


 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Type 'S' against the *“Department-Information”* field in the table and press Enter. See Figure 25: Table Edit panel on page 225.

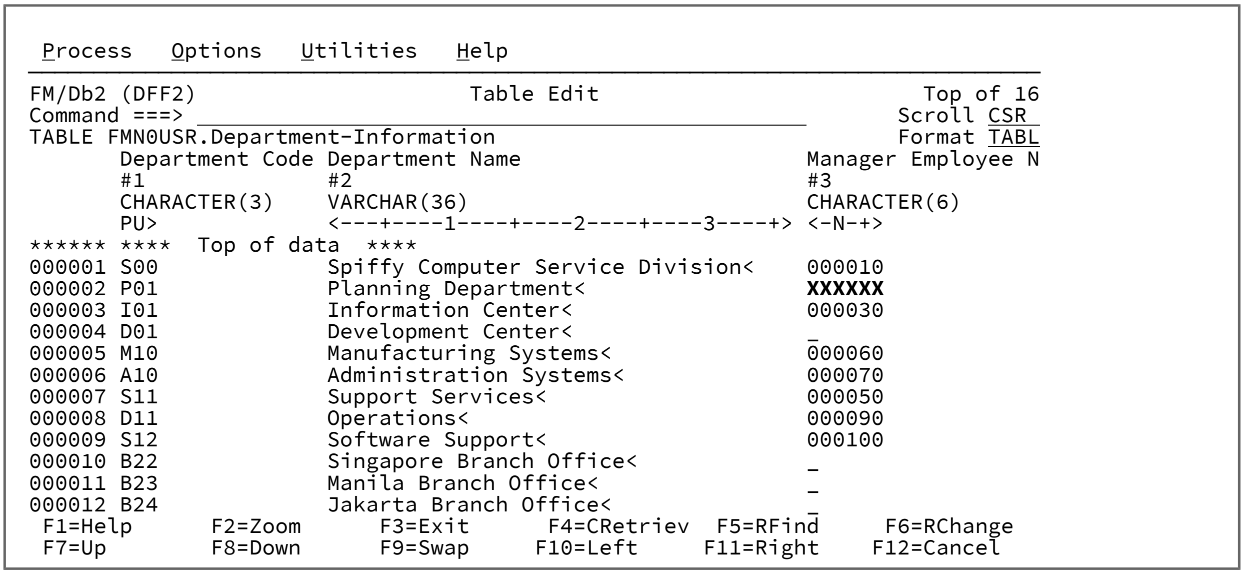
Figure 25. Table Edit panel



```
 Process   Options   Utilities   Help
 ─────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                     Table Edit                         Top of 16
Command ===> _______________________________________________       Scroll CSR_
TABLE FMN0USR.Department-Information                               Format TABL
       Department Code Department Name                      Manager Employee N
       #1              #2                                   #3
       CHARACTER(3)    VARCHAR(36)                          CHARACTER(6)
       PU>             <---+----1----+----2----+----3----+> <-N-+>
****** ****  Top of data  ****
000001 S00             Spiffy Computer Service Division<    000010
000002 P01             Planning Department<                 000020
000003 I01             Information Center<                  000030
000004 D01             Development Center<                  _
000005 M10             Manufacturing Systems<               000060
000006 A10             Administration Systems<              000070
000007 S11             Support Services<                    000050
000008 D11             Operations<                          000090
000009 S12             Software Support<                    000100
000010 B22             Singapore Branch Office<             _
000011 B23             Manila Branch Office<                _
000012 B24             Jakarta Branch Office<               _
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

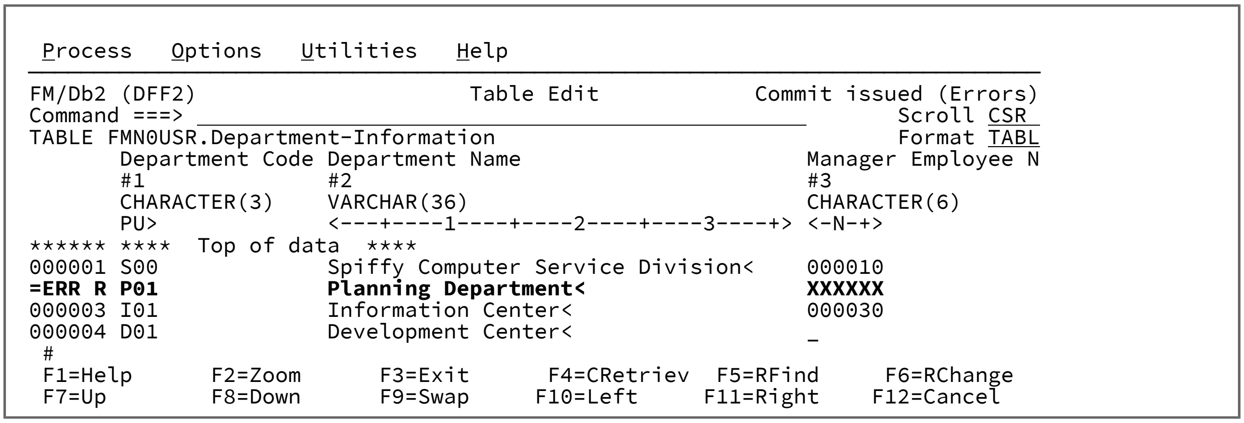
Type over the *“Manager Employee Number”* value in the second row, as shown in Figure 26: Table Edit panel (showing typeover Manager Employee number) on page 225. Change the value to 'XXXXXX' and press PF3. See Figure 27: Table Edit panel (showing amended Manager Employee number) on page 226 to see the FM/Db2 response when the PF3 (EXIT) key is pressed.

Figure 26. Table Edit panel (showing typeover Manager Employee number)



```
 Process   Options   Utilities   Help
 ─────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                     Table Edit                         Top of 16
Command ===> _______________________________________________       Scroll CSR_
TABLE FMN0USR.Department-Information                               Format TABL
       Department Code Department Name                      Manager Employee N
       #1              #2                                   #3
       CHARACTER(3)    VARCHAR(36)                          CHARACTER(6)
       PU>             <---+----1----+----2----+----3----+> <-N-+>
****** ****  Top of data  ****
000001 S00             Spiffy Computer Service Division<    000010
000002 P01             Planning Department<                 XXXXXX
000003 I01             Information Center<                  000030
000004 D01             Development Center<                  _
000005 M10             Manufacturing Systems<               000060
000006 A10             Administration Systems<              000070
000007 S11             Support Services<                    000050
000008 D11             Operations<                          000090
000009 S12             Software Support<                    000100
000010 B22             Singapore Branch Office<             _
000011 B23             Manila Branch Office<                _
000012 B24             Jakarta Branch Office<               _
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 27. Table Edit panel (showing amended Manager Employee number)

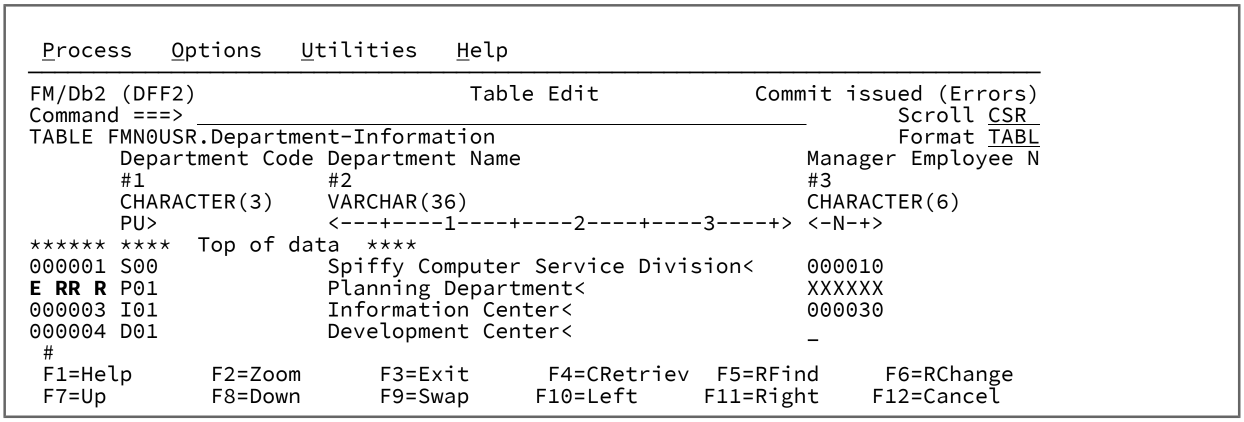


```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                   Table Edit           Commit issued (Errors)
Command ===> _______________________________________________ Scroll CSR 
TABLE FMN0USR.Department-Information                         Format TABL
       Department Code Department Name                     Manager Employee N
       #1              #2                                  #3
       CHARACTER(3)    VARCHAR(36)                         CHARACTER(6)
       PU>             <---+----1----+----2----+----3----+> <-N-+>
****** ****  Top of data  ****
000001 S00             Spiffy Computer Service Division<    000010
=ERR R P01             Planning Department<                 XXXXXX
000003 I01             Information Center<                  000030
000004 D01             Development Center<                  _
 #
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

The change to the *“Manager Employee Number”* could not be saved. The row in error is marked with `'=ERR R'`.

Overtype 'e' in the prefix area for this row and press Enter. See Figure 28: Table Edit with amended Manager Employee number on page 226.

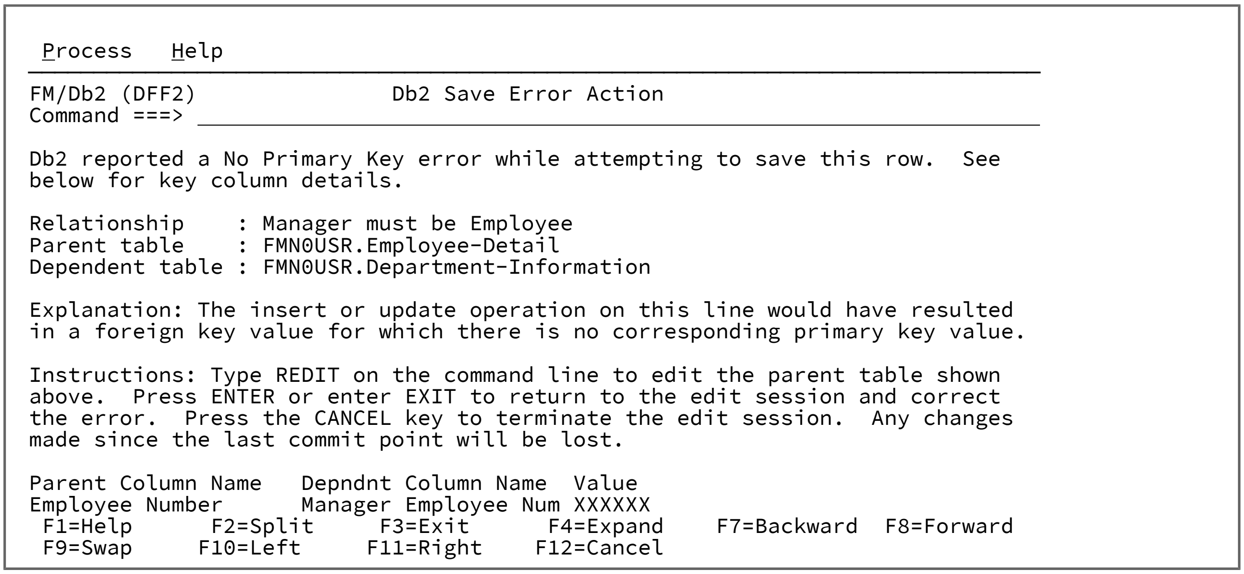
Figure 28. Table Edit with amended Manager Employee number



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                   Table Edit           Commit issued (Errors)
Command ===> _______________________________________________ Scroll CSR 
TABLE FMN0USR.Department-Information                         Format TABL
       Department Code Department Name                     Manager Employee N
       #1              #2                                  #3
       CHARACTER(3)    VARCHAR(36)                         CHARACTER(6)
       PU>             <---+----1----+----2----+----3----+> <-N-+>
****** ****  Top of data  ****
000001 S00             Spiffy Computer Service Division<    000010
E RR R P01             Planning Department<                 XXXXXX
000003 I01             Information Center<                  000030
000004 D01             Development Center<                  _
 #
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

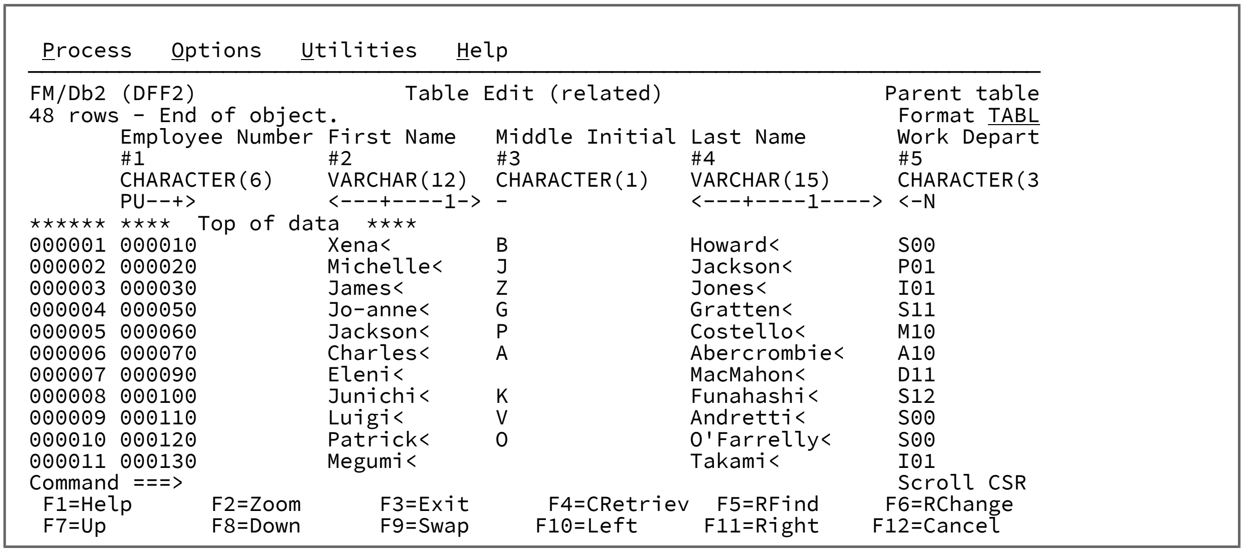
FM/Db2 displays the error panel shown in Figure 29: Db2 Save Error Action panel on page 227. Type the REDIT command on the command line, and press Enter. See Figure 30: Table Edit (related) panel on page 227 to display the parent table for the relationship.

Figure 29. Db2® Save Error Action panel



```
 Process   Help
FM/Db2 (DFF2)               Db2 Save Error Action
Command ===>

Db2 reported a No Primary Key error while attempting to save this row.  See
below for key column details.

Relationship    : Manager must be Employee
Parent table    : FMN0USR.Employee-Detail
Dependent table : FMN0USR.Department-Information

Explanation: The insert or update operation on this line would have resulted
in a foreign key value for which there is no corresponding primary key value.

Instructions: Type REDIT on the command line to edit the parent table shown
above.  Press ENTER or enter EXIT to return to the edit session and correct
the error.  Press the CANCEL key to terminate the edit session.  Any changes
made since the last commit point will be lost.

Parent Column Name   Depndnt Column Name  Value
Employee Number      Manager Employee Num XXXXXX
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 30. Table Edit (related) panel



```
 Process   Options   Utilities   Help
FM/Db2 (DFF2)                Table Edit (related)                 Parent table
48 rows - End of object.                                           Format TABL
       Employee Number First Name   Middle Initial Last Name       Work Depart
       #1              #2           #3             #4              #5
       CHARACTER(6)    VARCHAR(12)  CHARACTER(1)   VARCHAR(15)     CHARACTER(3
       PU--+>          <---+----1-> -              <---+----1----> <-N
****** ****  Top of data  ****
000001 000010          Xena<        B              Howard<         S00
000002 000020          Michelle<    J              Jackson<        P01
000003 000030          James<       Z              Jones<          I01
000004 000050          Jo-anne<     G              Gratten<        S11
000005 000060          Jackson<     P              Costello<       M10
000006 000070          Charles<     A              Abercrombie<    A10
000007 000090          Eleni<                      MacMahon<       D11
000008 000100          Junichi<     K              Funahashi<      S12
000009 000110          Luigi<       V              Andretti<       S00
000010 000120          Patrick<     O              O'Farrelly<     S00
000011 000130          Megumi<                     Takami<         I01
Command ===>                                                       Scroll CSR
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

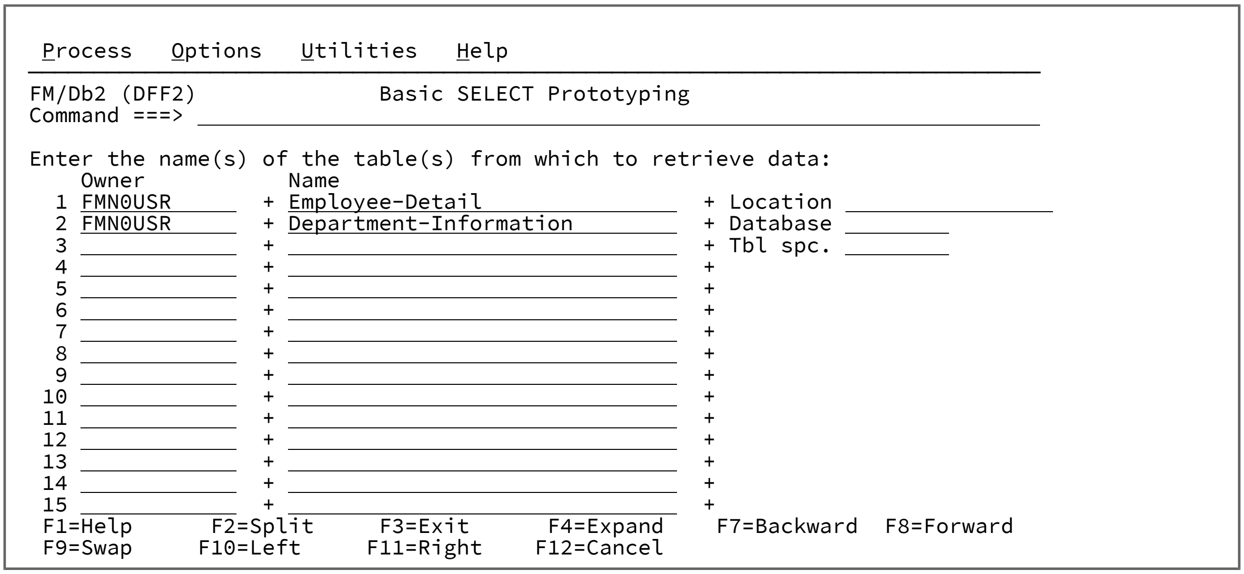
Press PF3 to exit from the FM/Db2 edit session of the *“Employee-Detail”* table and return to the FM/Db2 edit session of the *“Department-Information”* table. Press PF12 (CANCEL) to return to the FM/Db2 Editor function entry panel. A pop-up panel is displayed asking for confirmation to cancel the operation. Press Enter to continue.

Step 5. Use the FM/Db2 Basic SELECT Prototyper

Log on to FM/Db2, and connect to the same Db2® system specified in Step 1. Define Db2 objects to be used during verification on page 218. From the FM/Db2 main menu enter `4.1` to display the *“Basic SELECT Prototyping”* function.

Enter the names of the two FM/Db2 IVP tables as shown in Figure 31: Basic SELECT Prototyping panel on page 228.

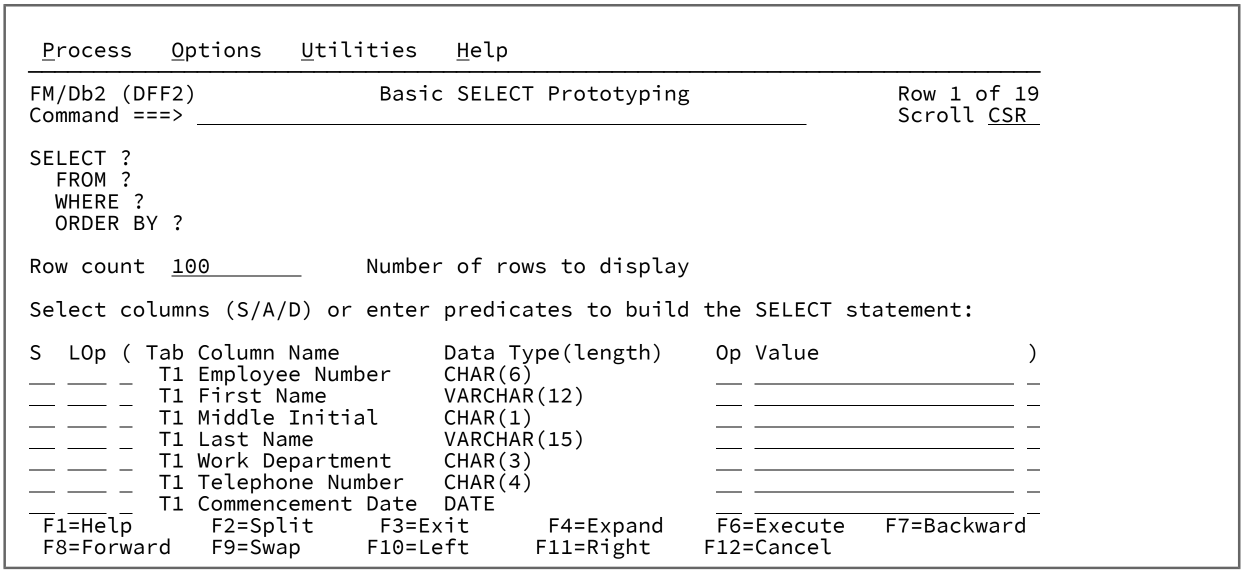
Figure 31. Basic SELECT Prototyping panel



```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
FM/Db2 (DFF2)              Basic SELECT Prototyping
Command ===> _________________________________________________________________

Enter the name(s) of the table(s) from which to retrieve data:
    Owner           Name
  1 FMN0USR_____  + Employee-Detail_______________  + Location ________________
  2 FMN0USR_____  + Department-Information________  + Database ________
  3 ____________  + ______________________________  + Tbl spc. ________
  4 ____________  + ______________________________  +
  5 ____________  + ______________________________  +
  6 ____________  + ______________________________  +
  7 ____________  + ______________________________  +
  8 ____________  + ______________________________  +
  9 ____________  + ______________________________  +
 10 ____________  + ______________________________  +
 11 ____________  + ______________________________  +
 12 ____________  + ______________________________  +
 13 ____________  + ______________________________  +
 14 ____________  + ______________________________  +
 15 ____________  + ______________________________  +
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Press Enter to display the second *“Basic SELECT Prototyping”* panel. See Figure 32: Basic SELECT Prototyping panel (2) on page 228.

Figure 32. Basic SELECT Prototyping panel (2)



```
 Process   Options   Utilities   Help
------------------------------------------------------------------------------
FM/Db2 (DFF2)              Basic SELECT Prototyping                Row 1 of 19
Command ===> _______________________________________________       Scroll CSR_

SELECT ?
  FROM ?
  WHERE ?
  ORDER BY ?

Row count  100_______     Number of rows to display

Select columns (S/A/D) or enter predicates to build the SELECT statement:

S  LOp ( Tab Column Name        Data Type(length)    Op Value                )
__ ___ _  T1 Employee Number    CHAR(6)              __ ____________________ _
__ ___ _  T1 First Name         VARCHAR(12)          __ ____________________ _
__ ___ _  T1 Middle Initial     CHAR(1)              __ ____________________ _
__ ___ _  T1 Last Name          VARCHAR(15)          __ ____________________ _
__ ___ _  T1 Work Department    CHAR(3)              __ ____________________ _
__ ___ _  T1 Telephone Number   CHAR(4)              __ ____________________ _
__ ___ _  T1 Commencement Date  DATE                 __ ____________________ _
 F1=Help      F2=Split     F3=Exit      F4=Expand    F6=Execute   F7=Backward
 F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

In this example, an SQL query is developed that shows the Employees (Names and Employee number only), and the Department name; for Department code = "A10". There are several steps required to prototype this statement. They are:

1. Select the following columns by typing 'S' next to the column name:
    - First name
    - Middle Initial
    - Last Name

    Press Enter.

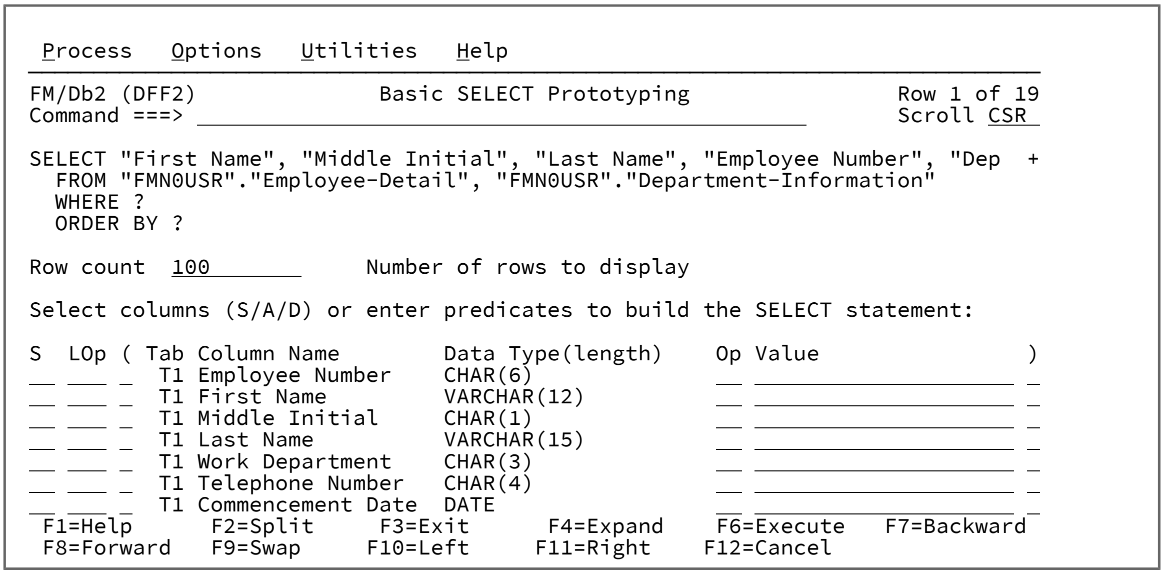
The selected columns are added to the SELECT clause, which is displayed at the top of the panel.

- Employee Number
- Department Code

Press Enter.

See Figure 33: Basic SELECT Prototyping panel (3) on page 229

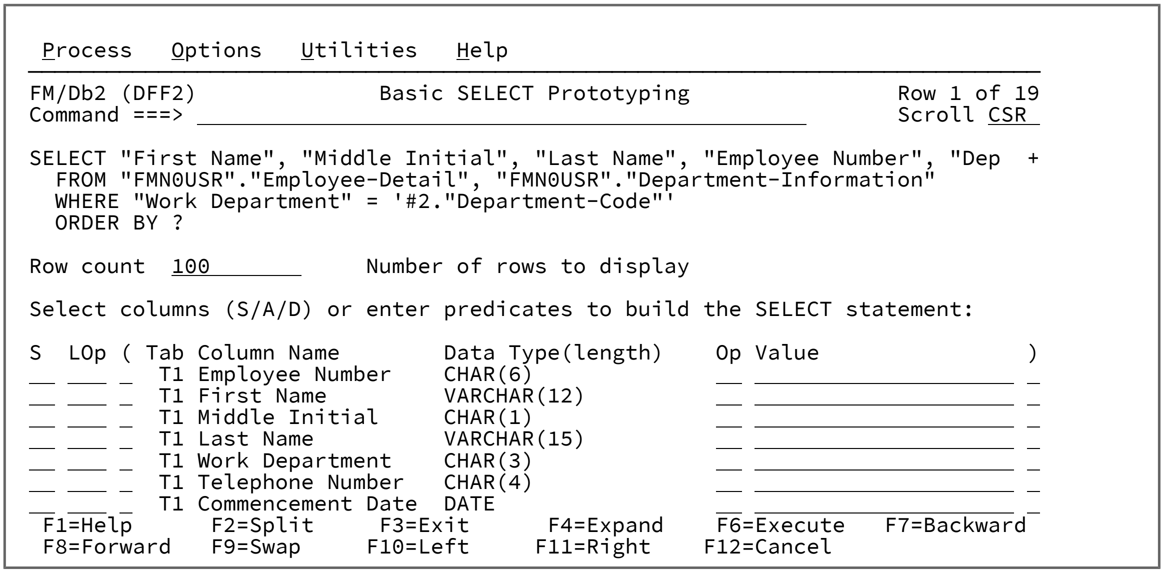
Figure 33. Basic SELECT Prototyping panel (3)



```
 Process   Options   Utilities   Help
───────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)              Basic SELECT Prototyping                Row 1 of 19
Command ===> _______________________________________________       Scroll CSR_

SELECT "First Name", "Middle Initial", "Last Name", "Employee Number", "Dep  +
  FROM "FMN0USR"."Employee-Detail", "FMN0USR"."Department-Information"
  WHERE ?
  ORDER BY ?

Row count  100_______     Number of rows to display

Select columns (S/A/D) or enter predicates to build the SELECT statement:

S  LOp ( Tab Column Name        Data Type(length)    Op Value                )
__ ___ _  T1 Employee Number    CHAR(6)              __ ____________________ _
__ ___ _  T1 First Name         VARCHAR(12)          __ ____________________ _
__ ___ _  T1 Middle Initial     CHAR(1)              __ ____________________ _
__ ___ _  T1 Last Name          VARCHAR(15)          __ ____________________ _
__ ___ _  T1 Work Department    CHAR(3)              __ ____________________ _
__ ___ _  T1 Telephone Number   CHAR(4)              __ ____________________ _
__ ___ _  T1 Commencement Date  DATE                 __ ____________________ _
 F1=Help      F2=Split     F3=Exit      F4=Expand    F6=Execute   F7=Backward
 F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

2. Specify the join between the two tables. To do this, type '#2."Department-Code"' in the "Value" column next to the entry for "Work Department" in the "Employee-Detail" table. You should also specify an '=' in the "Op" column. Press Enter. See Figure 34: Basic SELECT Prototyping panel (4) on page 229

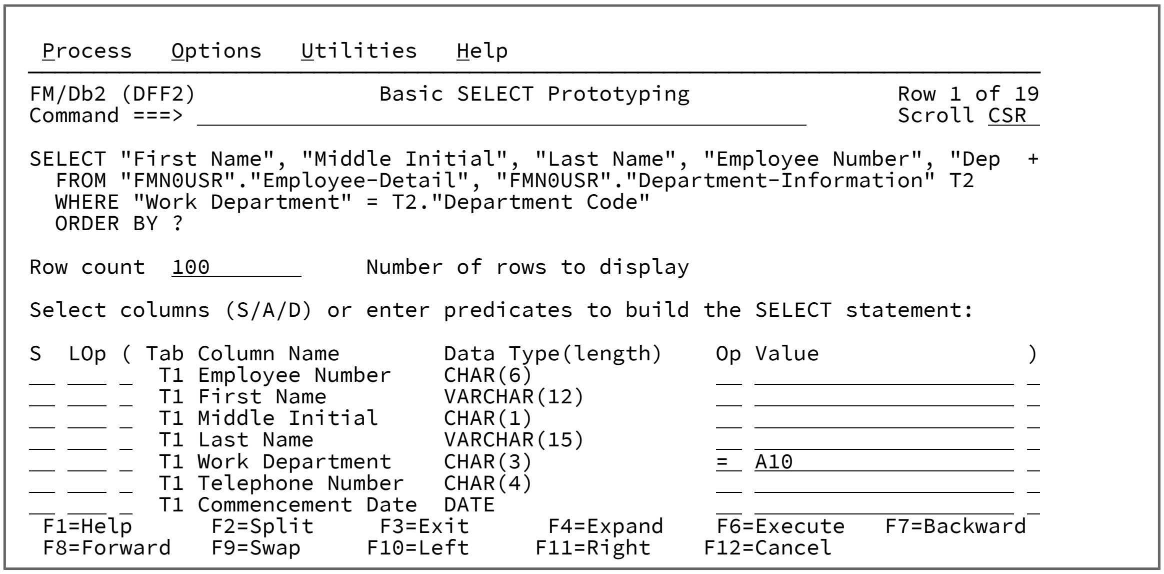
Figure 34. Basic SELECT Prototyping panel (4)



```
 Process   Options   Utilities   Help
───────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)              Basic SELECT Prototyping                Row 1 of 19
Command ===> _______________________________________________       Scroll CSR_

SELECT "First Name", "Middle Initial", "Last Name", "Employee Number", "Dep  +
  FROM "FMN0USR"."Employee-Detail", "FMN0USR"."Department-Information"
  WHERE "Work Department" = '#2."Department-Code"'
  ORDER BY ?

Row count  100_______     Number of rows to display

Select columns (S/A/D) or enter predicates to build the SELECT statement:

S  LOp ( Tab Column Name        Data Type(length)    Op Value                )
__ ___ _  T1 Employee Number    CHAR(6)              __ ____________________ _
__ ___ _  T1 First Name         VARCHAR(12)          __ ____________________ _
__ ___ _  T1 Middle Initial     CHAR(1)              __ ____________________ _
__ ___ _  T1 Last Name          VARCHAR(15)          __ ____________________ _
__ ___ _  T1 Work Department    CHAR(3)              __ ____________________ _
__ ___ _  T1 Telephone Number   CHAR(4)              __ ____________________ _
__ ___ _  T1 Commencement Date  DATE                 __ ____________________ _
 F1=Help      F2=Split     F3=Exit      F4=Expand    F6=Execute   F7=Backward
 F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

3. Add a WHERE clause to show only the employees in the "A10" Department by typing 'A10' in the *"Work Department"* column. Press Enter. See Figure 35: Basic SELECT Prototyping panel (5) on page 230.

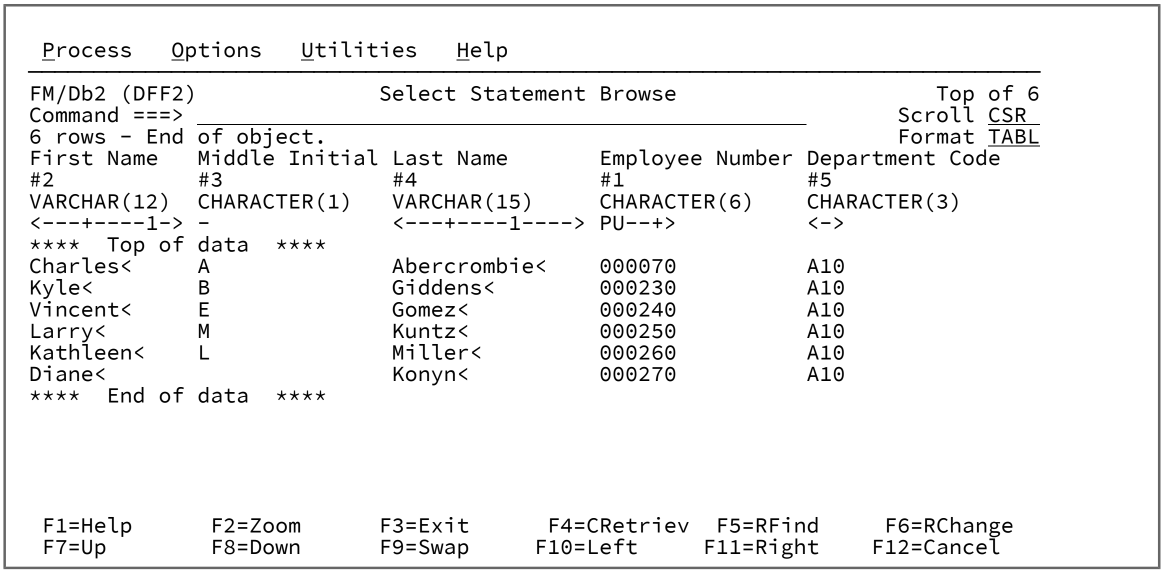
Figure 35. Basic SELECT Prototyping panel (5)



```
 Process   Options   Utilities   Help
 ──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)              Basic SELECT Prototyping                 Row 1 of 19
Command ===> _______________________________________________        Scroll CSR

SELECT "First Name", "Middle Initial", "Last Name", "Employee Number", "Dep  +
  FROM "FMN0USR"."Employee-Detail", "FMN0USR"."Department-Information" T2
  WHERE "Work Department" = T2."Department Code"
  ORDER BY ?

Row count  100             Number of rows to display

Select columns (S/A/D) or enter predicates to build the SELECT statement:

S  LOp ( Tab Column Name        Data Type(length)    Op Value                )
__ ___ _  T1 Employee Number    CHAR(6)              __ ____________________ _
__ ___ _  T1 First Name         VARCHAR(12)          __ ____________________ _
__ ___ _  T1 Middle Initial     CHAR(1)              __ ____________________ _
__ ___ _  T1 Last Name          VARCHAR(15)          __ ____________________ _
__ ___ _  T1 Work Department    CHAR(3)              =  A10                  _
__ ___ _  T1 Telephone Number   CHAR(4)              __ ____________________ _
__ ___ _  T1 Commencement Date  DATE                 __ ____________________ _
 F1=Help      F2=Split     F3=Exit      F4=Expand    F6=Execute   F7=Backward
 F8=Forward   F9=Swap     F10=Left     F11=Right    F12=Cancel
```

4. Either type EXECUTE on the command line, or press the PF6 key to run the SQL statement and display the result table. See Figure 36: Select Statement Browse panel on page 230.

Figure 36. Select Statement Browse panel



```
 Process   Options   Utilities   Help
 ──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)              Select Statement Browse                   Top of 6
Command ===> _______________________________________________        Scroll CSR
6 rows - End of object.                                              Format TABL
First Name   Middle Initial Last Name       Employee Number Department Code
#2           #3             #4              #1              #5
VARCHAR(12)  CHARACTER(1)   VARCHAR(15)     CHARACTER(6)    CHARACTER(3)
<---+----1-> -              <---+----1----> PU--+>          <->
****  Top of data  ****
Charles<     A              Abercrombie<    000070          A10
Kyle<        B              Giddens<        000230          A10
Vincent<     E              Gomez<          000240          A10
Larry<       M              Kuntz<          000250          A10
Kathleen<    L              Miller<         000260          A10
Diane<                      Konyn<          000270          A10
****  End of data  ****




 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F6=RChange
 F7=Up        F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

This completes the Basic SQL Prototyping part of the IVP. Press PF3 repeatedly to return to the FM/Db2 main menu.

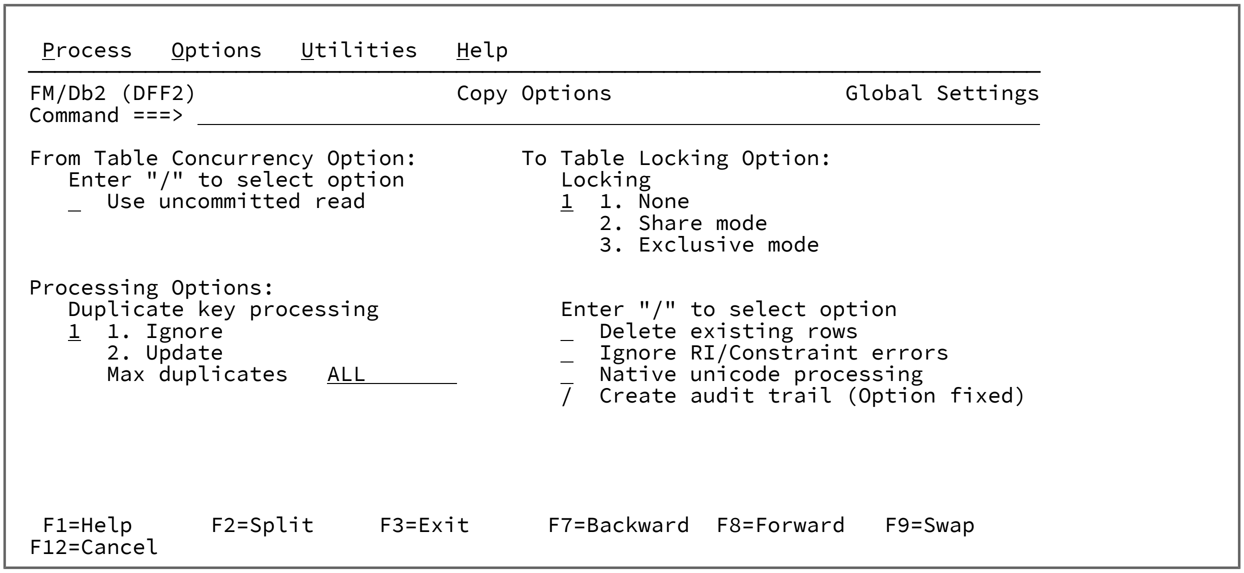
Step 6. Use the FM/Db2 Copy function

Log on to FM/Db2, and connect to the same Db2® system specified in Step 1. Define Db2 objects to be used during verification on page 218.

Set the Copy options as shown in Figure 37: Copy Options panel on page 231. Enter '0.3.3' on the FM/Db2 main menu to access the *"Copy Options"* panel.

Note that the "Create audit trail" option may not be displayed, or may be displayed with different text, depending on choices made during the product install. The auditing setting does not affect the IVP.

Figure 37. Copy Options panel

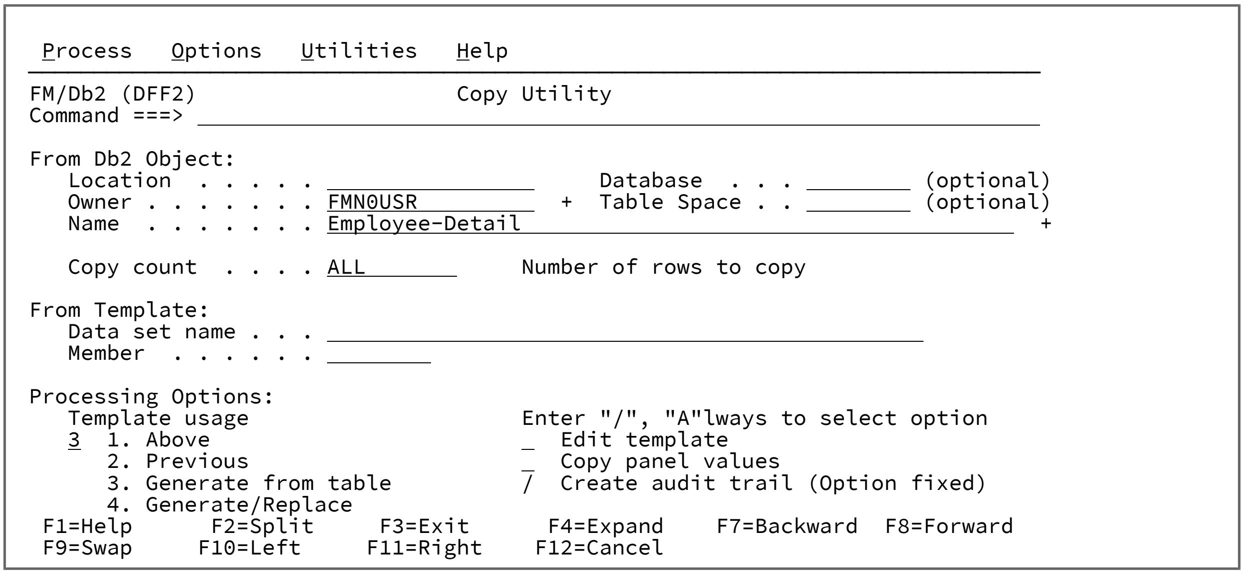


```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                    Copy Options                  Global Settings
Command ===> _________________________________________________________________

From Table Concurrency Option:        To Table Locking Option:
   Enter "/" to select option            Locking
   _  Use uncommitted read               1  1. None
                                            2. Share mode
                                            3. Exclusive mode

Processing Options:
   Duplicate key processing              Enter "/" to select option
   1  1. Ignore                          _  Delete existing rows
      2. Update                          _  Ignore RI/Constraint errors
      Max duplicates   ALL_______        _  Native unicode processing
                                         /  Create audit trail (Option fixed)




 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

Return to the FM/Db2 main menu by pressing PF3. Select the FM/Db2 *“Copy function”* by typing '3.3' on the FM/Db2 main menu and pressing Enter.

Type 'FMN0USR' and 'Employee-Detail' on the *“From Db2® Object”* field as shown in Figure 38: Copy Utility panel on page 231. Press Enter to show the COPY *“To”* field in the panel.

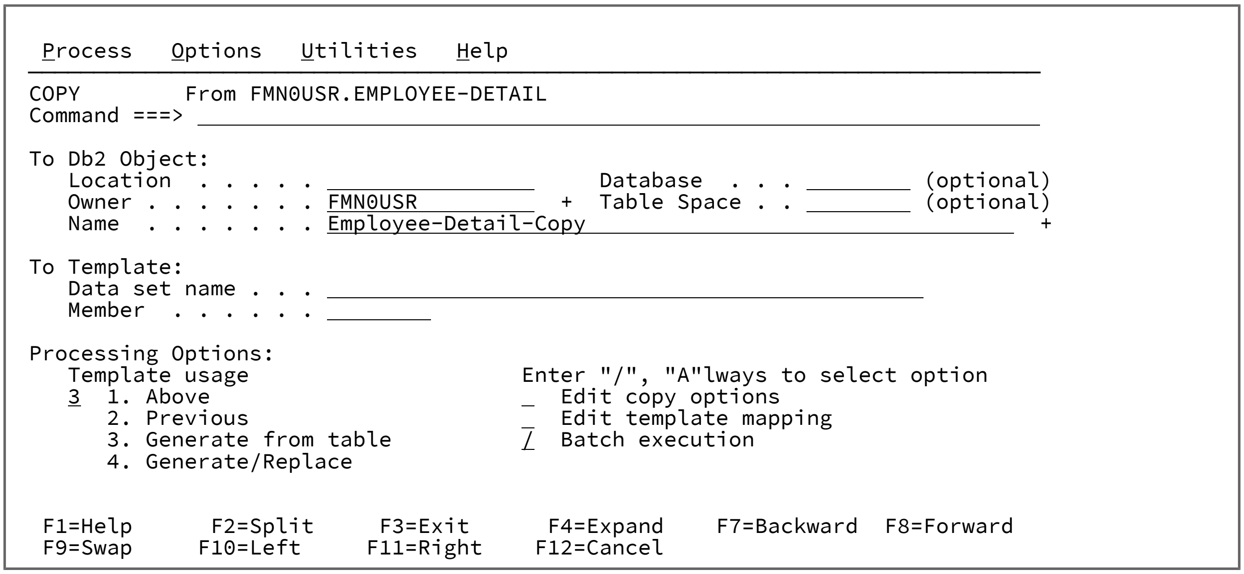
Figure 38. Copy Utility panel



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                    Copy Utility
Command ===> _________________________________________________________________

From Db2 Object:
   Location  . . . . . ________________     Database  . . . ________ (optional)
   Owner . . . . . . . FMN0USR_________  +  Table Space . . ________ (optional)
   Name  . . . . . . . Employee-Detail______________________________________  +

   Copy count  . . . . ALL_______     Number of rows to copy

From Template:
   Data set name . . . ______________________________________________
   Member  . . . . . . ________

Processing Options:
   Template usage                     Enter "/", "A"lways to select option
   3  1. Above                        _  Edit template
      2. Previous                     _  Copy panel values
      3. Generate from table          /  Create audit trail (Option fixed)
      4. Generate/Replace
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Type 'FMN0USR' and 'Employee-Detail-Copy' on the FM/Db2 COPY *“To”* field panel, and select the *“Batch execution”* option under the *“Processing Options”* field, as shown in Figure 39: Basic SELECT Prototyping panel (2) on page 232.

Figure 39. Basic SELECT Prototyping panel (2)



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
COPY        From FMN0USR.EMPLOYEE-DETAIL
Command ===> _________________________________________________________________

To Db2 Object:
   Location . . . . . ________________     Database  . . . ________ (optional)
   Owner . . . . . . . FMN0USR_________  +  Table Space . . ________ (optional)
   Name  . . . . . . . Employee-Detail-Copy_______________________________   +

To Template:
   Data set name . . . ______________________________________________
   Member  . . . . . . ________

Processing Options:
   Template usage                     Enter "/", "A"lways to select option
   3  1. Above                        _  Edit copy options
      2. Previous                     _  Edit template mapping
      3. Generate from table          /  Batch execution
      4. Generate/Replace


 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Press Enter to display the generated JCL for the FM/Db2 batch copy job.

Review the generated JCL. It should not be necessary to change the STEPLIB DD statements if the correct Db2® libraries are specified in the FMN2SSDM macro entry for the currently connected Db2® system.

Submit the job. The expected return code is RC=0, with 48 rows copied.

You can examine the data copied to the *"Employee-Detail-Copy"* table using the FM/Db2 editor. Note that the columns are similar to those in the *"Employee-Detail"* table, but are in a different order.

Step 7. Use the FM/Db2 Export function

Log on to FM/Db2, and connect to the same Db2® system specified in Step 1. Define Db2 objects to be used during verification on page 218.

For this step of the IVP a PDS or PDS/E is required, where the FM/Db2 template for the exported data will be stored. The examples given assume a name of

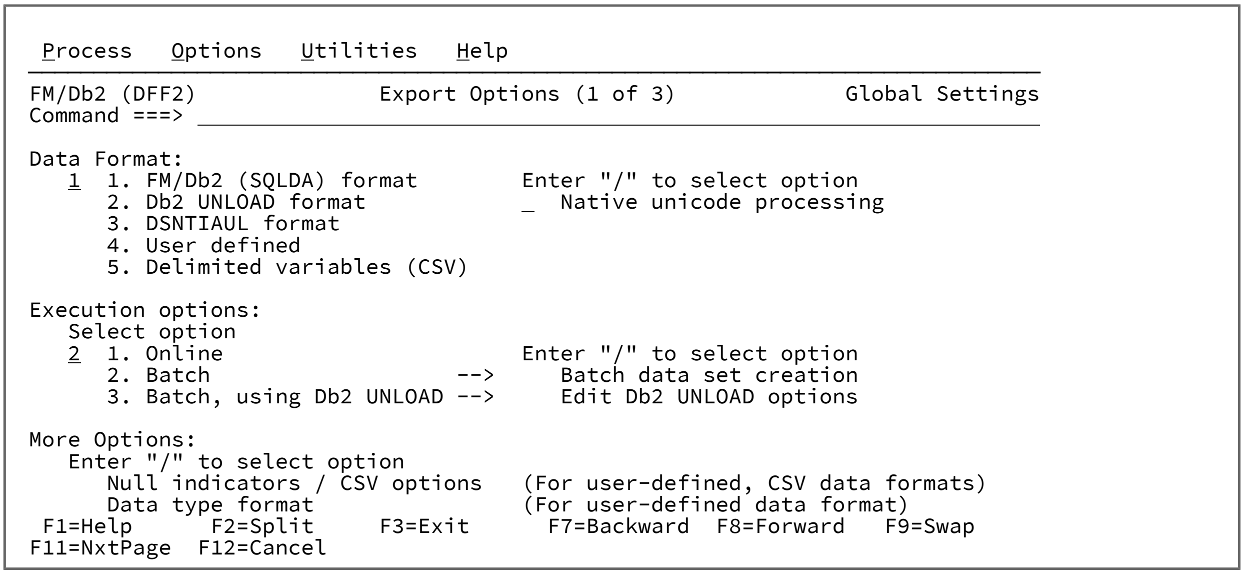
```
TSO logonid.FM.TEMPLATE
```

You can specify any valid data set name that your TSO logon ID has update access to. You can use ISPF option 3.2, or a batch job, to create the data set. Create this data set with these attributes:

```
Organization..:PO
Record format...:FB
Record length...:80
Data set name type   PDS or Library
```

Set the Export options as shown in Figure 40: Export Options (1 of 3) on page 233. Enter '0.3.7' on the FM/Db2 main menu to access the *"Export Options panel"*.

Figure 40. Export Options (1 of 3)



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)              Export Options (1 of 3)            Global Settings
Command ===> _________________________________________________________________

Data Format:
   1  1. FM/Db2 (SQLDA) format        Enter "/" to select option
      2. Db2 UNLOAD format            _  Native unicode processing
      3. DSNTIAUL format
      4. User defined
      5. Delimited variables (CSV)

Execution options:
   Select option
   2  1. Online                       Enter "/" to select option
      2. Batch                   -->     Batch data set creation
      3. Batch, using Db2 UNLOAD -->     Edit Db2 UNLOAD options

More Options:
   Enter "/" to select option
      Null indicators / CSV options   (For user-defined, CSV data formats)
      Data type format                (For user-defined data format)
 F1=Help      F2=Split     F3=Exit      F7=Backward   F8=Forward   F9=Swap
F11=NxtPage  F12=Cancel
```

Return to the FM/Db2 main menu by pressing PF3.

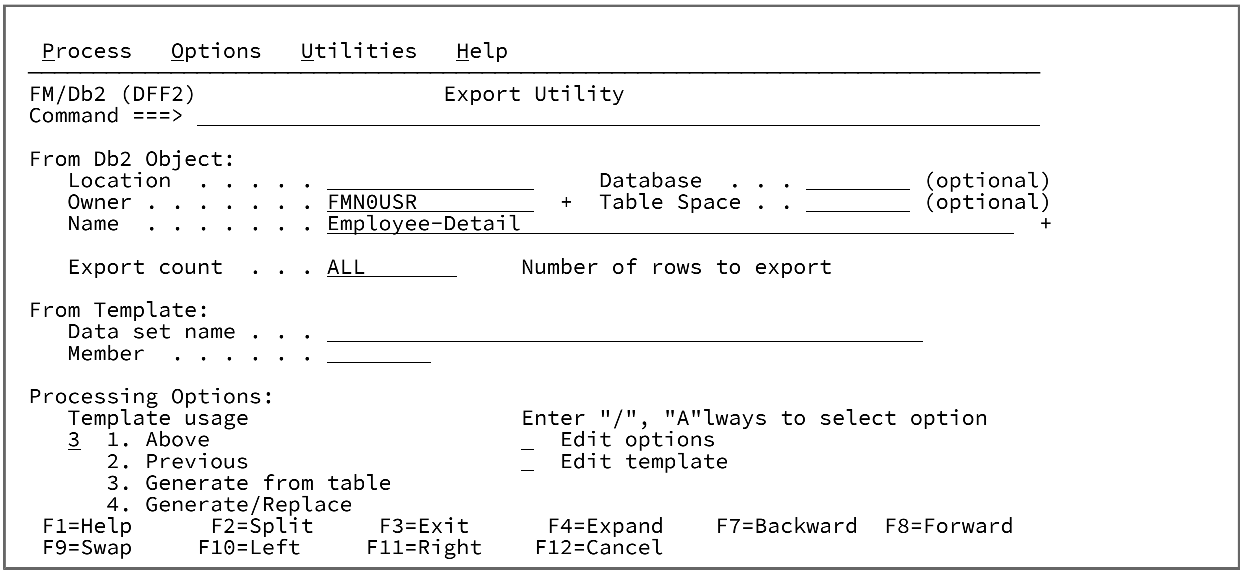
Select the FM/Db2 Export function by typing '3.7' on the FM/Db2 main menu and pressing Enter.

Type 'FMN0USR' and 'Employee-Detail' on the *“From Db2® object”* field as shown in Figure 41: Export Utility on page 234. Press Enterto show the FM/Db2 *“Export To”* panel.

Enter these values on the EXPORT *“To partitioned...”* panel:

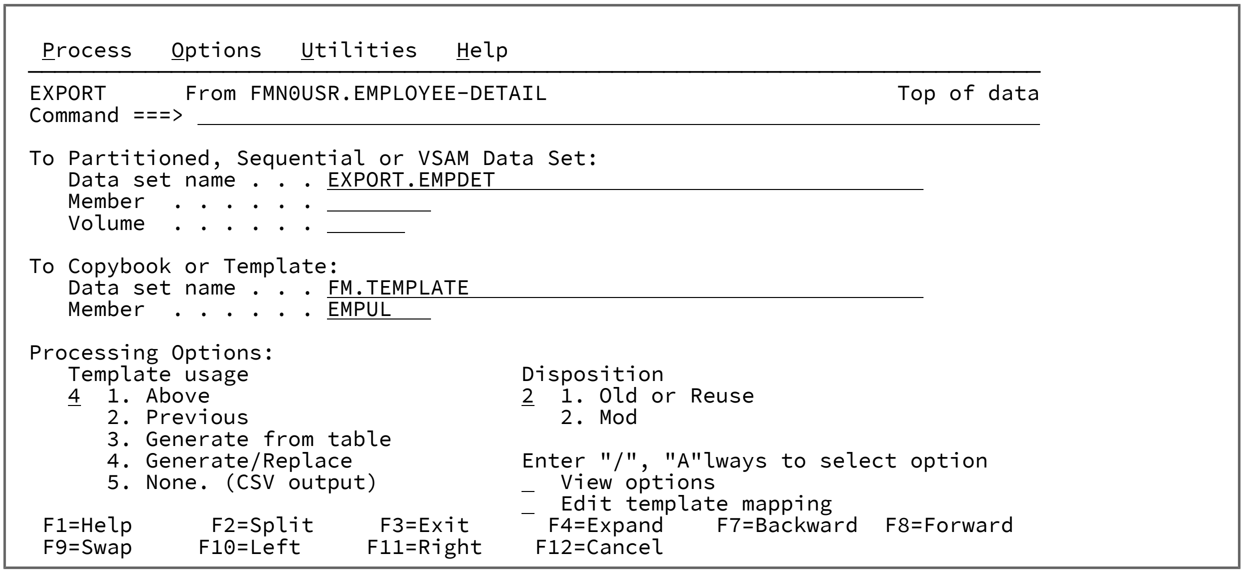
Export data set name: `EXPORT.EMPDET`
Template data set name: `FM.TEMPLATE`
Template member name: `EMPUL`
Template usage: `4`
Disposition: `2`

Figure 41. Export Utility



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                     Export Utility
Command ===> _________________________________________________________________

From Db2 Object:
   Location  . . . . . ________________      Database  . . . ________ (optional)
   Owner . . . . . . . FMN0USR_________  +   Table Space . . ________ (optional)
   Name  . . . . . . . Employee-Detail_________________________________________  +

   Export count  . . . ALL_______      Number of rows to export

From Template:
   Data set name . . . ______________________________________________
   Member  . . . . . . ________

Processing Options:
   Template usage                      Enter "/", "A"lways to select option
   3  1. Above                         _  Edit options
      2. Previous                      _  Edit template
      3. Generate from table
      4. Generate/Replace
 F1=Help      F2=Split     F3=Exit      F4=Expand     F7=Backward   F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 42. Export...From panel



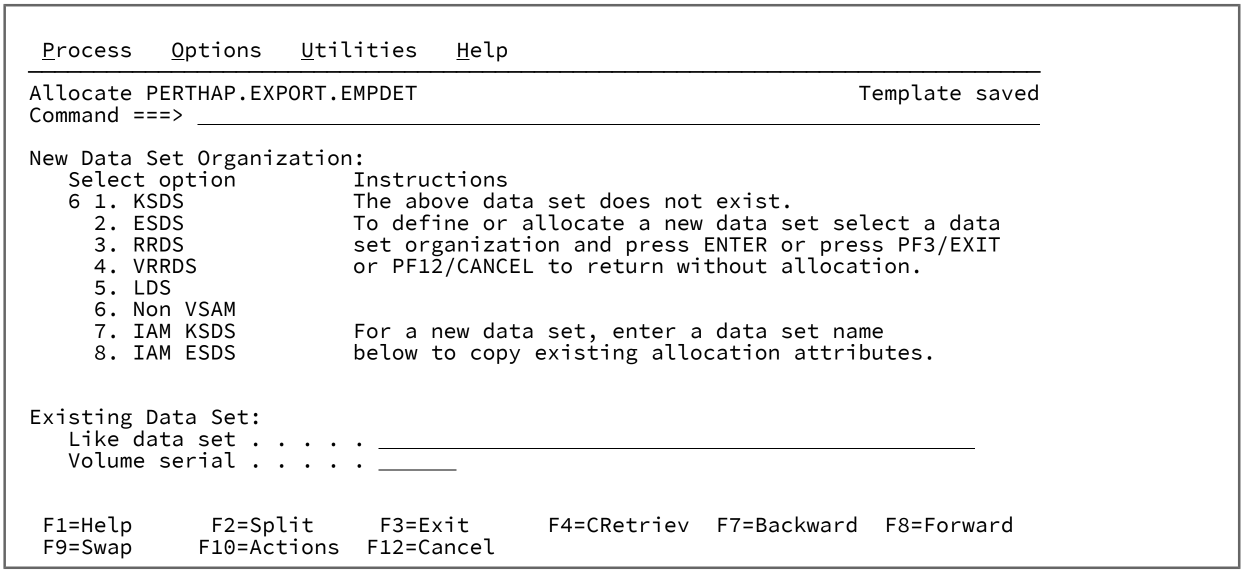
```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
EXPORT      From FMN0USR.EMPLOYEE-DETAIL                           Top of data
Command ===> _________________________________________________________________

To Partitioned, Sequential or VSAM Data Set:
   Data set name . . . EXPORT.EMPDET_________________________________
   Member  . . . . . . ________
   Volume  . . . . . . ______

To Copybook or Template:
   Data set name . . . FM.TEMPLATE___________________________________
   Member  . . . . . . EMPUL___

Processing Options:
   Template usage                      Disposition
   4  1. Above                         2  1. Old or Reuse
      2. Previous                         2. Mod
      3. Generate from table
      4. Generate/Replace              Enter "/", "A"lways to select option
      5. None. (CSV output)            _  View options
                                       _  Edit template mapping
 F1=Help      F2=Split     F3=Exit      F4=Expand     F7=Backward   F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Press Enter. An allocation panel for the export data set is displayed. See Figure 43: Allocation panel for Export data on page 235. Type '6' for a *"Non VSAM"* data set and press Enter to move to the second allocation panel. See Figure 44: Allocation panel for Export data (2) on page 235.

Note:

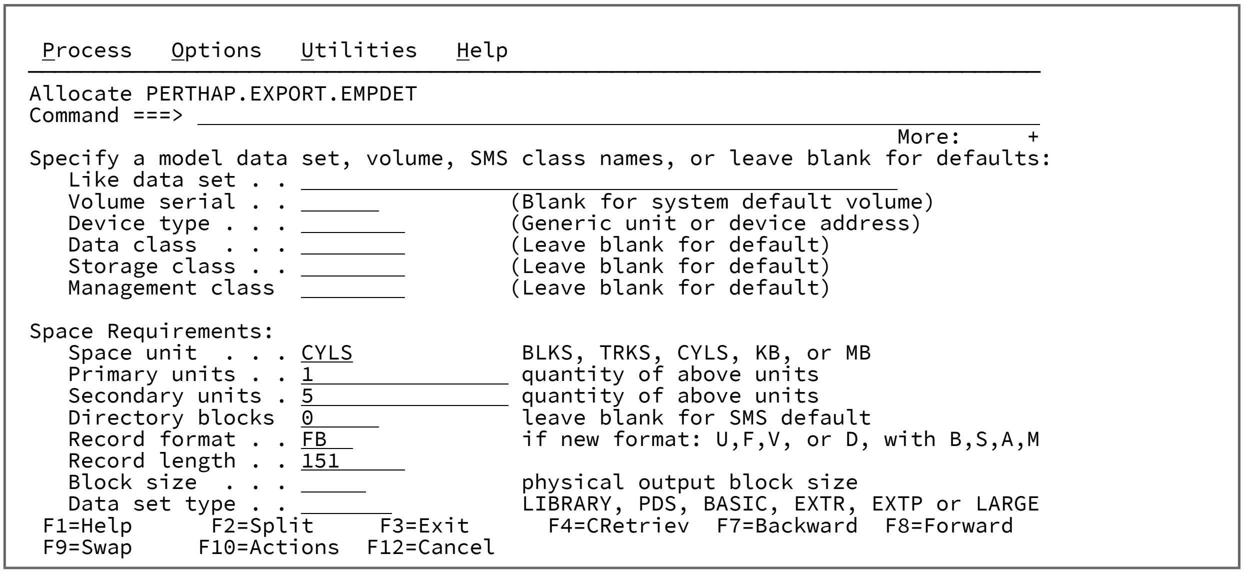
- If you execute this IVP step on subsequent occasions you may see a "Template replace" pop-up panel. This is a warning panel displayed when an FM/Db2 function is about to replace a template in a member of PDS that already exists. Press Enter to continue.
- On subsequent executions of this IVP step the allocation panel for the export data set is not shown.

Figure 43. Allocation panel for Export data



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
Allocate PERTHAP.EXPORT.EMPDET                                  Template saved
Command ===> _________________________________________________________________

New Data Set Organization:
   Select option         Instructions
   6 1. KSDS             The above data set does not exist.
     2. ESDS             To define or allocate a new data set select a data
     3. RRDS             set organization and press ENTER or press PF3/EXIT
     4. VRRDS            or PF12/CANCEL to return without allocation.
     5. LDS
     6. Non VSAM
     7. IAM KSDS         For a new data set, enter a data set name
     8. IAM ESDS         below to copy existing allocation attributes.


Existing Data Set:
   Like data set . . . . . ____________________________________________
   Volume serial . . . . . ______


 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 44. Allocation panel for Export data (2)



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
Allocate PERTHAP.EXPORT.EMPDET
Command ===> _________________________________________________________________
                                                                   More:     +
Specify a model data set, volume, SMS class names, or leave blank for defaults:
   Like data set . . ____________________________________________
   Volume serial . . ______         (Blank for system default volume)
   Device type . . . ________       (Generic unit or device address)
   Data class  . . . ________       (Leave blank for default)
   Storage class . . ________       (Leave blank for default)
   Management class  ________       (Leave blank for default)

Space Requirements:
   Space unit  . . . CYLS            BLKS, TRKS, CYLS, KB, or MB
   Primary units . . 1______________ quantity of above units
   Secondary units . 5______________ quantity of above units
   Directory blocks  0_____          leave blank for SMS default
   Record format . . FB__            if new format: U,F,V, or D, with B,S,A,M
   Record length . . 151_____
   Block size  . . . _____           physical output block size
   Data set type . . _______         LIBRARY, PDS, BASIC, EXTR, EXTP or LARGE
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Review the generated JCL. It should not be necessary to change the STEPLIB DD statements if the correct Db2® libraries are specified in the FMN2SSDM macro entry for the currently connected Db2® system.

Submit the job. The expected return code is RC=0, with 48 rows exported.

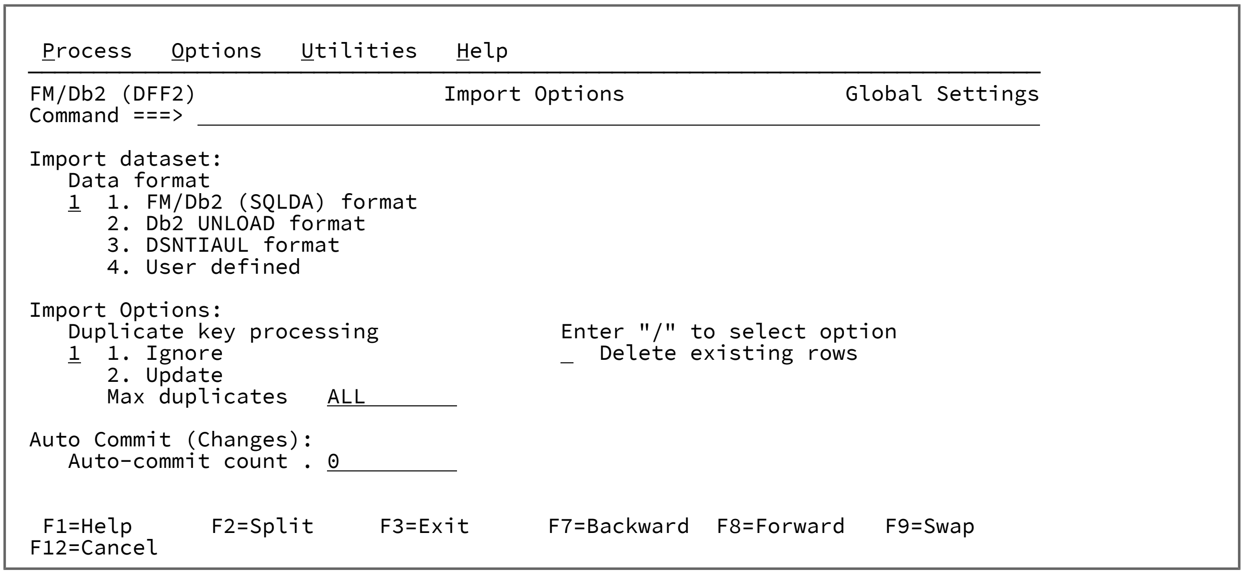
You can browse the exported data set using the View option in the File Manager base component.

Step 8. Use the FM/Db2 Import function

Log on to FM/Db2, and connect to the same Db2® system specified in Step 1. Define Db2 objects to be used during verification on page 218.

Set the Import options as shown in Figure 45: Import Options panel on page 236. Enter '0.3.6' on the FM/Db2 main menu to access the "Import Options" panel.

Figure 45. Import Options panel



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                   Import Options                  Global Settings
Command ===> ___________________________________________________________________

Import dataset:
   Data format
   1  1. FM/Db2 (SQLDA) format
      2. Db2 UNLOAD format
      3. DSNTIAUL format
      4. User defined

Import Options:
   Duplicate key processing              Enter "/" to select option
   1  1. Ignore                          _  Delete existing rows
      2. Update
      Max duplicates   ALL_______

Auto Commit (Changes):
   Auto-commit count . 0_________


 F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
F12=Cancel
```

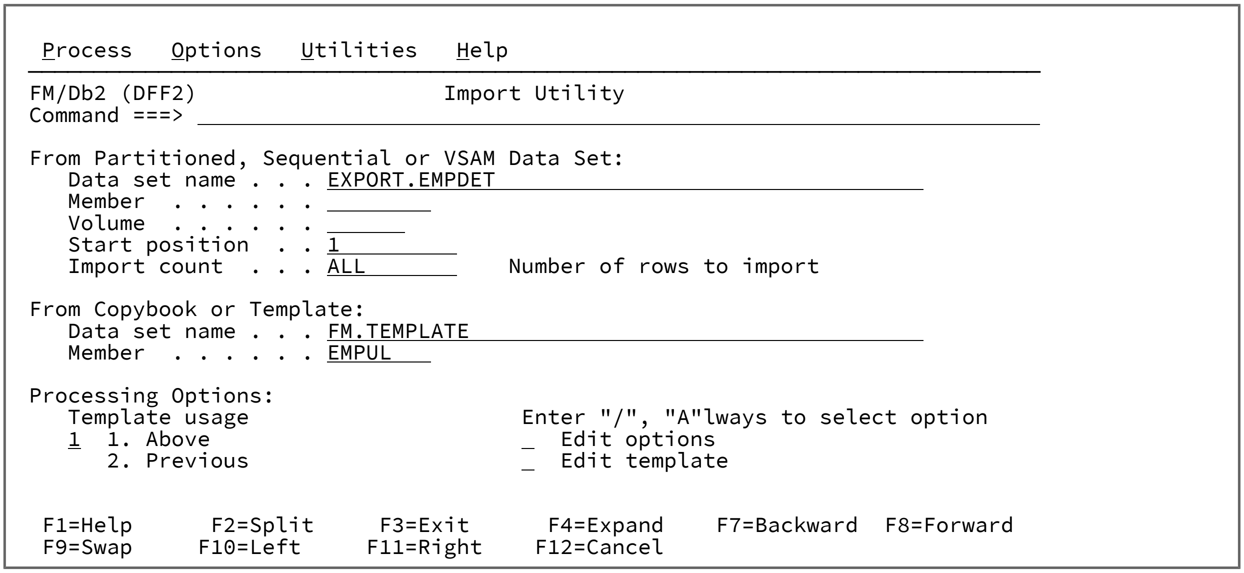
Return to the FM/Db2 main menu by pressing PF3. Select the FM/Db2 Import function by typing '3.6' on the FM/Db2 main menu and pressing Enter.

Enter these values on the FM/Db2 Import *"From"* panel (See Figure 46: Import Utility panel on page 237):

From data set name: `EXPORT.EMPDET`
Template data set name: `FM.TEMPLATE`
Template member name: `EMPUL`
Template usage: `1`

Note: The template data set and member name should match the values you entered in Step 7. Use the FM/Db2 Export function on page 232.

Figure 46. Import Utility panel

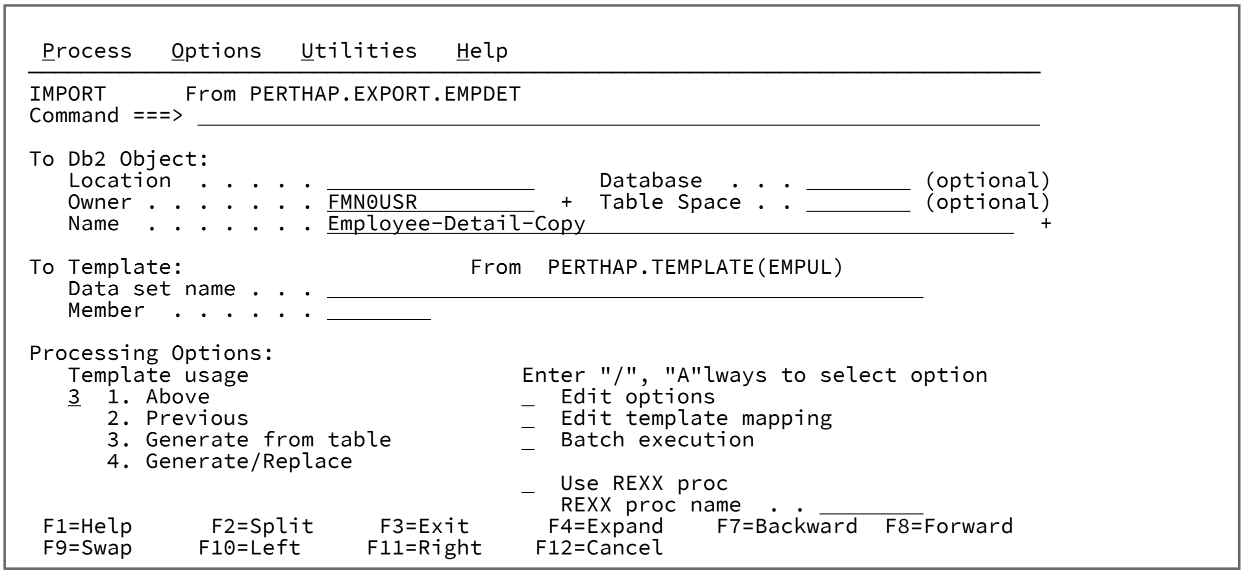


```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                  Import Utility
Command ===> _________________________________________________________________

From Partitioned, Sequential or VSAM Data Set:
   Data set name . . . EXPORT.EMPDET____________________________________
   Member  . . . . . . ________
   Volume  . . . . . . ______
   Start position  . . 1_________
   Import count  . . . ALL_______    Number of rows to import

From Copybook or Template:
   Data set name . . . FM.TEMPLATE______________________________________
   Member  . . . . . . EMPUL___

Processing Options:
   Template usage                     Enter "/", "A"lways to select option
   1  1. Above                        _  Edit options
      2. Previous                     _  Edit template


 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Press Enter to display the FM/Db2 Import "To" panel.

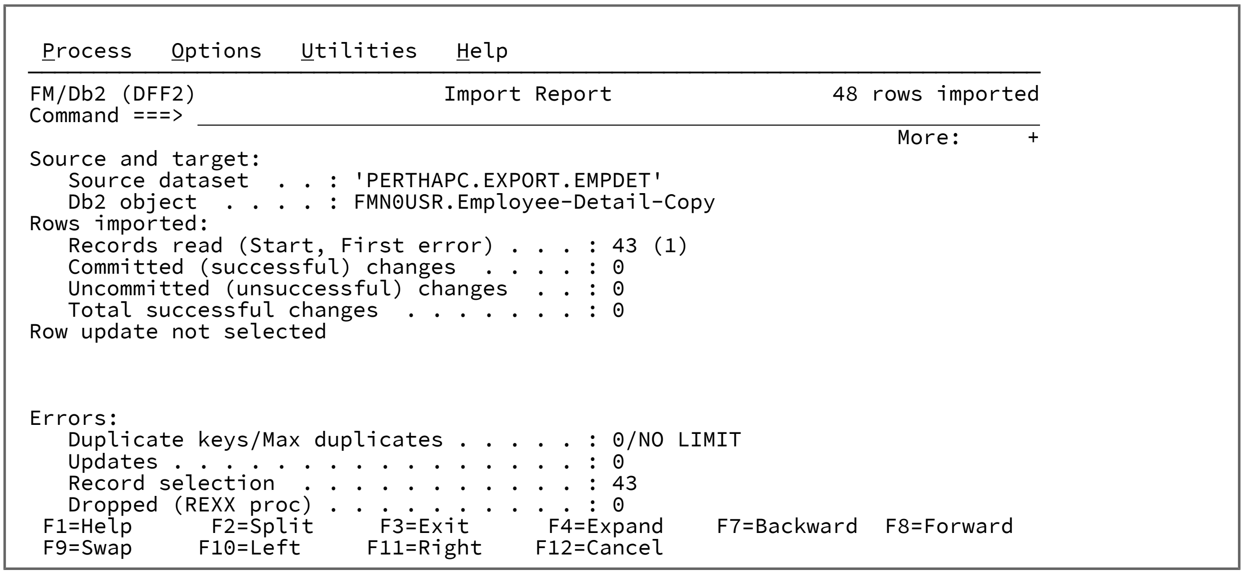
Type 'FMN0USR' and 'Employee-Detail-Copy' on the FM/Db2 Import *"To"* field panel as shown in Figure 47: IMPORT From panel on page 237. Press Enter to start the Import process. The *"Import Report"* panel is displayed. See Figure 48: Import Report panel on page 238.

Figure 47. IMPORT From panel



```
 Process   Options   Utilities   Help
──────────────────────────────────────────────────────────────────────────────
IMPORT      From PERTHAP.EXPORT.EMPDET
Command ===> _________________________________________________________________

To Db2 Object:
   Location  . . . . . ________________     Database  . . . ________ (optional)
   Owner . . . . . . . FMN0USR_________  +  Table Space . . ________ (optional)
   Name  . . . . . . . Employee-Detail-Copy_____________________________________  +

To Template:                         From  PERTHAP.TEMPLATE(EMPUL)
   Data set name . . . ______________________________________________
   Member  . . . . . . ________

Processing Options:
   Template usage                     Enter "/", "A"lways to select option
   3  1. Above                        _  Edit options
      2. Previous                     _  Edit template mapping
      3. Generate from table          _  Batch execution
      4. Generate/Replace
                                      _  Use REXX proc
                                         REXX proc name  . . ________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Figure 48. Import Report panel



```
 Process   Options   Utilities   Help
 ──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                   Import Report                  48 rows imported
Command ===> __________________________________________________________________
                                                                     More:    +
Source and target:
   Source dataset  . . : 'PERTHAPC.EXPORT.EMPDET'
   Db2 object  . . . . : FMN0USR.Employee-Detail-Copy
Rows imported:
   Records read (Start, First error) . . . : 43 (1)
   Committed (successful) changes  . . . . : 0
   Uncommitted (unsuccessful) changes  . . : 0
   Total successful changes  . . . . . . . : 0
Row update not selected



Errors:
   Duplicate keys/Max duplicates . . . . . : 0/NO LIMIT
   Updates . . . . . . . . . . . . . . . . : 0
   Record selection  . . . . . . . . . . . : 43
   Dropped (REXX proc) . . . . . . . . . . : 0
 F1=Help      F2=Split     F3=Exit      F4=Expand     F7=Backward   F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

You can use the FM/Db2 editor to display the contents of the FMN0USR.*"Employee-Detail-Copy"* table.

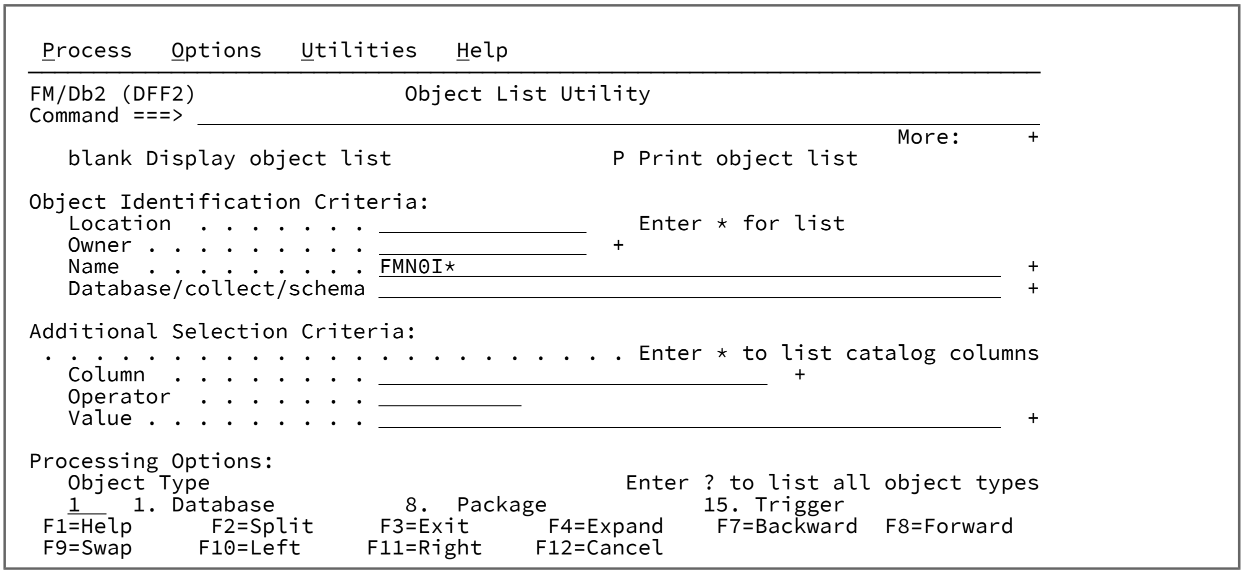
Step 9. Use the FM/Db2 Object List utility

Log on to FM/Db2, and connect to the same Db2® system specified in Verifying the customization of FM/Db2 on page 218.

Select the FM/Db2 Object List utility by typing '3.4' on the FM/Db2 main menu panel and press Enter.

Type 'FMN0I*' in the Name field and change the *"Object type"* value to '1'. This will show all databases that start with *"FMN0I"*. See Figure 49: Object List Utility panel on page 238

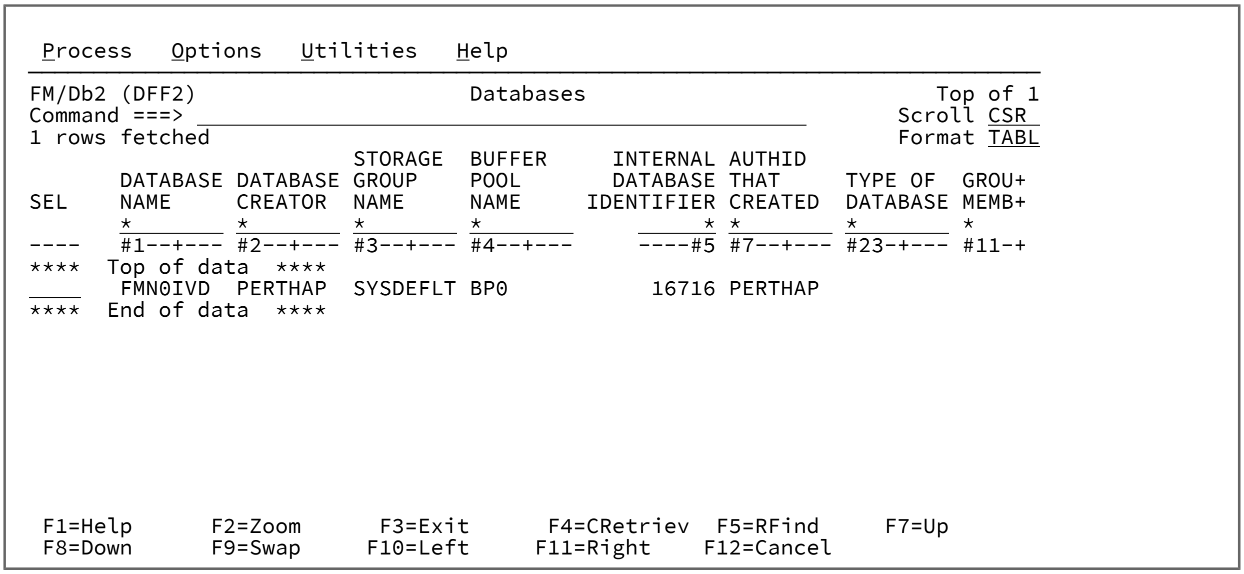
Figure 49. Object List Utility panel



```
 Process   Options   Utilities   Help
 ──────────────────────────────────────────────────────────────────────────────
FM/Db2 (DFF2)                Object List Utility
Command ===> __________________________________________________________________
                                                                     More:    +
   blank Display object list                P Print object list

Object Identification Criteria:
   Location  . . . . . . . ________________    Enter * for list
   Owner . . . . . . . . . ________________  +
   Name  . . . . . . . . . FMN0I*____________________________________________   +
   Database/collect/schema ________________________________________________   +

Additional Selection Criteria:
 . . . . . . . . . . . . . . . . . . . . . . . Enter * to list catalog columns
   Column  . . . . . . . . ______________________________  +
   Operator  . . . . . . . _________
   Value . . . . . . . . . ________________________________________________   +

Processing Options:
   Object Type                               Enter ? to list all object types
   1__  1. Database          8.  Package          15. Trigger
 F1=Help      F2=Split     F3=Exit      F4=Expand     F7=Backward   F8=Forward
 F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Press Enter to display the list of databases. See Figure 50: Databases panel on page 239

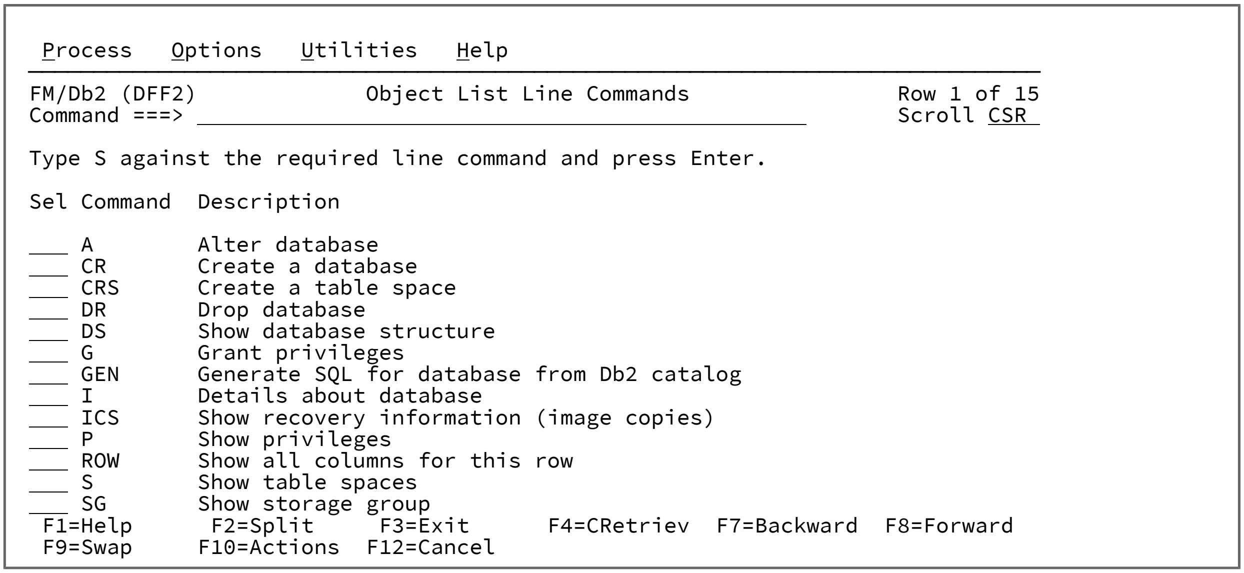
Figure 50. Databases panel



```
 Process   Options   Utilities   Help
 ------------------------------------------------------------------------------
FM/Db2 (DFF2)                     Databases                            Top of 1
Command ===> ______________________________________________         Scroll CSR_
1 rows fetched                                                      Format TABL
                         STORAGE  BUFFER     INTERNAL AUTHID
       DATABASE DATABASE GROUP    POOL       DATABASE THAT     TYPE OF  GROU+
SEL    NAME     CREATOR  NAME     NAME     IDENTIFIER CREATED  DATABASE MEMB+
       *_______ *_______ *_______ *_______      _____* *_______ *_______ *
----   #1--+--- #2--+--- #3--+--- #4--+---     ----#5 #7--+--- #23-+--- #11-+
****  Top of data  ****
____   FMN0IVD  PERTHAP  SYSDEFLT BP0           16716 PERTHAP
****  End of data  ****






 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Type '?' against the FMN0IVD database entry and press Enter. See Figure 51: Object List Line Commands panel on page 239.

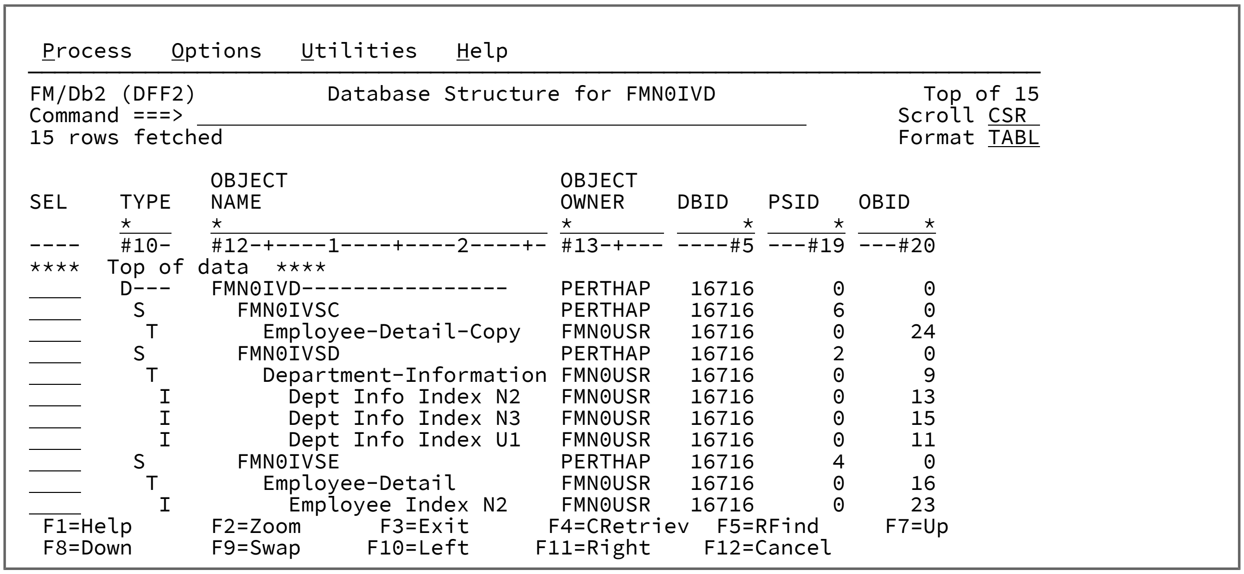
Figure 51. Object List Line Commands panel



```
 Process   Options   Utilities   Help
 ------------------------------------------------------------------------------
FM/Db2 (DFF2)            Object List Line Commands                 Row 1 of 15
Command ===> ______________________________________________         Scroll CSR_

Type S against the required line command and press Enter.

Sel Command  Description

___ A        Alter database
___ CR       Create a database
___ CRS      Create a table space
___ DR       Drop database
___ DS       Show database structure
___ G        Grant privileges
___ GEN      Generate SQL for database from Db2 catalog
___ I        Details about database
___ ICS      Show recovery information (image copies)
___ P        Show privileges
___ ROW      Show all columns for this row
___ S        Show table spaces
___ SG       Show storage group
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Type an 'S' against the "DS" entry to show the database structure. Press Enter. See Figure 52: Database Structure for FMN0IVD panel on page 240.

Figure 52. Database Structure for FMN0IVD panel



```
 Process   Options   Utilities   Help
FM/Db2 (DFF2)          Database Structure for FMN0IVD                 Top of 15
Command ===> _______________________________________________       Scroll CSR
15 rows fetched                                                    Format TABL

              OBJECT                     OBJECT
SEL    TYPE   NAME                       OWNER    DBID   PSID   OBID
       *___   *_________________________ *_______ _____* _____* _____*
----   #10-   #12-+----1----+----2----+- #13-+--- ----#5 ---#19 ---#20
****  Top of data  ****
____   D---   FMN0IVD----------------    PERTHAP   16716      0      0
____    S       FMN0IVSC                 PERTHAP   16716      6      0
____     T        Employee-Detail-Copy   FMN0USR   16716      0     24
____    S       FMN0IVSD                 PERTHAP   16716      2      0
____     T        Department-Information FMN0USR   16716      0      9
____      I         Dept Info Index N2   FMN0USR   16716      0     13
____      I         Dept Info Index N3   FMN0USR   16716      0     15
____      I         Dept Info Index U1   FMN0USR   16716      0     11
____    S       FMN0IVSE                 PERTHAP   16716      4      0
____     T        Employee-Detail        FMN0USR   16716      0     16
____      I         Employee Index N2    FMN0USR   16716      0     23
 F1=Help      F2=Zoom      F3=Exit      F4=CRetriev  F5=RFind     F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Press PF3 repeatedly to return to the FM/Db2 main menu.

This completes the FM/Db2 IVP.

Chapter 20. Re-installing FM/Db2 after migration of a Db2® system

When migration of an existing Db2® system to a new version of Db2® is complete, you should revisit the following FM/Db2 installation tasks:

- Granting access to the Db2 catalog (required) on page 159
- Binding Db2 (required) on page 170

Verifying FM/Db2 has access to any new/changed Db2® catalog tables.

Depending on the method used to grant access to the Db2® catalog tables, you made need to execute additional grant statements, or drop and re-create views on the Db2® catalog tables.

The important point here is that you should start with the samples for the new version, rather than the old version.

Binding Db2®

Locate the sample bind plan and bind package jobs for the new version of Db2®. Customize these jobs as required and run them.

The important point here is that you should start with the samples for the new version, rather than the old version.

Note: File Manager provides sample bind jobs only for supported versions of Db2® running in new-function mode (NFM).

Part III. Customizing File Manager IMS™ Component

Chapter 21. Preparing to customize FM/IMS

Before you can install and customize FM/IMS, you must have installed the File Manager base function. These topics 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 Customizing File Manager on page 12.

Table 38: Summary of steps for customizing FM/IMS and the operating environment on page 243 lists the customization tasks you can perform for FM/IMS. 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 38. Summary of steps for customizing FM/IMS and the operating environment

	Description
__ 1	Concatenate FM/IMS libraries to the LINKLIST. See Concatenating libraries to the LINKLIST on page 257.
__ 2	Modify the TSO logon procedure See Modifying the TSO logon procedure on page 257.
__ 3	Add FM/IMS to the ISPF menu. See Adding FM/IMS to your ISPF menu on page 260.
__ 4	Define FM/IMS in an ISPF command table. See Defining FM/IMS in an ISPF command table on page 260.
__ 5	Customize IMS™ to support the use of dynamic PSBs in BMP mode. See Customizing IMS to support the use of dynamic PSBs in BMP mode on page 261.
__ 6	Customize the FM/IMS installation options module See Customizing the FM/IMS installation options module on page 263.
__ 7	Tailor the job control skeletons. See Tailoring the job control skeletons on page 277.
__ 8	Customize the FM/IMS security exit, FMN1SXT. See Customizing the FM/IMS security exit on page 287.
__ 9	Provide access to protected functions using RACF®. See IMS subsystems and FM/IMS functions access control facility on page 281.
__ 10	Decide how to customize the FM/IMS audit facility. See Alternatives for controlling FM/IMS auditing on page 257.
__ 11	Customize FM/IMS for national languages. See Customizing FM/IMS for national languages on page 308.

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)

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.
- Whether FM/IMS functions which are using PSBs that have update intent use an IMS™ log when accessing each subsystem and, if they do use a log, whether the log is kept when the function ends.
- The name pattern that FM/IMS functions use to generate IMS™ log data set names 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 263.

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 263.

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 Controlling access to databases by FM/IMS functions on page 248 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 databases in BMP mode, do the following steps:

- 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 261.
- Provide an ACBLIB data set into which the ACB Maintenance utility can generate DOPT PSBs. For more information on how you do this, see Providing a DOPT ACBLIB data set on page 262.

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 databases in BMP mode, do the following steps:

- Use the APPLCTN macro or the CREATE PGM command to declare the PSB.
- Use the ACB Maintenance utility or the ACB Generation and Catalog Populate utility (DFS3UACB) to generate an ACB 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 265.

Security

This section outlines how you can protect IMS™ resources that can be accessed through FM/IMS functions.

Controlling access to databases by FM/IMS functions

Security administrators can control the access that users have to databases when using FM/IMS functions.

Different methods of controlling access are required depending on whether the functions run in DLI or BMP mode.

For functions that run in DLI mode, you must control access to the database data sets. At most installations, the database data sets will already be protected by RACF® or an equivalent security product. If this is not the case, you will need to define RACF® profiles for the database data set resources.

For functions that run in BMP mode, there are several different ways of controlling users' access to databases.

The standard way is to use Resource Access Security (RAS) and the IIMS and JIMS RACF® security classes to control which PSBs each user can use. An advantage of this method is that the protection that it provides is not just restricted to when the access is through FM/IMS functions. However, the method has one serious limitation: it can only be used to control access by functions that use static PSBs. One of the following methods must be used to control access by functions that use dynamic PSBs.

1. You can use the FM/IMS Database Access Control facility.

   For information on how to customize this facility, see The Database Access Control facility on page 280.
2. You can provide a version of the FMN1SXT security exit that prevents users accessing databases that they don't have authority to access.

   For information on writing your own version of FMN1SXT, see Customizing the FM/IMS security exit on page 287.

Both of these methods can also be used to control access by functions that use static PSBs.

Controlling access to IMS™ subsystems and FM/IMS functions

Through the IMS™ subsystem and FM/IMS Database Access Control facility you can control access to these resources:

- The update or read-only functions
- Individual FM/IMS functions
- Individual IMS™ subsystems by the update or read-only functions
- Individual IMS™ subsystems by individual functions

A number of functions in FM/IMS are protected by default. They are listed in Table 47: Protected FM/IMS Functions on page 281. 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 IMS subsystems and FM/IMS functions access control facility on page 281.

Resource access security and AGN security considerations

Resource access security (RAS) prevents an application program that is running in a dependent region from using IMS™ resources unless it is authorized to do so.

This means that an FM/IMS function running in BMP mode is prevented from using a PSB unless the user is authorized to use that PSB. This in turn means that users cannot use a dynamic PSB unless they are authorized to use all of the dynamic PSBs that FM/IMS can use. See Declaring the dynamic PSBs on page 261.

You might have to customize the IIMS and JIMS class profiles to provide FM/IMS users with the access they require. No special customization is required for FM/IMS to be able to access subsystems that use RAS.

Before IMS™ Version 9, Application Group Names (AGNs) were used to secure dependent regions. In IMS Version 9 RAS superseded AGN security. Support for AGNs was removed from IMS in Version 10.

FM/IMS still allows the user to specify an AGN on the entry panel or in batch JCL. However, if the IMS™ subsystem you are accessing is Version 10 or later, IMS ignores the specified AGN value.

If you do use FM/IMS to access IMS™ subsystems that use AGNs to secure BMP regions, you must specify that the subsystem uses AGN security when customizing the FM/IMS installation options module. You also have the option of specifying the AGNs that you want FM/IMS to use. For information on both of these tasks, see Specifying AGNS on page 271.

RACF® PADS security considerations

Program Access to Data Sets (PADS) allows users or groups of users to access data sets with higher authority than they would normally have, but only while running a specified program.

To run FM/IMS functions, users generally require more access to IMS™ data sets than they would ordinarily have. If you want this access to be restricted to when they are using FM/IMS functions, you should use PADS.

To run an FM/IMS function in DLI mode, users must be given UPDATE access to the RECON data sets, and if you want this function to use an IMS™ log, users must be given ALTER access to the IMS™ log data set.

When FM/IMS functions only require READ access to an IMS™ data set, you may choose not to restrict the access to when FM/IMS functions are used. However, when FM/IMS functions require UPDATE or ALTER access to the data set, as is the case for the RECON and IMS™ log data sets, the consequences of not restricting access can be more serious. So, access to these data sets should always be provided through PADS.

If PADS is used in the environment in which FM/IMS functions are run, you should take the following actions:

- Set PADS=Y when you customize the FM/IMS installation options module.
- For each subsystem specified in the FM/IMS installation options module, create a PDSE program library and set the DYNPSB parameter for the subsystem to the name of this library.

For more information on how to code the PADS and DYNPSB parameters, see FM/IMS options on page 469.

IMS™ subsystem access

FM/IMS allows you to control the way in which your IMS™ subsystems are accessed as described in the following sections.

For more information on these controls, see Controlling access to the subsystems on page 267 and FM/IMS options on page 469.

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.

Enforcing the usage and retention of IMS logs

FM/IMS allows you to force FM/IMS functions to use an IMS™ log when they are run in DLI mode and use a PSB that has update intent. FM/IMS also allows you to force the user to keep the log when the function ends.

Avoiding resource contention

This section describes how to avoid resource contention between FM/IMS Edit/Browse sessions and other concurrently executing processes at your installation.

When a user starts an Edit or Browse session, FM/IMS starts a batch message processing (BMP) program or a DL/I batch processing program. Ordinarily, the BMP or DL/I batch processing program continues to run until the user exits from the Edit or Browse session. These programs have periods of inactivity when they are waiting for a user response and periods of activity when they are processing the user response.

We need to consider measures you can take to reduce resource contention when the programs are active and when they are inactive. However, the latter measures are more important because, for the majority of the time, the programs just wait for a user response.

Inactive Edit and Browse sessions

Two types of resource contention are considered here:

- Lock contention
- Contention with processes that use the /DBRECOVERY command

These are discussed in turn.

Lock contention

Lock contention occurs when a concurrently executing program attempts to acquire a lock that the inactive Edit or Browse session is holding. The way to avoid it, is to get FM/IMS to release all locks before it returns control to the user.

There are two types of locks that need to be considered:

- Locks for unmodified data
- Locks protecting updates

Before returning control to the user, FM/IMS issues a RLSE call which releases all locks held for unmodified data, so we can restrict our attention to locks protecting updates.

Only Edit sessions hold locks protecting updates. If you want FM/IMS to release any locks protecting updates that the Edit session holds, before returning control to the user, then you should set AUTOSAVE=Y, EDITFREQ=1 and UAUTOSAV=N when you customize the FM/IMS installation module. You can specify these settings for some IMS™ subsystems, so all locks protecting updates 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 268 and FM/IMS options on page 469.

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 and CANCEL commands to discard their editing changes.

Contention with processes that use the /DBRECOVERY command

The contention described in this section occurs when the Edit or Browse session runs in BMP mode.

A process that uses the /DBRECOVERY command to deallocate a database cannot be run while an inactive Edit or Browse BMP is accessing this database. The process can be run when the user exits out of the Edit or Browse session. However, at most installations, processes that use the /DBRECOVERY command to deallocate databases are run at night after the user has gone home, so they won't always be available to do this.

Some utilities, such as the Online Reorganization Facility, support pausing BMPs and this allows them to use the /DBRECOVERY command to deallocate the databases that the BMPs are accessing. However, a BMP can only be paused when it issues a checkpoint, so if the inactive Edit or Browse BMP is to be paused, it would have to be awakened periodically to issue a checkpoint.

There are two installation options that you can use to stop this type of resource contention occurring - a time-based checkpointing option and a timeout option.

If the time-based checkpointing option is selected, then the Edit/Browse BMP issues checkpoints while it is waiting for a user response. With this option selected, utilities that support pausing BMPs are able to pause inactive Edit/Browse BMPs, allowing them to use the /DBRECOVERY command to deallocate the databases that the Edit/Browse BMPs are accessing.

To select the time-based checkpointing option for an IMS™ subsystem, use the CHKPINTVL parameter on the FMN1POPI macro statement to specify the time interval between checkpoints. For information on how to code this parameter, see FM/IMS options on page 469.

If the timeout option is selected, then the Edit/Browse session is timed out after a period of inactivity. With this option selected, you are able to run processes that use the /DBRECOVERY command to deallocate databases that inactive Edit/Browse BMPs are accessing, once the Edit/Browse BMPs are timed out.

To select the timeout option for an IMS™ subsystem, use the TIMEOUTI parameter on the FMN1POPI macro statement to specify the time period that a user has to respond before an Edit/Browse BMP is timed out. For information on how to code this parameter, see FM/IMS options on page 469.

Active Edit and Browse sessions

Typically locks held while the Edit or Browse session is active are not responsible for any resource contention, because the locks are usually held for a very short period of time and a very small number of resources are locked at any time. There are exceptions, however. The CHANGE ALL and REPEAT ALL commands can hold a large number of locks for long periods of time, so their usage needs to be controlled. If you want to control their usage for an IMS™ subsystem, you should set AUTOSAVE=Y, and UAUTOSAV=N and use the CHGAFREQ parameter to specify the checkpoint frequency that is to be used during Change All and Repeat All operations. For more information on how to code these parameters, see Specifying selected processing options on page 268 and FM/IMS options on page 469. Also, you can use the MAXGN parameter to limit the number of DL/I calls that can be issued when searching a database. For more information on how to code this parameter, see Controlling access to the subsystems on page 267 and FM/IMS options on page 469.

IMS™ management of ACBs

This topic describes how FM/IMS operates when IMS™ is configured to manage the runtime application control blocks (ACBs).

When management of ACBs by IMS™ is enabled:

- The IMS™ catalog is the trusted source for DBDs and PSBs.
- IMS™ mostly no longer requires DBD, PSB, and ACB libraries.

In keeping with this, FM/IMS retrieves the PSBs and DBDs from the IMS catalog.

Mostly the IMS™ Catalog API (DFS3CATQ) is used to retrieve these control blocks. However, when the control blocks are required by IMS™ utilities that FM/IMS uses to provide dynamic PSB support, the IMS™ Catalog Library Builder utility (DFS3LU00) is used to retrieve them. This is because the IMS utilities require the control blocks to be in libraries.

Dynamic PSB support when IMS™ management of ACBs is enabled

Dynamic PSBs are supported differently when IMS management of ACBs is enabled.

Table 39. IMS™ management of ACBs and dynamic PSB support

When ACBs are managed by your installation	When ACBs are managed by IMS™
IMS builds the ACB for the DOPT PSB when functions are run in DLI mode.	IMS does not build an ACB for the DOPT PSB when functions are run in DLI mode or BMP mode.
FM/IMS builds the ACB for the DOPT PSB when functions are run in BMP mode.	FM/IMS builds an ACB for the DOPT PSB when functions run in BMP mode or DLI mode.
The ACB maintenance utility uses your installation's DBD libraries.	The ACB maintenance utility uses DBD libraries that are dynamically generated by the IMS Catalog Library Builder utility.
The DOPT ACB library is concatenated with the primary ACBLIB data set in the IMS execution JCL.	The IMS Catalog Populate utility adds the DOPT PSBs to the IMS catalog.

Restrictions on FM/IMS functions when IMS™ management of ACBs is enabled

- FM/IMS does not support accessing logical databases, as the IMS Catalog API and the IMS Catalog Library Builder utility do not support retrieval of logical DBDs.
- FM/IMS adds DOPT PSBs to the IMS catalog but it does not delete them. Therefore periodically use the IMS Catalog Record Purge utility (DFS3PU10) to remove the DOPT PSBs from the IMS catalog.

Configuring FM/IMS when IMS™ management of ACBs is enabled

Set ACBMGMT=CATALOG on the FMN1POPI macro statement to specify that IMS™ management of ACBs is enabled.

Also set the following FMN1POPI parameters:

- Use the BSDSHLQ parameter to specify the high level qualifier for the IMS™ bootstrap data set.
- Use the DFSDF parameter to specify the 3-character suffix of the DFSDFxxx member of the IMS™ PROCLIB data set that contains the settings and attributes of the IMS™ catalog.

- Use the PROCLIB parameter to specify the name of the IMS™ PROCLIB data set that contains the required DFSDFxxx member.
- Use the REGCATLG parameter to specify whether the IMS catalog is registered with DBRC.

If you plan to use dynamic PSBs to access databases, use the DYNACB parameter to specify the name of the ACBLIB data set into which the ACB Maintenance utility generates the DOPT PSBs. You would also do this if the ACBs are managed by your installation, but when IMS management of ACBs is enabled, the specified data set is *required* by functions that access databases in BMP mode or in DLI mode.

For more information on how to code these parameters, see FM/IMS options on page 469.

Note: The DBDLIBn, DBRC, and PSBLIBn parameters are not used when IMS management of ACBs is enabled.

Templates

FM/IMS uses templates to format database segments into their individual fields. One template is required for each database.

Templates are used by the Edit, Browse, Extract, Print, Batch Edit, and Batch Browse functions.

Static templates

Static templates are created from COBOL copybooks or PL/I include members that define the fields of the database segments. They must be created manually and kept up to date.

Use the Template function (option 4.1), to create individual static templates from scratch or to update individual static templates. The function runs in the foreground.

Use the Template Update Utility (option 4.4), to update one or more static templates in either foreground or batch.

Use the TPLLIBn parameters on the FMN1POPI macro statement to specify the names of the data sets that contain the static templates that you want the above functions to use. If you do not want users to override the values you specify for these parameters, set UTPLLIB=N on the FMN1POPI macro statement.

Note: Static templates are simply called templates in the *File Manager for z/OS User's Guide and Reference for IMS Data*.

Dynamic templates

Installations that have added application-defined fields to their DBDs have the option of using dynamic templates. Dynamic templates are templates that FM/IMS dynamically generates from the field definitions in the DBDs.

FM/IMS gets the field definitions from the IMS catalog when ACBs are managed by IMS, or from the DBD libraries for the subsystem when ACBs are managed by your installation. The advantage of dynamic templates over static templates is that they don't have to be created manually nor do they have to be kept up to date.

To enable dynamic generation of templates, set DYNTPLT=Y on the FMN1POPI macro statement. When dynamic generation of templates is enabled, functions that have requested a new or an existing view or criteria set attempt to generate a template for the database that the function is accessing. If the function is unable to generate a dynamic template for the database, it reverts to using the static template.

The Edit, Browse, Print, Batch Edit, and Batch Browse functions use new and existing views. The Extract function uses new and existing criteria sets.

How static and dynamic templates are used

How the static or dynamic template is used depends on whether the function requests a new or an existing view or criteria set.

If the function requests a new view or criteria set, the static or dynamic template is used to format database segments into their individual fields. The function creates a temporary view or criteria set that is based on the static or dynamic template. You can save the temporary view or criteria set for future use.

If the function has requested an existing view or criteria set, how the static or dynamic template is used depends on which *usage rules* have been specified.

View and criteria set usage rules

Usage rules specify whether a view or criteria set can be used with a database and, if it can, whether the function uses the view or criteria set specified by the user or a composite view or criteria set created by adding the view or criteria set’s selection information to the static or dynamic template.

When a composite view or composite criteria set is used, the template is used to format database segments into their individual fields. Otherwise, the view or criteria set is used to format database segments into their individual fields.

You can specify one set of rules for when the function generates a template for the database and another set of rules for when it doesn’t.

When the function generates a template

The VCURUDT parameter specifies the usage rules that apply to views and criteria sets when the function generates a template for the database.

Use the VCURUDT parameter on the FMN1POPI macro statement to set one of the following values:

C

A view or criteria set can be used with a database only if it provides one or more layouts for each segment in the database. If it does provide one or more layouts for each segment in the database, the function uses a composite view or criteria set that is created by adding the view or criteria set’s selection information to the template that was generated for the database. So, the generated template is used to format the database segments into their individual fields.

R

A view or criteria set can be used with a database only if it has the same fields as the template that was generated for the database. If it does have the same fields as the generated template, the function uses the view or criteria set that is specified by the user. So, the view or criteria set is used to format the database segments into their individual fields. However, as the view or criteria set and the generated template have the same fields, this is equivalent to using the template to format the database segments into their individual fields.

VCURUDT=C is the default setting and the recommended setting.

When the function does not generate a template

The VCURULE parameter specifies the usage rules for views and criteria sets that apply when the function does not generate a template for the database.

Use the VCURULE parameter on the FMN1POPI macro statement to set one of the following values:

C

A view or criteria set can be used with a database only if it provides one or more layouts for each segment in the database. If it does provide one or more layouts for each segment in the database, the function uses a composite view or criteria set that is created by adding the view or criteria set's selection information to the static template for the database. So, the static template for the database is used to format the database segments into their individual fields.

R

A view or criteria set can be used with a database only if it has the same fields as the static template for the database. If it does have the same fields as the static template for the database, the function uses the view or criteria set that is specified by the user. So, the view or criteria set is used to format the database segments into their individual fields. However, as the view or criteria set and the static template for the database have the same fields, this is equivalent to using the template to format the database segments into their individual fields.

S

A view or criteria set can be used with a database only if it provides one or more layouts for each segment in the database. If it does provide one or more layouts for each segment in the database, the function uses the view or criteria set that is specified by the user. So, the view or criteria set is used to format the database segments into their individual fields.

The static template for the database is not used.

T

A view or criteria set can be used with a database only if both of these conditions are true:

- The view or criteria set provides one or more layouts for each segment in the database.
- The view or criteria set was created for the database OR the view or criteria set and the static template for the database have the same fields.

If the view or criteria set satisfies these conditions, the function uses the view or criteria set that is specified by the user. So, the view or criteria set is used to format the database segments into their individual fields.

Note: VCURULE=T is the default setting, but VCURULE=C is the recommended setting.

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 Customizing the operating environment for File Manager on page 20.

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 17.

Alternatives for controlling FM/IMS auditing

FM/IMS auditing is an optional facility. There is no requirement to implement it and FM/IMS works if auditing is not implemented. You should consider:

- Whether user access to IMS™ databases using File Manager IMS™ component requires auditing.
- The information that File Manager audit log records can provide.
- The information that File Manager audit log records cannot provide, and possible alternatives to obtaining that information.
- If you do decide to use File Manager auditing, how you will handle any issues associated with large audit log data sets, or additional SMF records.
- How you will use the information provided by File Manager audit log records.

If your site requires a record of a user's read access to IMS™ databases, an external security product such as RACF® can be configured to log access by some or all users, and may be a better alternative.

File Manager audit of read access to IMS™ data does not write audit log records for every segment processed, rather the name of the database and how many segments were processed are written to the audit log.

File Manager audit of changes to IMS™ data typically writes two log records, a before and after image of the segment that was changed. If you intend to log update changes to IMS™ databases that are subject to heavy update activity you need to consider the performance impact of writing many audit log records, also the size of any audit log data sets that may be produced

You have two choices as to how you control auditing of FM/IMS activities:

Use FMN1POPT controlled audit logging

This was the original method of controlling auditing and as such only provides limited functionality.

With this method, you control audit logging by specifying the required audit settings in the FM/IMS installation options module.

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.

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 implement SAF-rule controlled auditing you need to decide how File Manager auditing will be enabled. This is described in more detail in Customizing the File Manager audit facility for IMS component on page 295. There are two alternatives. One requires an enabling SAF rule and the presence of a member in SYS1.PARMLIB. The other requires an enabling SAF rule but has no requirement for a member in SYS1.PARMLIB. The use of a member in SYS1.PARMLIB provides additional facilities compared with the alternative that does not require the use of SYS1.PARMLIB. The additional facilities are documented in File Manager options specified in PARMLIB members on page 532.

When you have determined the appropriate type of auditing for your installation, follow the relevant instructions in Customizing the File Manager audit facility for IMS component on page 295.

Chapter 22. 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 (◄ⓃⓔⓌ) as shown in Figure 53: on page 260. You could add FM/IMS in your Primary Option Menu after the File Manager base function.

Figure 53.

```
#
)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        FM/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
#
```

To invoke FM/IMS with LIBDEFs, see Example 1. Invoking File Manager, FM/IMS, and FM/Db2 primary options from a selection panel on page 24.

For 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, 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 entering 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

To invoke FM/IMS with LIBDEFs, see Example 2. ISPF command table with LIBDEF invocations on page 24.

For information about ISPF command tables, see z/OS ISPF Planning and Customizing and *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 databases in BMP mode, you must declare the dynamic PSBs and 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 479).
- 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 480).

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 an ACBLIB data set into which the ACB Maintenance utility can generate DOPT PSBs. Use the DYNACB parameter on the FMN1POPI macro statement to specify the name of this data set.

When the ACBs are managed by your installation (ACBMGMT=ACBLIB), you must concatenate the specified data set with the primary ACBLIB data set in the IMS™ execution JCL (that is, the JCL that starts the IMS™ control region). The DOPT ACBLIB data set must be concatenated to both the IMSACBA and IMSACBB DDs in the IMS™ execution JCL.

When IMS™ management of ACBs is enabled (ACBMGMT=CATALOG), you do not have to concatenate the specified data set with the primary ACBLIB data set. Instead, the specified data set is used as input for the IMS Catalog Populate utility (DFS3PU00), which adds the DOPT PSBs to the IMS catalog. Note that the specified data set is also required by functions that run in DLI mode.

Chapter 23. 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.

Migration considerations

If you are migrating 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.

However, you can use the source code from which the earlier version of the FMN1POPT module was created. Add any new parameters that you need to specify, then assemble the source using the latest version of the File Manager macro library (FMN.SFMNMAC1) and lastly link-edit the module.

FM/IMS macro statements

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 File Manager options on page 383. 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 FM/IMS options on page 469.

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 FM/IMS options on page 469.

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 FM/IMS options on page 469.

FMN1END macro

The FMN1END macro statement is required. FMN1END can only be specified once and it must be the last 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 40: DLIBATCH parameters on page 265 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. Where applicable, use the parameters in column 3 to specify whether or not the user can override the values you specify.

Table 40. DLIBATCH parameters

DLIBATCH parameter	FMN1POPD/FMN1POPI parameter	Use this parameter to allow override
BKO	IMSBKO	UIMSBKO
BUF		UBUF

Table 40. DLIBATCH parameters (continued)

DLIBATCH parameter	FMN1POPD/FMN1POPI parameter	Use this parameter to allow override
DBRC	DBRC[1]	UDBRC
DFSDF	DFSDF[2]	Cannot override
GSGNAME	GSGNAME	URSR
IRLM	IRLM	UIRLM
IRLMNM	IRLMNAME	UIRLM
LOCKMAX	LOCKMAX	ULOCKMAX
TMINAME	TMINAME	URSR

Use the parameters in column 2 of Table 41: IMSBATCH parameters on page 266 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 41. 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 40: DLIBATCH parameters on page 265 and Table 41: IMSBATCH parameters on page 266 can be specified on the FMN1POPD and FMN1POPI macro statements. For detailed information on how to code these parameters, see FM/IMS options on page 469.

PARDLI considerations

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.

[1] Not used when IMS management of ACBs is enabled (ACBMGMT=CATALOG).

[2] Not used when the ACBs are managed by your installation (ACBMGMT=ACBLIB).

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 42: Control options on page 267 to control the access that FM/IMS functions have to each subsystem.

Table 42. Control options

Parameter	Description
CHKPINTVL	Issue checkpoints while Edit/Browse BMPs are waiting for a user response so that utilities that support pausing BMPs are able to pause inactive Edit/Browse BMPs.
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.
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.
TIMEOUTI	Time out Edit/Browse BMPs after a period of inactivity.
VCURULE	Control the views and criteria sets that can be used with a database in the subsystem.

Use the parameters in Table 43: IMS log usage parameters on page 268 to control the IMS™ log usage of functions which run in DLI mode and use PSBs that have update intent.

Table 43. IMS™ log usage parameters

Parameter	Description
LOGUSAGE	Whether the functions use an IMS™ log data set and, if they do use a log, whether the log is kept when the function ends.
ULOGUSAG	Whether the user can override the value that is specified for the LOGUSAGE parameter.
LOGDSN	The name pattern that the functions use to generate IMS™ log data set names.
ULOGDSN	Whether the user can override the name pattern that is specified in the LOGDSN parameter.
UIEFRDER	Whether batch functions use the IEFRDER DD in the JCL.

If you want to force FM/IMS functions to use an IMS™ log when they are run in DLI mode and use a PSB that has update intent, then specify ULOGUSAG=N, ULOGDSN=N, UIEFRDER=N and one of the following options:

LOGUSAGE=KEEP

If you want to force the user to keep the log when the function ends.

LOGUSAGE=KEEPUPD

If you want to force the user to keep the log if the function updates databases or does not end normally.

LOGUSAGE=DELETE

If you want to force the user to keep the log if the function does not end normally.

All the parameters in Table 42: Control options on page 267 and Table 43: IMS log usage parameters on page 268 can be specified on the FMN1POPD and FMN1POPI macro statements. For detailed information on how to code these parameters, see FM/IMS options on page 469.

Specifying selected processing options

Use the parameters in column 1 of Table 44: Processing options on page 268 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 44. 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

Table 44. Processing options (continued)

Parameter	Description	Use this parameter to allow override
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 44: Processing options on page 268 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 FM/IMS options on page 469.

Specifying IMS™ and FM/IMS data sets

Use the parameters in column 2 of Table 45: IMS and FM/IMS data sets on page 269 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 45. IMS™ and FM/IMS data sets

Data set	Parameter	Use this parameter to allow override
DBDLIB	DBDLIB*n*[3]	UDBDLIB
DFSVSAMP data set	DFSVSAMP	UDFSVSMP
DFSVSAMP member	VSMPMEM	UDFSVSMP
DOPT ACBLIB	DYNACB	Cannot override
Dynamic PSBLIB	DYNPSB	Cannot override

[3] Not used when IMS management of ACBs is enabled (ACBMGMT=CATALOG).

Table 45. IMS™ and FM/IMS data sets (continued)

Data set	Parameter	Use this parameter to allow override
IMS™ bootstrap	BSDSHLQ[4]	Cannot override
IMS™ macros	MACLIB	UMACLIB
IMS™ PROCLIB	PROCLIB[4]	Cannot override
RESLIB	RESLIB*n*	URESLIB
RECON		URECON
Staging ACBLIB	ACBLIB	UACBLIB
Static PSBLIB	PSBLIBn[3]	UPSBLIB
Template	TPLLIB*n*	UTPLLIB

All the parameters in columns 2 and 3 of Table 45: IMS and FM/IMS data sets on page 269 can be specified on the FMN1POPD and FMN1POPI macro statements. For detailed information on how to code these parameters, see FM/IMS options on page 469.

Specifying miscellaneous IMS™ subsystem details

Use the parameters in Table 46: Miscellaneous IMS subsystem details on page 270 to specify some miscellaneous IMS™ subsystem details that FM/IMS requires.

Table 46. Miscellaneous IMS™ subsystem details

Parameter	Description
ACBMGMT	Whether IMS™ manages the active application control blocks (ACBs) in the IMS catalog or your installation manages the ACBs in ACB libraries (ACBLIBs).
ACBSHR	Whether the ACBs that this IMS subsystem uses are also used by other IMS™ subsystems.
CATALIAS	The alias for the IMS™ catalog.
DYNPRFN	The maximum number of dynamic PSBs required by concurrent BMP users.
DYNPRFX	The first 1 - 5 characters of the dynamic PSB names used by FM/IMS functions.
DFSRRC00	The name of the program that FM/IMS attaches to invoke IMS™.
PADS	Whether or not Program Access to Data Sets (PADS) is used in the environments in which FM/IMS is run.
REGCATLG	Whether the IMS™ catalog is registered with DBRC.
UAGNS	Whether or not the subsystem uses AGNs to secure dependent regions.
XDOPTLB	When APAR PH17975 is not applied:

4. Not used when the ACBs are managed by your installation (ACBMGMT=ACBLIB).

Table 46. Miscellaneous IMS™ subsystem details (continued)

Parameter	Description
	Whether the ACB Maintenance utility deletes all PSBs and DBDs from the DOPT ACBLIB data set and makes all their space available for reuse, before building control blocks for the dynamic PSB. When APAR PH17975 is applied: This parameter is ignored. The ACB Maintenance utility deletes all PSBs and DBDs from the DOPT ACBLIB data set and makes all their space available for reuse, before building control blocks for the dynamic PSB.

All the parameters in Table 46: Miscellaneous IMS subsystem details on page 270 can be specified on the FMN1POPD and FMN1POPI macro statements except for the PADS parameter, which can only be specified on the FMN1POPD macro statement. For detailed information on how to code these parameters, see FM/IMS options on page 469.

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 FM/IMS options on page 469.

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.

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 405 for more information about the LANGUAGE parameter.

For other customization you can do for FM/IMS for national languages, refer to Customizing FM/IMS for national languages on page 308.

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. See FM/IMS options on page 469 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,                                    +      ❶
               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,                                 +      ❷
               READONLY=N,                                    +
               REGTYPES=BOTH,                                 +      ❸
               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,                                           +         IF❺❷
               DESC="Dynamic PSB only, unprotected",          +
               ACBLIB=IMSV910.IF52.ACBLIB,                    +      ❼
               DBDLIB1=FMN.IF52.DBDLIB,                       +      ❷❾
               DYNACB=IMSV910.IF52.DOPTLIB,                   +      ❽
               IRLM=Y,                                        +      ❻
               IRLMNAME=IRLM,                                 +
               MACLIB=IMS.V910.SDFSMAC,                       +      ❾
               MAXGN=(0,0),                                   +
               PSBTYPES=DYNAMIC,                              +      ❹
               RESLIB1=IMSV910.IF52.SDFSRESL,                 +      ❿
               UAUTOSAV=Y,                                    +
               UIEBFREQ=Y,                                    +
               UIMSNBA=Y,                                     +
               ULOADFRQ=Y,                                    +
               ULOCKMAX=Y,                                    +      ❺
               UPARDLI=Y,                                     +
               UPROCOPB=Y,                                    +
               UPROCOPP=Y,                                    +
               UPROCOPX=Y,                                    +
               UPROCOPY=Y,                                    +
               XDOPTLB=                                       +
*
         FMN1POPI SSID=IF42,                                  +         IF❹❷
               DESC="Static PSB only, all protected",         +
               ACBLIB=IMSV910.IF42.ACBLIB,                    +      ❶❺
               DBRC=N,                                        +      ❶❹
               DYNACB=IMSV910.IF42.DOPTLIB,                   +      ❶❻
               PSBLIB1=FMN.IF42A.PSBLIB,                      +      ❸❸❸
               PSBLIB2=FMN.IF42B.PSBLIB,                      +
               PSBTYPES=STATIC,                               +      ❶❶
               RESLIB1=IMSV910.IF42.SDFSRESL,                 +      ❶❼
               RESLIB2=IMSV910.IF42.USERLIB,                  +      ❶❽
               UACBLIB=N,                                     +      ❶❷
               UAUTOSAV=(,N),                                 +      ❶❸
               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,                                  +         IF❸❷
               DESC="Dynamic allocation ds only",             +
               ACBLIB=IMSV810.IF32.ACBLIB,                    +
               AUTOSAVE=N,                                    +      ❷❷❷
               CHGAFREQ=(44,44),                              +
               DBRC=Y,                                        +      ❷❶
               DFSVSAMP=IMSV810.IF32.PROCLIB,                 +
               DYNACB=IMSV810.IF32.DOPTLIB,                   +
               DYNALLOC=Y,                                    +      ❶❾
               RESLIB1=IMSV810.IF32.SDFSRESL,                 +
               RESLIB2=,                                      +
               TPLLIB1=IFA2.XXX1.TEMPLATE,                    +      ❸❶
               TPLLIB2=IFA2.XXX2.TEMPLATE,                    +
               TPLLIB3=IFA2.XXX3.TEMPLATE,                    +

               UACBLIB=N,                                     +
               UAGNS=Y,                                       +      ❷❷
               XDOPTLB=Y                                      +      ❷❸
*
         FMN1POPI SSID=IFA2,                                  +         IFA❷
               DESC="BMP only system",                        +
               ACBLIB=IMSV11.IFA2.ACBLIB,                     +
               DYNACB=IMSV11.IFA2.DOPTLIB,                    +
               IMSAUDLG=Y,                                    +      ❷❻
               MACLIB=IMS.V10.BETA.SDFSMAC,                   +
               REGTYPES=BMP,                                  +      ❷❹
               RESLIB1=IMSV11.IFA2.SDFSRESL,                  +
               RESLIB2=IMSV11.IFA2.USERLIB,                   +
               UAGNS=Y                                        +      ❷❺
*
         FMN1POPI SSID=IFB2,                                  +         IFB❷
               DESC="READONLY system",                        +
               READONLY=Y                                     +      ❷❼
*
         FMN1AGNT SSID=IF52,AGN=FMN52                         +      ❷❽
         FMN1AGNT SSID=IFA2,AGN=FMNA0001,DESC="HR"            +
         FMN1AGNT SSID=IFA2,AGN=FMNA0002,DESC="Finance"       +
*
         FMN1END                                                        END
```

Note:

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 **IF❺❷**. 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" [❹].
3. To unprotect all parameters that have a default value of "protect" all of the usage parameters must be set to "Y" [❺].
4. Furthermore this subsystem will use IRLM [❻].
5. All data set names for this subsystem are different from those that are set as default and as such need to be specified [❼, ❽, ❾, ❿].
6. When this one parameter is included [❷❾], 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 **IF❺❷** 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 **IF❹❷**. 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" [❶❶].
3. To protect all parameters that have a default value of "unprotect" all of the usage parameters must be set to "N" [❶❷] or "(,N)" [❶❸], depending on whether or not that specific parameter has one or two values.
4. Furthermore this subsystem will not use DBRC [❶❹]. 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 [❶❺, ❶❻, ❶❼, ❶❽].
6. When these parameters [❶❺, ❶❻] 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 **IF❸❷**. 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" [❶❾].
3. When these parameters [❸❶] are included, they override all six TPLLIB parameters set up in FMN1POPD.
4. The AUTOSAVE option is turned off [❷❷❷].

5. DBRC is set to "Yes" [❷❶], but is not needed as the default value is "Yes" also.
6. This subsystem will use AGNs [❷❷].
7. When dynamic PSBs are used, the DOPT ACBLIB is cleared of all members. This is done as XDOPTLB is set to "Y" [❷❸].

IFA2:

1. These statements define an IMS™ subsystem, called **BMP only system**, as specified on the DESC keyword. The SSID is **IFA**❷. 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" [❷❹].
3. This subsystem will use AGNs [❷❺].
4. To force the use of audit logging, the parameter IMSAUDLG is set to "Y" [❷❻].

IFB2:

1. These statements define an IMS™ subsystem, called **READONLY system**, as specified on the DESC keyword. The SSID is **IFB**❷. 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" [❷❼].

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.

FMN1FTEX

A sample job control skeleton for FM/IMS is in FMN.SFMNSLIB(FMN1FTEX). You might need to change the following DD statements in the sample skeleton:

1. The STEPLIB DD statement specifies DSN=&FMNSMOD1. FM/IMS sets &FMNSMOD1 to either the value specified for the SFMNMOD1 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 SFMNMOD1 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 32 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 33.)

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.

Customizing for DEDB randomizing modules

If your installation uses DEDB randomizing modules that abend or cause an IMS™ abend, you can create your own FMN1RNDM load module.

In the FMN1RNDM load module, you provide the names, or name patterns, of the DEDB randomizing modules at your installation that abend, or cause an IMS™ abend, when they receive key field values that are not valid packed decimal numbers.

If the key field of the root segment of a DEDB database is defined as TYPE=P in the DBD, and the name of the randomizing module that the DEDB database uses is included in the FMN1RNDM load module, then FM/IMS checks that the root key values specified in DL/I calls are valid packed decimal numbers before issuing the calls.

Providing your own FMN1RNDM module

FM/IMS does not provide a default FMN1RNDM module. You can create your own using the sample FMN1RNDM and usermod FMN1UMD1.

To create your own FMN1RNDM module:

1. Copy the member FMN1RNDM from FMN.SFMNSAM1 to your own source library.
2. Code the names of the DEDB randomizing routines that you want to include, on the FMN1RAND RANDNAME statements, in your copy of FMN1RNDM. You can provide as many statements as you need. You can specify wildcards, using * (multiple characters) and % (single character).
3. Modify the FMN1UMD1 member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about the changes you need to make.
4. Install SMP/E usermod FMN1UMD1.

Chapter 24. Customizing the FM/IMS security environment

This chapter describes:

- How you can control access to databases when functions are run in BMP mode.
- 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.

The Database Access Control facility

Use the Database Access Control facility to control users' access to databases when running functions in BMP mode.

Depending on which profiles you define, access checking may be performed when the function uses a dynamic PSB, when the function uses a static PSB, or when the function uses both a dynamic PSB and a static PSB.

To activate access checking for functions using a dynamic PSB, do the following:

1. Define a FACILITY class profile named `FILEM.IMS1.DBDYNAM`
2. Give all users whose access you want checked READ access to this resource.

To activate access checking for functions using a static PSB, do the following:

1. Define a FACILITY class profile named `FILEM.IMS1.DBSTATIC`
2. Give all users whose access you want checked READ access to this resource.

Once access checking is activated, FM/IMS issues RACROUTE calls to determine whether the user is authorized to access the database specified on the entry panel or in the batch JCL. The RACROUTE call is for the XFACILIT class resource `FILEM.IMS1.`*`ssid`*`.DB.`*`dbname`*, where *ssid* is the IMS™ subsystem name and *dbname* is the name of the database that the user is attempting to access.

- For access by a read-only function, a user requires a minimum of READ access.
- For access by an update function, a user requires a minimum of UPDATE access.

Create corresponding profiles to protect this resource.

RACF® examples

1. Activating access checking for functions using a dynamic PSB.

   With these commands, checking is activated for all users:

   ```
   RDEFINE FACILITY FILEM.IMS1.DBDYNAM AUDIT(NONE) +
           UACC(READ) OWNER(ownerid)
   SETROPTS RACLIST(FACILITY) REFRESH
   ```

2. Activating access checking for functions using a static PSB.

   With these commands, checking is activated for all users:

```
RDEFINE FACILITY FILEM.IMS1.DBSTATIC AUDIT(NONE) +
        UACC(READ) OWNER(ownerid)
SETROPTS RACLIST(FACILITY) REFRESH
```

3. Ensure users do not obtain access to databases through profiles that are less specific than the profile FILEM.IMS1.*.DB.*

```
RDEFINE XFACILIT FILEM.IMS1.*.DB.* AUDIT(NONE) +
        UACC(NONE) OWNER(ownerid)
SETROPTS RACLIST(XFACILIT) REFRESH
```

4. Allow user *dbadmin* to update all databases in subsystem IF52, and allow all other users to read these databases.

```
RDEFINE XFACILIT FILEM.IMS1.IF52.DB.* AUDIT(NONE) +
        UACC(READ) OWNER(ownerid)
PERMIT FILEM.IMS1.IF52.DB.* CLASS(XFACILIT) +
       ID(dbadmin) ACC(UPDATE)
SETROPTS RACLIST(XFACILIT) REFRESH
```

5. Allow user *fmuser1* to update database DJ1E in subsystem IF52, and allow all other users to read this database.

```
RDEFINE XFACILIT FILEM.IMS1.IF52.DB.DJ1E AUDIT(NONE) +
        UACC(READ) OWNER(ownerid)
PERMIT FILEM.IMS1.IF52.DB.DJ1E CLASS(XFACILIT) +
       ID(fmuser1) ACCESS(UPDATE)
SETROPTS RACLIST(XFACILIT) REFRESH
```

IMS™ subsystems and FM/IMS functions access control facility

FM/IMS allows you to control which IMS™ subsystems a user can access when using each of the functions listed in Table 47: Protected FM/IMS Functions on page 281. These functions are protected by default when you receive FM/IMS.

Table 47. 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

Table 47. Protected FM/IMS Functions (continued)

Function code	Description	UPDATE or 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

You can grant or deny some or all users access to:

1. Individual IMS™ subsystems by individual functions in Table 47: Protected FM/IMS Functions on page 281.
2. Individual functions in Table 47: Protected FM/IMS Functions on page 281. 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.

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® 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® (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".

Note:

1. The FMNSECUR module is not used (even if present) if SAF with RACF® or an equivalent security product is active when an FM/IMS function is started.
2. FM/IMS functions that are not listed in Table 47: Protected FM/IMS Functions on page 281 cannot be protected by RACF® (or an equivalent security product) or by the FMNSECUR exit.

The rest of this section describes how you implement security controls in RACF® (or an equivalent security product) for the functions in Table 47: Protected FM/IMS Functions on page 281.

Controlling access to the update or read-only functions

The update functions in Table 47: Protected FM/IMS Functions on page 281 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.

Enter the following RACF® commands to define these profiles in the FACILITY class:

```
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 user ID *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 user ID *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 47: Protected FM/IMS Functions on page 281 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 47: Protected FM/IMS Functions on page 281 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 47: Protected FM/IMS Functions on page 281, you will need to define the aforementioned profiles.

Enter the following RACF® commands to define these profiles in the FACILITY class:

```
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 283.

Controlling access to individual functions

Individual functions in Table 47: Protected FM/IMS Functions on page 281 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 47: Protected FM/IMS Functions on page 281, you will need to define the profiles for these functions.

Enter the following RACF® command to define these profiles in the FACILITY class:

```
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 47: Protected FM/IMS Functions on page 281 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 47: Protected FM/IMS Functions on page 281, you will need to define the aforementioned profiles.

Enter the following RACF® command to define these profiles in the FACILITY class:

```
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 283.

What governs whether access is granted or denied

Access to a subsystem *ssid* by an *update* function in Table 47: Protected FM/IMS Functions on page 281 with function code *fc* is governed by the first profile in this list

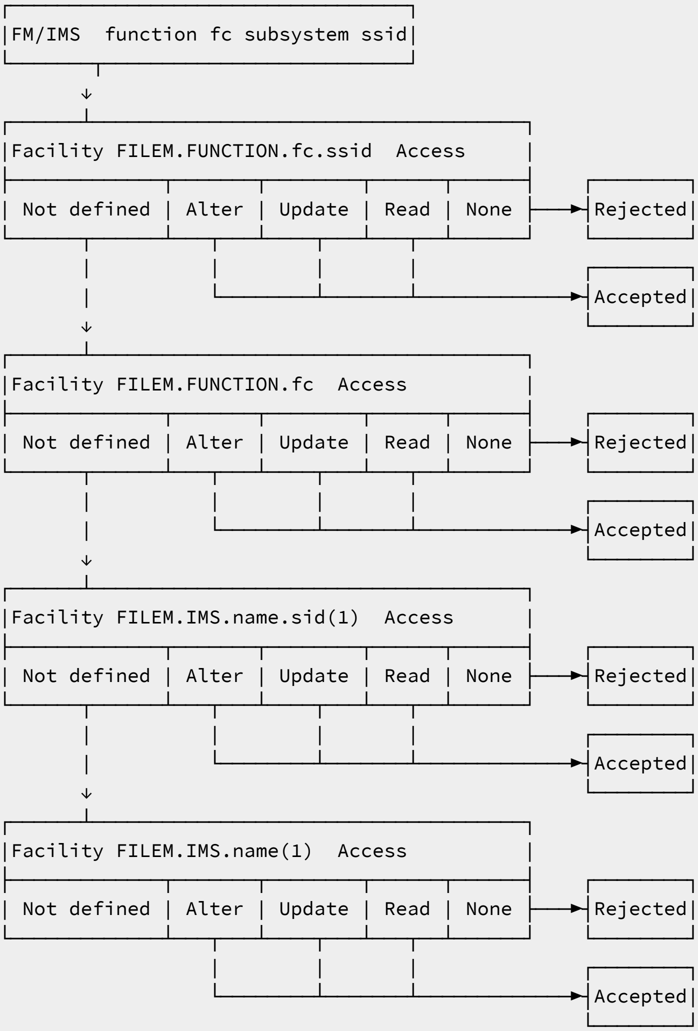
- FILEM.FUNCTION.*fc*.*ssid*
- FILEM.FUNCTION.*fc*
- FILEM.IMS.UPDATE.*ssid*
- FILEM.IMS.UPDATE

Access to a subsystem *ssid* by a *read-only* function in Table 47: Protected FM/IMS Functions on page 281 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 54: Security checking for FM/IMS functions on page 286 illustrates the security checking that FM/IMS performs when a function in Table 47: Protected FM/IMS Functions on page 281 attempts to access an IMS™ subsystem.

Figure 54. Security checking for FM/IMS functions



Note:

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.ssid*, the first profile in the list. If this profile is not defined RACF® returns RC=4. If RC=4 is returned, FM/IMS issues a RACROUTE for the second, third, and fourth profile in the list, in turn, until RACF® returns something other than RC=4.

Note that not all non-IBM security products issue RC=4 when a RACROUTE TYPE=AUTH is issued for a profile that is not defined; in this case FM/IMS only issues a RACROUTE for FILEM.FUNCTION.*fc.ssid* and, if this profile has not been defined, the request fails immediately. If this applies to your security product, your only option is to use the FILEM.FUNCTION.*fc.ssid* profiles to control access to your FM/IMS functions and IMS™ subsystems.

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 version of FMN1SXT that is distributed with FM/IMS performs no security checking. Each exit type returns control immediately, thus allowing normal processing to continue.

You can provide your own version of FMN1SXT using either of the sample source decks, FMN1XITA (High Level Assembler) or FMN1XITC (COBOL), as a base. FMN1XITA and FMN1XITC are distributed in FMN1.SFMNSAM1.

Tip: Do not use ISPLINK calls to invoke ISPF services from an FMN1SXT security exit. This is because the ISPF environment does not exist when FM/IMS is invoked from batch and IPV clients.

Types of security exits

This topic describes what you can use each security 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 (FM_SECURITY_PARAMETERS, FM_SECURITY_WORKAREA, FM_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 48. 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 user ID
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 49. 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.

Table 49. Parameters - Exit Type A. (continued)

Field	Update	Size	Description
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 50. 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 51. 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:

Table 51. Parameters - Exit Type I (continued)

Field	Update	Size	Description
			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.

Table 51. Parameters - Exit Type I (continued)

Field	Update	Size	Description
PSB Type	N	Char(1)	**S** Static **D** Dynamic
PSB Name	N	Char(8)	The name of the Program Specification Block (static 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 25. Customizing the File Manager audit facility for IMS™ component

FM/IMS can optionally write audit log records to either SMF or an audit log data set.

If auditing is not required:

- Set IMSAUDLG=N in the FMN1POPD macro specification of the FMN1POPT module. See AUDITLOG on page 388.
- Set SMFNO=0 in the FMN0POPI macro specification of the FMN1POPT module. See SMFNO on page 421 for more information.
- Skip the customization described in the rest of this chapter.

File Manager provides two different methods (FMN1POPT-controlled and SAF-controlled audit) for controlling whether audit records are written for File Manager IMS™ component. These are described in detail in Alternatives for controlling FM/IMS auditing on page 257.

You should determine which of the two methods is appropriate for your site's requirements.

Use the checklist to determine the customization required for the File Manager IMS™ audit facility.

Table 52. Checklist for customizing FM/IMS auditing. This table lists choices and decisions.

Audit customization choice	Decision (Yes|No|Not applicable)
1. Control auditing using the FMN1POPT options module	
2. Control auditing using SAF rules and a member in SYS1.PARMLIB	
3. Control auditing using SAF rules, without any changes to SYS1.PARMLIB	
4. Audit records are to be written to a data set	
5. Audit records are to be written to SMF	

For choices 1 to 3, you should answer YES for one choice only. Mark the other two choices as 'Not applicable'.

If you answer YES for choice 1, you can answer YES for one of choices 4 or 5. Mark the other choice as 'Not applicable'.

If you answer YES for choices 2 or 3, you can answer YES for one or both of choices 4 and 5. If you answer YES for both choices you are implementing dual-logging.

Table 53. Customization steps for audit customization choices. This table lists choices and related actions.

Customization choice	Sections to complete
1. Control auditing using the FMN1POPT options module	• FMN1POPT-controlled audit logging on page 296
2. Control auditing using SAF rules and a member in SYS1.PARMLIB	• SAF-controlled auditing for File Manager IMS component on page 298 • Implementing SAF-rule controlled auditing on page 301
3. Control auditing using SAF rules, without any changes to SYS1.PARMLIB	• SAF-controlled auditing for File Manager IMS component on page 298 • Implementing SAF-rule controlled auditing on page 301
4. Audit records are to be written to a data set	• Audit data set configuration on page 297
5. Audit records are to be written to SMF	• Customizing File Manager to write audit records to SMF on page 79

FMN1POPT-controlled audit logging

This was the original method of controlling auditing and, as such, only provides limited functionality.

With this method, you control audit logging by specifying the required values for the IMSAUDLG parameter on the FMN1POPI macro statement and the SMFNO parameter on the FMN0POPI macro statement.

You use the IMSAUDLG parameter to specify whether or not auditing is enforced for FM/IMS Edit and, if it is, whether or not an audit report is generated at the end of the Edit session. For information on how to code the IMSAUDLG parameter, see IMSAUDLG on page 483.

You use the SMFNO to specify whether the audit records are written to SMF or an audit log data set, and, if they are written to SMF, the SMF record type. For information on how to code the SMFNO parameter, see SMFNO on page 421.

If auditing of an Edit session is not enforced, users can still get FM/IMS to create an audit trail of their updates, by selecting the Create audit trail option on the Edit Entry panel.

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 is determined by skeleton member FMN1FTAD found in FMN.SFMNSLIB. 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 Users Guide and Reference*. For more information, see IMSAUDLG on page 483.

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.

Audit data set configuration

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 386 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

Any 1-8 character constant that is valid in the context of a data set name.

userid

The user ID creating the data set

Dyymmdd

The date of the activity.

Thhmmss

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 is null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix is also 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.

Set the AUDITHLQ parameter in the FMN0POPI macro for the FMN1POPT to the required value, based on the above information and your site's requirements.

You can print the information in a File Manager IMS™ audit data set using the FM/IMS Print Audit Trail utility. To do this select option 3.8 from the FM/IMS Primary Option Menu.

SAF-controlled auditing for File Manager IMS™ component

There are two methods for implementing SAF-controlled auditing for File Manager IMS™ component. These are:

1. Control auditing for File Manager IMS™ component using an enabling SAF Facility class rule and a member in SYS1.PARMLIB.

   To use this method complete the customization described in SAF-controlled auditing using SYS1.PARMLIB on page 298.
2. Control auditing for File Manager IMS™ component using an enabling SAF Facility class rule, without any changes to SYS1.PARMLIB.

   To use this method complete the customization described in SAF-controlled auditing without SYS1.PARMLIB on page 300.

! **Important:** If you use a security product other than RACF®, review the information in When a security product other than RACF is in use on page 87 to avoid one possible cause of S047 abends.

SAF-controlled auditing using SYS1.PARMLIB

You need to define an enabling SAF facility profile as described below:

Define SAF facility profile

```
FILEM.PARMLIB.IMS
```

and ensure all File Manager IMS™ users to be audited have at least read access to that facility. See the example below:

Example

User PROD1 to have SAF-rule controlled auditing using SYS1.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)
```

Add member FMN1PARM to SYS1.PARMLIB (or any other library in the logical parmlib concatenation). See Defining the FMN1PARM member on page 299.

Once the above SAF rule is defined and activated, auditing for File Manager IMS™ component users is controlled by the FMAUDIT parameter in the FMN1PARM member. See FM/IMS options specified in FMN1PARM on page 538 for more information. If audit log records are to be written to SMF, the SMF record number is specified as an FMAUDIT parameter option. See FMAUDIT on page 539, and SMF_NO on page 539.

Note: File Manager IMS™ component does not start if a user has read access to the above facility and the FMN1PARM member does not exist in the logical parmlib concatenation.

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.

Defining the FMN1PARM member

If auditing is to be controlled from parmlib (user has read access to FILEM.PARMLIB.IMS, see SAF-controlled auditing for File Manager IMS component on page 298), 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 a FMSECRTY statement. This option is not used at present, and can be either omitted, or commented out. It has no effect.

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).

Note:

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.
   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
   ```
2. Add member FMN1PARM to the new library, specifying the appropriate FMAUDIT parameter.
3. 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 on page 300. If the FMN1PARM member specifies any include statements (see Facilities for customizing the FMN1PARM definitions on page 540), all of the included members must also reside in the same library.

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 FM/IMS options specified in FMN1PARM on page 538.

SAF-controlled auditing without SYS1.PARMLIB

You need to define an enabling SAF facility profile as described below:

Define SAF facility profile

```
FILEM.SAFAUDIT.IMS
```

and ensure that all File Manager IMS™ users to be audited have at least read access to that facility. See the example below.

Example

User PROD2 to have SAF-rule controlled auditing without using SYS1.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)
```

If you use this method and intend to write audit records to SMF, the required SMF number is specified in the FMN1POPT module. See Customizing File Manager to write audit records to SMF on page 79 for more information.

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 302.
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 303.
   - Controlling whether an audit trail is created on page 305.
   - Controlling if users can request an audit trail when one is not required (Edit function only) on page 306.
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 304.
   - FILEM.AUDIT1.ssid.TODSN on page 304.
   - FILEM.AUDIT1.ssid.fc.db. on page 306.
   - FILEM.AUDIT1.ssid.OPTION on page 306.

   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:

- SAF-controlled auditing for File Manager IMS component on page 298.
- FM/IMS options specified in FMN1PARM on page 538.

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:
   - SAF-controlled auditing for File Manager IMS component on page 298.
   - FM/IMS options specified in FMN1PARM on page 538.

Understanding how SAF controls FM/IMS audit logging

Table 54: FM/IMS functions that support audit logging when audit logging is controlled by SAF on page 302 lists the FM/IMS functions that can be made to create an audit trail when audit logging is controlled by SAF.

Table 54. 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 54: FM/IMS functions that support audit logging when audit logging is controlled by SAF on page 302 are started, SAF is invoked to answer these audit queries:

- Whether 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 the user's audit log data set should be printed at the end of the Edit/Browse session.
- For Edit function only, whether 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 55: SAF profiles that control FM/IMS audit logging on page 303 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 55: SAF profiles that control FM/IMS audit logging on page 303 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 55. SAF profiles that control FM/IMS audit logging

SAF profile	Class	Description
FILEM.AUDIT1.*ssid*.TOSMF	FACILITY	Controls whether audit log records are written to SMF.
FILEM.AUDIT1.*ssid*.TODSN	FACILITY	1. Controls whether audit log records are written to the user's audit log data set. 2. Controls whether 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 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 audit logging is required. 2. Controls whether 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 54: FM/IMS functions that support audit logging when audit logging is controlled by SAF on page 302 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 user ID *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 user ID *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 user ID *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 an audit trail is created

When a function in Table 54: FM/IMS functions that support audit logging when audit logging is controlled by SAF on page 302 accesses a given database in a given IMS™ subsystem, three profiles control whether 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 303 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 user ID *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.

Chapter 26. 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, you will need to perform **all** of the customization tasks listed in Table 56: Summary of steps for customizing FM/IMS for a national language on page 308.

Table 56. Summary of steps for customizing FM/IMS for a national language

Step	Description
__ 1	Set the LANGUAGE option for batch processing, if required. See Setting the default national language on page 272.
__ 2	Change the TERMTYPE option in FMN1POPT, if you are using a DBCS language. See TERMTYPE on page 423.
__ 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. See Changing the print and display translation tables for languages other than English on page 308.
__ 5	Translate the File Manager message text to your language. See Translating the message text on page 309.
__ 6	Provide a version of FMN1MENU for your language. See Providing a multicultural version of FMN1MENU on page 309.
__ 7	Translate the FM/IMS ISPF messages to your language. See Translating the ISPF messages text on page 310.
__ 8	Translate the FM/IMS panels to your language. See Translating the panel text on page 311.

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

If you plan to use FM/IMS 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 98.

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.

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

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 311.

To provide translated versions of the messages, you must provide a version of FMN1MENU in your language, as described in Providing a multicultural version of FMN1MENU on page 309 **and** provide translated versions of the appropriate members in FMN.SFMNMENU, as described in Translating the ISPF messages text on page 310.

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 multicultural 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 on page 311. 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 ISPF 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 on page 311.

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 57: Keyword values for the LANGUAGE option on page 311. See Customizing the FM/IMS installation options module on page 263 for information on how to do this.

Table 57. Keyword values for the LANGUAGE option

Language	Code	Specify on LANGUAGE option…
French	FRA	FRENCH
German	DEU	GERMAN
Italian	ITA	ITALIAN
Japanese	JPN	JAPANESE

Table 57. Keyword values for the LANGUAGE option (continued)

Language	Code	Specify on LANGUAGE option…
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 103.

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 104.

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.

Chapter 27. Verifying the customization of FM/IMS

After you have completed the initial installation and customization of FM/IMS, you can perform the following steps to verify your installation, before completing the installation process. You might need to refer to the *File Manager for z/OS 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.

1. Following the instructions in the sample job, make the required changes to the FMN1IVP1 job.
2. Run the FMN1IVP1 job. The expected return code is zero.

Note: View and Template data sets are empty.

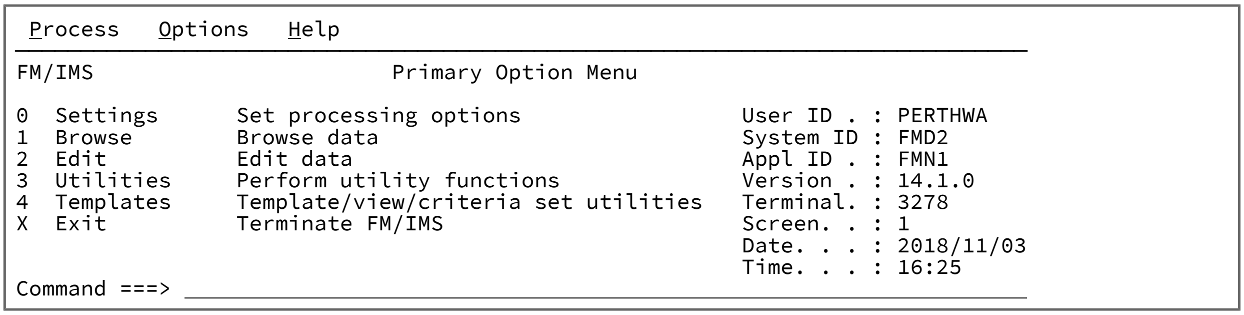
Step 2. Start FM/IMS

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 260), 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 260) 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.

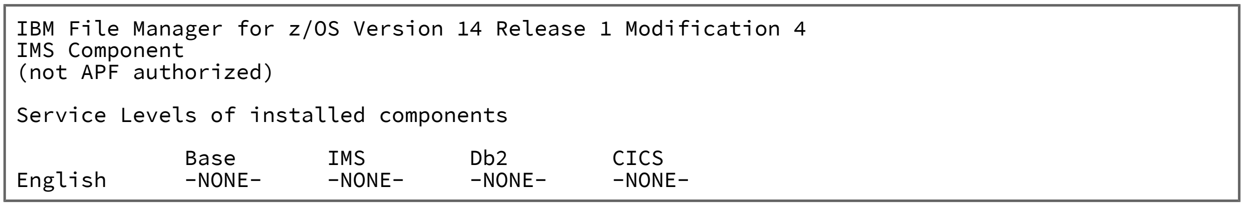
Figure 55. Primary Option Menu



```
 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 . : 14.1.0
4  Templates     Template/view/criteria set utilities  Terminal. : 3278
X  Exit          Terminate FM/IMS                      Screen. . : 1
                                                       Date. . . : 2018/11/03
                                                       Time. . . : 16:25

Command ===> _________________________________________________________________
```

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 14 Release 1 Modification 4
IMS Component
(not APF authorized)

Service Levels of installed components

             Base       IMS        Db2        CICS
English      -NONE-     -NONE-     -NONE-     -NONE-
```

Note:

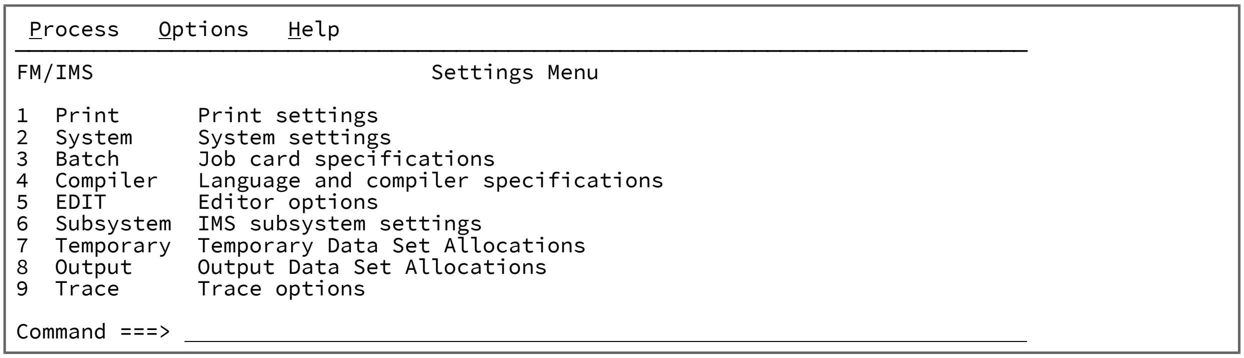
1. This will always show FM/IMS as `not APF authorized`, even if you have made File Manager APF-authorized, as File Manager cannot run APF-authorized under ISPF.
2. When you first install File Manager, `-NONE-` will be shown against each component. Subsequently, when you have applied service to File 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 is displayed.

Figure 56. Settings Menu

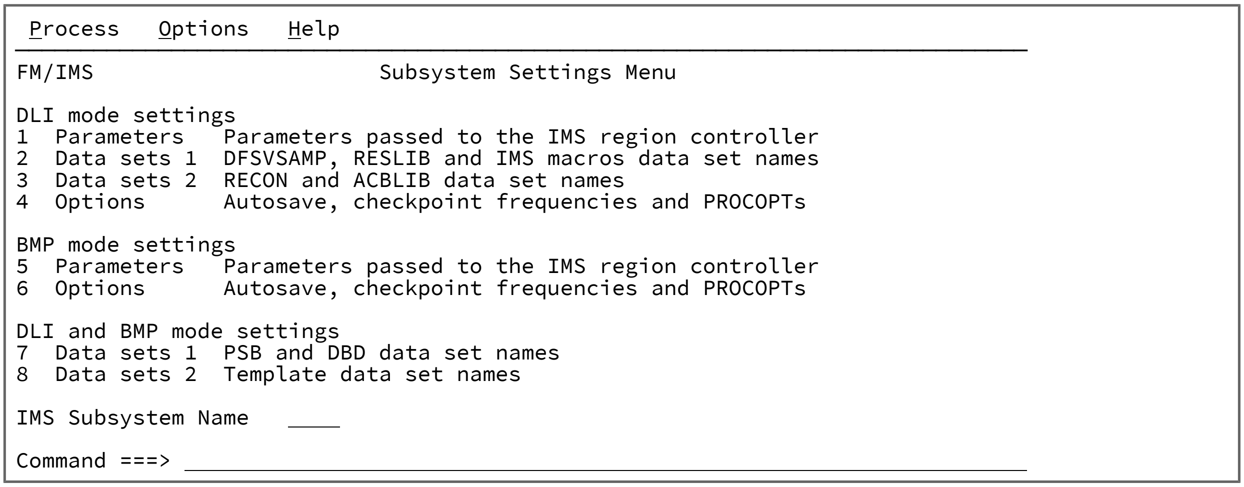


```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
FM/IMS                          Settings Menu

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 ===> _________________________________________________________________
```

Step 4. Display DLI mode settings menu

Enter **6** on the command line for Subsystem settings, and press Enter. The Subsystem Settings Menu is displayed.

Figure 57. Subsystem Settings Menu

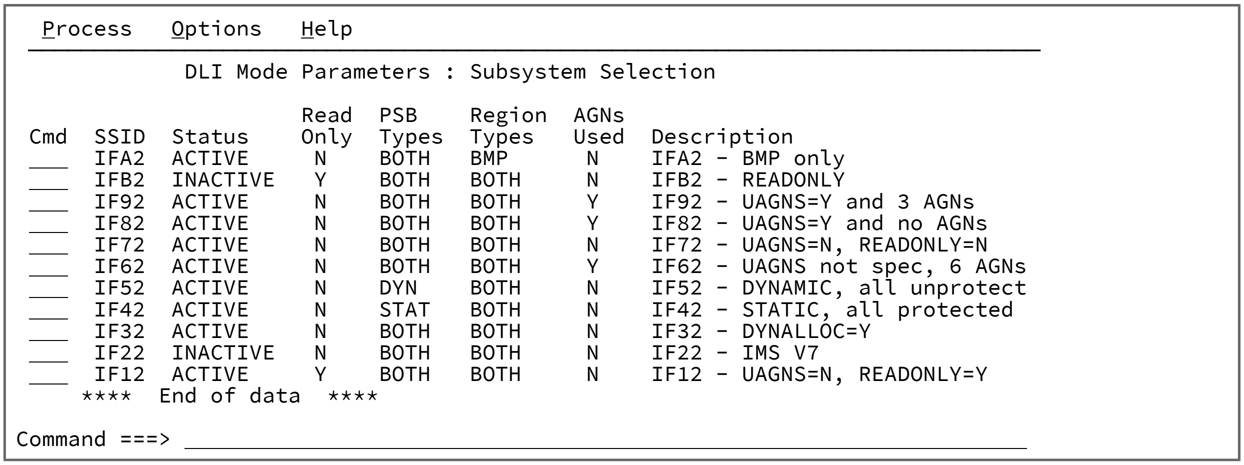


```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
FM/IMS                      Subsystem Settings Menu

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 ===> _________________________________________________________________
```

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 is displayed.

Step 5. Verify subsystem details

Figure 58. DLI Mode Parameters



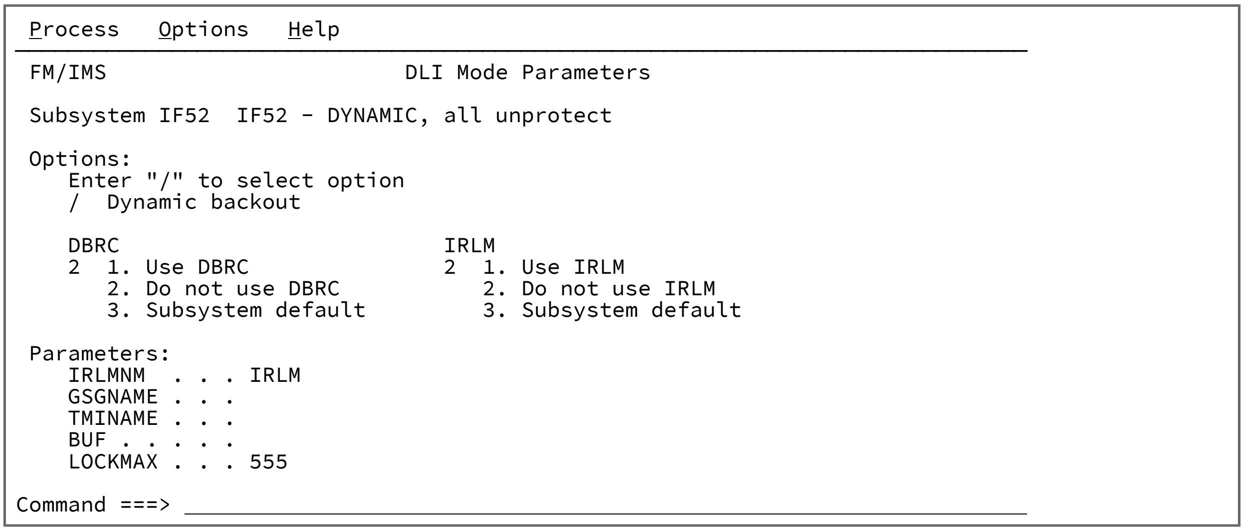
```
  Process   Options   Help
 ──────────────────────────────────────────────────────────────────────────────
             DLI Mode Parameters : Subsystem Selection

                      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 ===> _________________________________________________________________
```

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 is displayed.

Step 6. Verify DLI mode parameters

Figure 59. 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 ===> _________________________________________________________________
```

Enter **RESET** on the command line and press Enter.

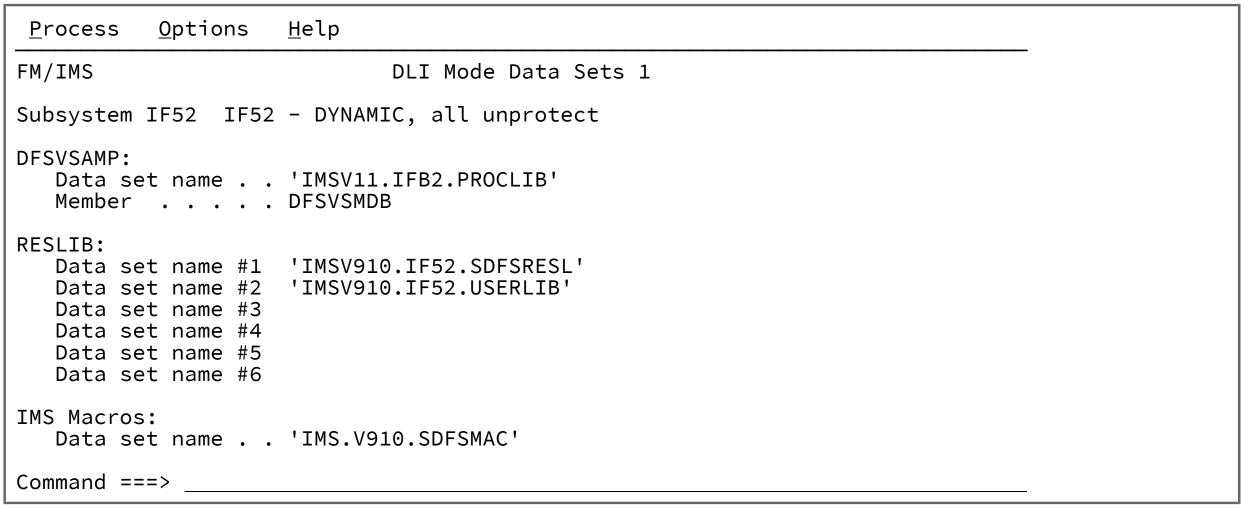
Check that the settings displayed for the subsystem are the settings you specified in the FM/IMS installation options module.

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 is displayed.

Figure 60. DLI Mode Data Sets 1



```
 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 ===> _________________________________________________________________
```

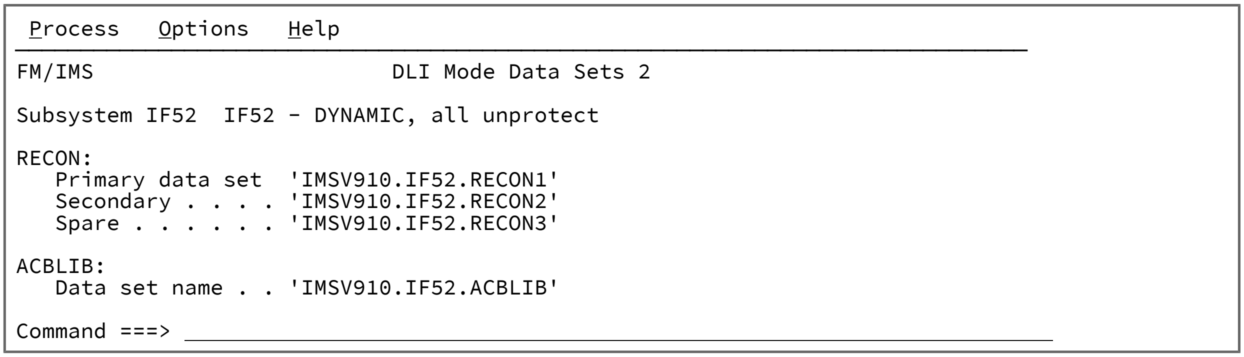
Enter **RESET** on the command line and press Enter.

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 57: Subsystem Settings Menu on page 316) is redisplayed

Enter **3** on the command line, for Data Sets 2, and press Enter. The DLI Mode Data Sets 2 is displayed.

Figure 61. DLI Mode Data Sets 2



```
 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 ===> ___________________________________________________________________
```

Enter **RESET** on the command line and press Enter.

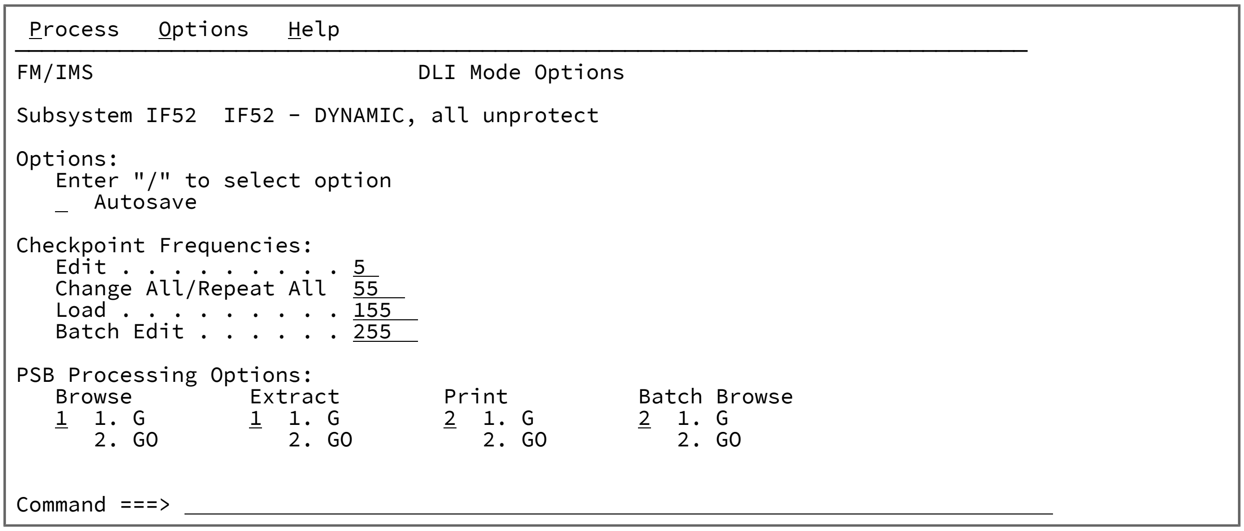
Check that the RECON data set name fields are blank. Also check that the ACBLIB data set name displayed for the subsystem is the data set name you specified in the FM/IMS installation options module.

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 is displayed.

Figure 62. DLI Mode Options



```
 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 ===>
```

Enter **RESET** on the command line and press Enter.

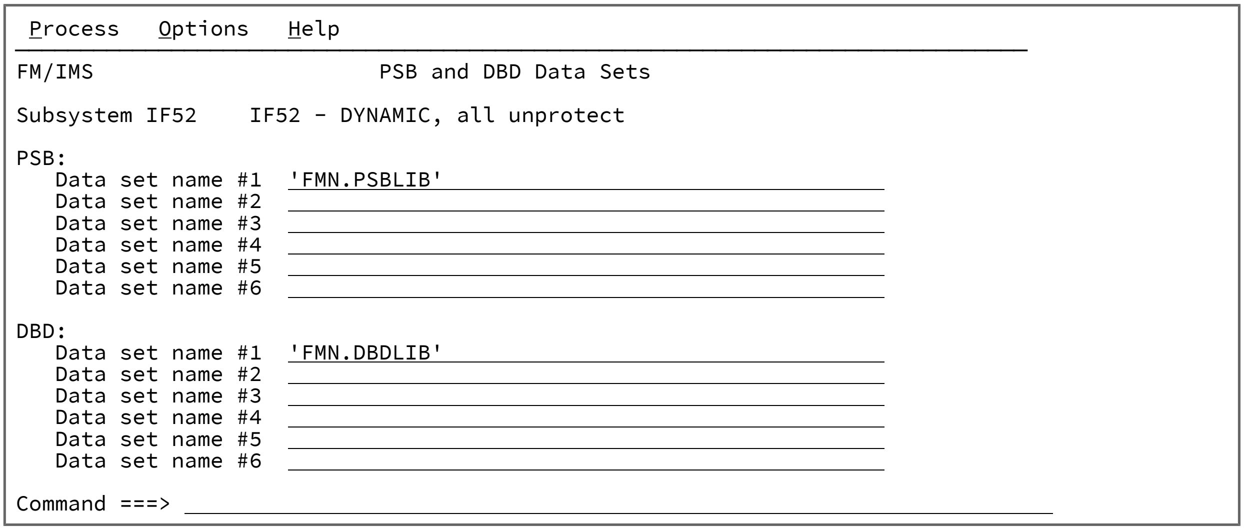
Check that the settings displayed for the subsystem are the settings you specified in the FM/IMS installation options module.

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 is displayed.

Figure 63. PSB and DBD Data Sets panel



```
 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 ===>
```

Press PF3 to save the settings in your profile and exit the panel. The Subsystem Settings Menu is again displayed.

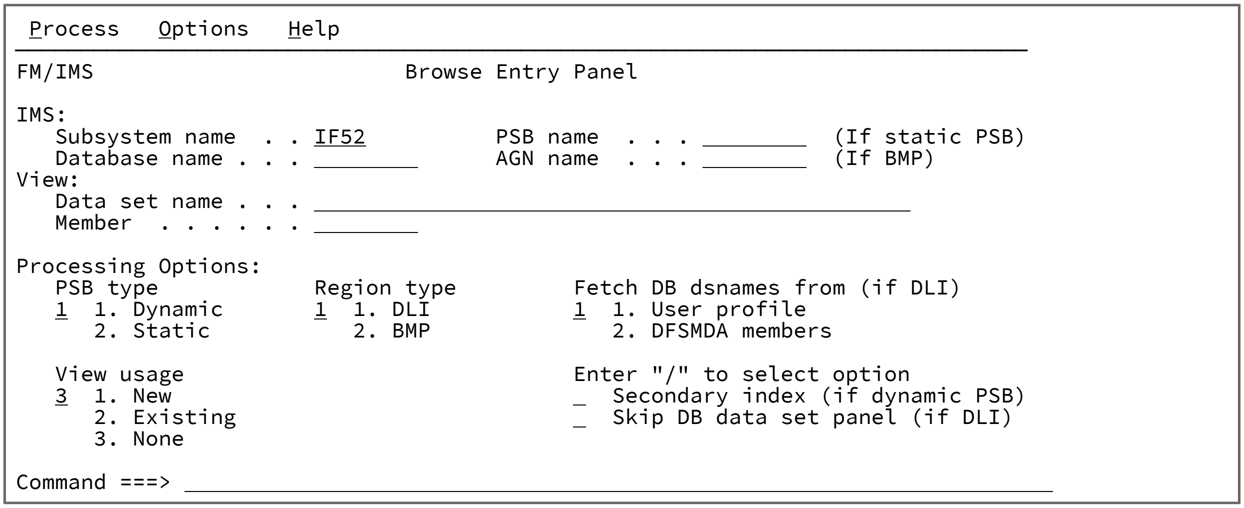
Press PF3 again to return to the Settings Menu (Figure 56: Settings Menu on page 316).

Press **PF3** again to return to the FM/IMS Primary Option Menu (Figure 55: Primary Option Menu on page 315).

Step 10. Start the browse dialog

Enter **1** on the command line, for Browse data, and press Enter. The Browse Entry Panel (Dynamic PSB) is displayed.

Figure 64. Browse Entry Panel (Dynamic PSB)



```
 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 ===> ____________________________________________________________________
```

The subsystem name you selected in Step 5. is displayed in the IMS™ **Subsystem name** field.

Enter **DJ1E** in the **Database name** field.

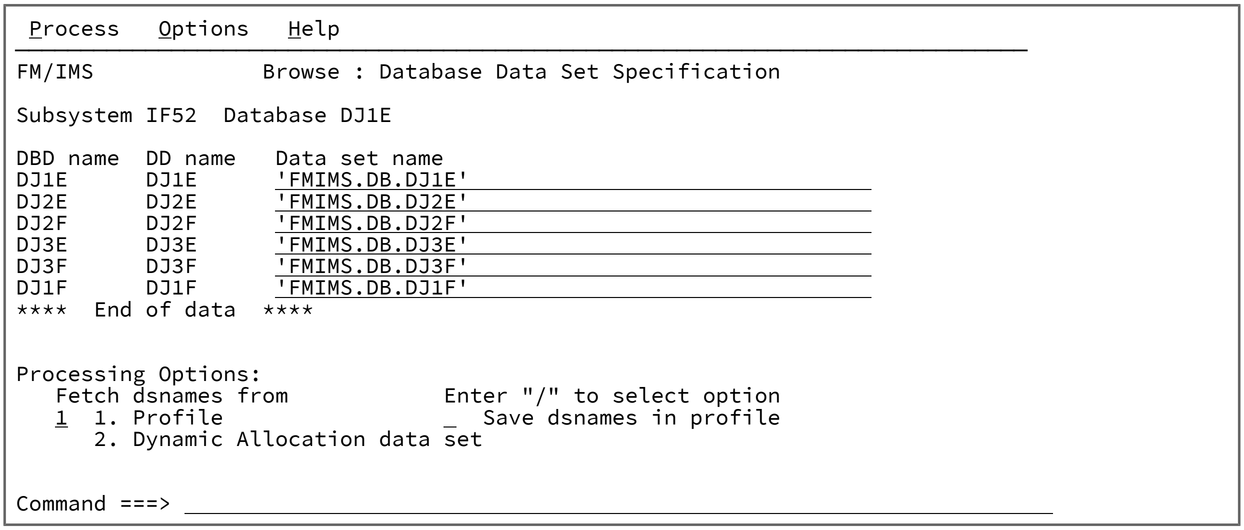
Enter **1** in the **PSB type** field, **1** in the **Region type** field, **2** in the **Fetch DB dsnames from (if DLI)** field, a space in the **Secondary index (if dynamic PSB)** field, **3**in the **View usage** field, and press Enter.

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 65: Browse : Database Data Set Specification on page 321 shows the Browse : Database Data Set Specification panel. Figure 66: Browse : Database Data Set Display on page 321 shows the Browse : Database Data Set Display panel.

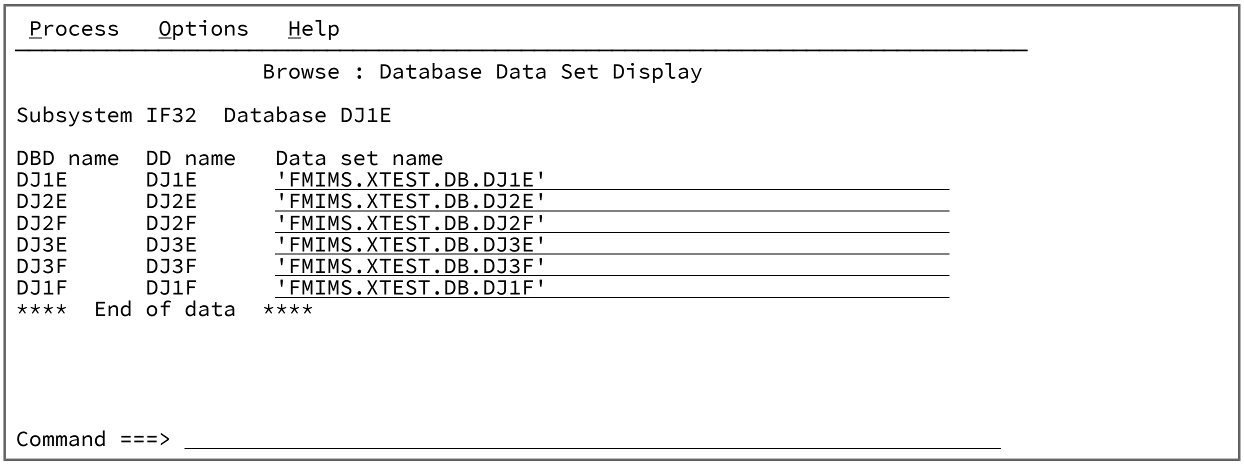
Step 11. Verify database data sets

Figure 65. Browse : Database Data Set Specification



```
 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 66. Browse : Database Data Set Display

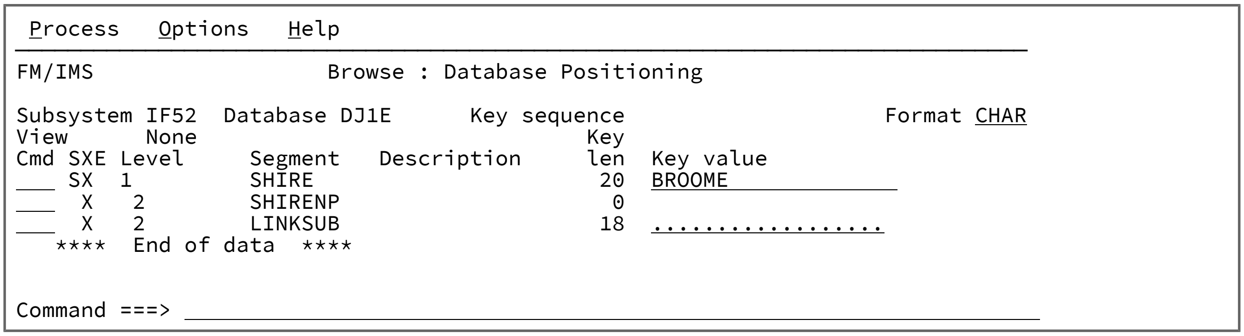


```
 Process   Options   Help
                    Browse : Database Data Set Display

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 ===>
```

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 65: Browse : Database Data Set Specification on page 321 and Figure 66: Browse : Database Data Set Display on page 321. Then press Enter. The Browse: Database Positioning panel is displayed.

Step 12. Database positioning

Figure 67. Browse : Database Positioning

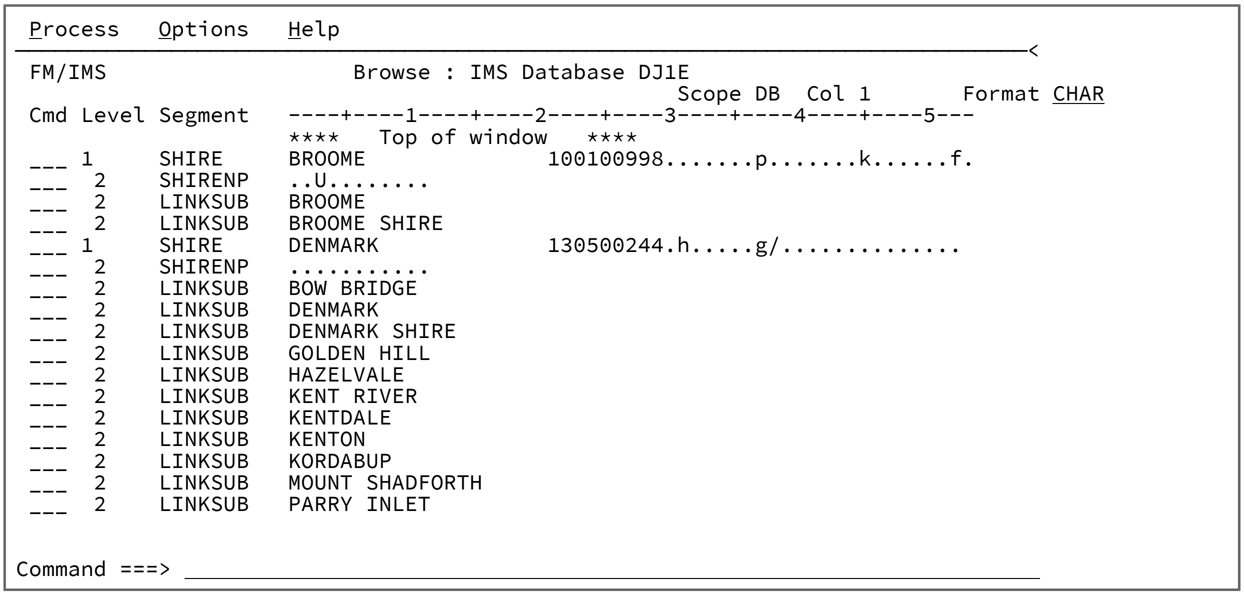


```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
FM/IMS                  Browse : Database Positioning

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 ===> ___________________________________________________________________
```

Enter an **S** in the **Cmd** field for the SHIRE segment and press Enter. The Browse : IMS™ Database panel is displayed.

Step 13. Browse the database

Figure 68. Browse : IMS™ 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 ===> ___________________________________________________________________
```

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.

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 IV. Customizing File Manager CICS® Component

Chapter 28. Preparing to customize FM/CICS

FM/CICS runs by means of a CICS® transaction, either by submitting a batch job or communicating with the ADFzCC 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. These topics 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.SFMNMOD2, whereas File Manager base function and the other File Manager component load modules are installed into FMN.SFMNMOD1.

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 ADFz Common Components, as described in Customizing ADFz Common Components for use with FM/CICS on page 374.

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 ADFzCC server method for the CICS® FMN transaction, you must also complete the customization of the ADFzCC server, as described in Customizing the ADFz Common Components server on page 558.

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 Customizing File Manager on page 12.

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 Customizing File Manager IMS Component on page 242.

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 Customizing File Manager Db2 Component on page 150.

Table 58: Summary of steps for customizing FM/CICS and the operating environment on page 324 lists the customization tasks you can perform for FM/CICS. Read the referenced sections in the table to determine the customization you need to do.

Checklist for installing and customizing FM/CICS

Table 58. Summary of steps for customizing FM/CICS and the operating environment

	Description
__ 1	Update your CICS® startup procedures - **required**. See Updating the CICS startup procedures on page 328.

Table 58. Summary of steps for customizing FM/CICS and the operating environment (continued)

	Description
__ 2	Provide OMVS segments for the CICS® region user ID and other users - **required**. See Providing OMVS segments for FM/CICS users on page 328.
__ 3	Customize TCP/IP, if necessary. See Customizing for CICS TCP/IP on page 329.
__ 4	Customize ADFzCC server, if necessary. See Customizing the ADFz Common Components server on page 558.
__ 5	Customize to run base FM, FM/IMS and FM/Db2 through FM/CICS, if necessary. See Accessing other File Manager functions from the primary option menu on page 330.
__ 6	Customize for FM/CICS on interconnected CICS® regions. See Providing CICS resource definitions for FM/CICS on interconnected regions on page 330.
__ 7	Customize the FM/CICS batch procedure. See Customizing the batch procedure on page 332.
__ 8	Modify and run the installation job, FMN3INST - **required**. See Modifying and submitting FMN3INST on page 333.
__ 9	Customize to process COBOL copybooks. See Customizing for processing COBOL copybooks on page 336.
__ 10	Customize to process PL/I include books. See Customizing for processing PL/I include books on page 337.
__ 11	Customize to process HLASM copybooks. See Customizing for processing HLASM copybooks on page 337.
__ 12	Customize to use DFSORT. See Customizing to use DFSORT to improve File Manager performance on page 337.
__ 13	Change the default options. See Changing the default options on page 337.
__ 14	Change the print and display tables. See Changing the print and display tables on page 341.
__ 15	Customize to use an I/O exit. See Customizing to use an I/O exit on page 341.
__ 16	Customize the FM/CICS security environment. See Customizing the FM/CICS security environment on page 342.
__ 17	Customize CICS® security for FM/CICS. See CICS security and FM/CICS on page 344.
__ 18	Determine how to customize the FM/CICS audit facility. See Alternatives for controlling File Manager CICS auditing on page 326.
__ 19	Customize FM/CICS for national languages. See Customizing FM/CICS for national languages on page 364.

Alternatives for controlling File Manager CICS® auditing

FM/CICS auditing is an optional facility. There is no requirement to implement it and FM/CICS works if auditing is not implemented. You should consider:

- Whether user access to data sets and other resources using File Manager CICS® requires auditing.
- The information that File Manager audit log records can provide.
- The information that File Manager audit log records cannot provide, and possible alternatives to obtaining that information.
- If you do decide to use File Manager auditing, how you will handle any issues associated with large audit log data sets, or additional SMF records.
- How you will use the information provided by File Manager audit log records.

If your site requires a record of a user's read access to data sets, an external security product such as RACF® can be configured to log access by some or all users, and may be a better alternative.

CICS® also provides logging facilities that may be a better alternative. File Manager audit of read access to data sets does not write audit log records for every record processed, rather the name of the data set and how many records were processed are written to the audit log.

File Manager audit of changes to data sets typically writes two log records, a before and after image of the record that was changed. If you intend to log update changes to data sets that are subject to heavy update activity you need to consider the performance impact of writing many audit log records, also the size of any audit log data sets that may be produced.

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 user IDs.
- 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 implement SAF-rule controlled auditing you need to decide how File Manager auditing will be enabled. This is described in more detail in Customizing the File Manager audit facility for CICS component on page 349. There are two alternatives. One requires an enabling SAF rule and the presence of a member in SYS1.PARMLIB. The other requires an enabling SAF rule but has no requirement for a member in SYS1.PARMLIB.

  The use of a member in SYS1.PARMLIB provides additional facilities compared with the alternative that does not require the use of SYS1.PARMLIB. The additional facilities are documented in File Manager options specified in PARMLIB members on page 532.

When you have determined the appropriate type of auditing for your installation, follow the instructions in Customizing the File Manager audit facility for CICS component on page 349.

Chapter 29. 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 sample job FMN3INST. See Modifying and submitting FMN3INST on page 333 for information about how to customize FMN3INST.

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 333 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 329 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 CICS security and FM/CICS on page 344.

When you have done all this, restart the CICS® region.

Providing OMVS segments for FM/CICS users

The CICS® region and each individual FM/CICS user must have a user ID defined to z/OS® with an OMVS segment. 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.

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. See "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 ADFzCC 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), 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.

See *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 Customizing the FM/CICS security environment on page 342.

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 332.

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 FMNLVL, in the CSD group, FMNCICS. You can do this using the sample job FMN3INSR. 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 on page 331 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 3 on page 331 on the CONN option of the FM/CICS options module, FMN3POPT. See Changing the default options on page 337 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 30. 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 ADFz Common Components server method for the CICS® FMN transaction, you must also perform the customization described in Customizing the ADFz Common Components server on page 558.

Customizing the batch procedure

Customizing the batch procedure is only required if you have specified START=BATCH in the FMN3POPT module or you plan to use the START=BATCH invocation option on the FMN transaction.

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 ADFzCC target libraries. ADFzCC is required for the operation of FM/CICS. Information about installing ADFzCC 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:

1. 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 333 for information about FMN3PRFD and FMN3INST.
2. For customization you can perform for ADFzCC, see Customizing ADFz Common Components for use with FM/CICS on page 374.

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 ADFzCC 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 this might be 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 user ID required by FM/CICS when the job is submitted.

   Note:

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 332 for information about FMN3CICB.)

PROFILE=*profile_dataset_name

Specifies the default data set name for the user's ADFz 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 user ID will be substituted for the variable &USER referenced in the name.

Note:

a. This profile data set 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 user ID to be provided when you run the transaction. If you specify *USERID=SIGNON then the user ID will default to the CICS® signed on user ID value when a user ID is not provided on transaction invocation. If you remove this statement and you invoke FM/CICS without a user ID then FM/CICS will prompt you for a user ID. If you specify *PASSWORD=REMEMBER or *PASSWORD=PASSTICKET you can remove this statement.

***PASSWORD=REMEMBER**

FM/CICS initially displays a logon panel where you provide user ID and password and other logon information. Specify this statement if you want to bypass the logon panel for subsequent invocations of FM/CICS. FM/CICS will remember your password, and if you sign on to CICS® with the user ID 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 enter:

```
FMuserid
```

The logon panel will also appear if the FM/CICS job does not respond to the CICS® region for any reason.

***PASSWORD=PASSTICKET**

Specify this parameter to direct FM/CICS to request a PassTicket for signed-on users. and, if the PassTicket is successful, bypass the logon panel. See *"Introduction to CICS security with RACF"* in the CICS® documentation for more information on generating and using PassTickets for secure sign-on.

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 `LEHLQ=` to the high level qualifier for your Language Environment® libraries. Remove the reference if Language Environment® is available from LINKLIST.
    - Change `CSD=` to the data set name of the DFHCSD file.
    - Change `FMPROF=` to the data set name to be used for the VSAM logon profile data set for FM/CICS.
    - Change `FMHLQ=` to the high level qualifier for the File Manager target libraries.
    - Change `FMPROFD=` 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 FMN.SFMNSAM1(FMN3CONN).
    - 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 333 for more information.
    - You can also change the CICS® resource definition names, listed in Table 59: FM/CICS resource definition names on page 335, 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 339 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 59. 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 user ID	Yes
FMNJ	Transient Data queue name for job submission.	Yes

Resource name	Description	FMN3POPI change required?
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.

Modifying and submitting FMN3PRDU

If you need to update the parameters for a CICS® region after FMN3INST has been run to create the profile data set, use the job FMN3PRDU.

There are three steps to modifying and running FMN3PRDU:

- Modify FMN3PRFD (see FMN3PRFD on page 333)
- Modify FMN3PRDU
- Run FMNPRDU

FMN3PRDU

You modify FMN3PRDU as follows:

1. Copy the member FMN3PRDU from FMN.SFMNSAM1 into your own JCL library.
2. Change the following values in your copy of FMN3PRDU:
    - Change the job card to meet your site's requirements.
    - Change `FMNHLQ=` to the high level qualifier for the File Manager target libraries.
    - Change `FMNPROF=profile.dsn` to the data set name to be used for the VSAM logon profile data set for FM/CICS.
    - Change `FMNPROFD=` to refer to your customized version of FMN3PRFD.

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. See the instructions in Customizing for processing COBOL copybooks on page 30, 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.

See the instructions in Customizing for processing PL/I include books on page 34, 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.

See the instructions in Customizing to use DFSORT to improve File Manager performance on page 35, 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). See File Manager options on page 383 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.

Note:

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. See Customizing the FM/IMS installation options module on page 263 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 user ID 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 ADFzCC server if START=TASK is specified (PORT option).
- The host name of ADFzCC 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 File Manager options on page 383.

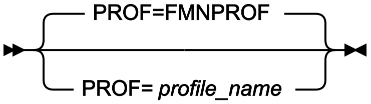
3. Modify FMN3POPI in the copy of FMN3POPT in your library, if required. See FMN3POPI on page 339 for information about FMN3POPI.
4. Modify the FMN3UMDP member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about changes you might need to make.
5. Install SMP/E 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.

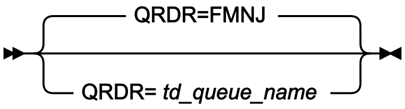
PROF



PROF

Specifies the 1-8 character file name used to define the profile data set to CICS®. The default is FMNPROF.

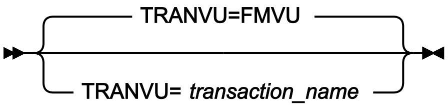
QRDR



QRDR

Specifies the 1-4 character transient data queue name that is used by FM/CICS for job submission.

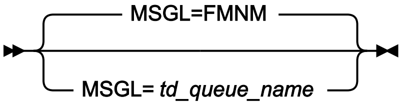
TRANVU



TRANVU

Specifies the 1-4 character transaction ID that is used to run the user verification program.

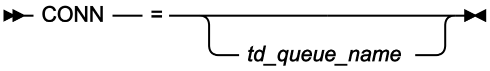
MSGL



MSGL

Specifies the 1-4 character transient data queue name that is used by FM/CICS as the message log.

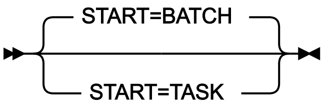
CONN



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

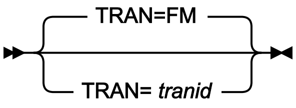
A batch job will be submitted to start the address space.

TASK

The address space will be started by connecting to a running ADFzCC server task.

Note: If you specify START=TASK then you must specify a value for PORT and a value for HOST.

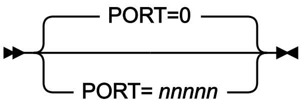
TRAN



TRAN

Specifies the 1-4 character transaction ID for FMN3CICS. The default is FM.

PORT

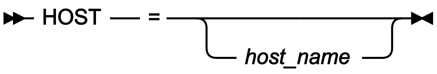


PORT

Specifies a 1 - 5 digit port number to be dedicated to the ADFzCC 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 ADFzCC 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 48.

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. See Customizing File Manager to use an I/O exit on page 120 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 428. For information about changing the options in FMN3POPT, see Changing the default options on page 337. This exit name can also be overridden by the user interactively. See the *File Manager for z/OS User's Guide and Reference* for information about how to use the I/O exit.

Chapter 31. 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 Customizing the File Manager security environment on page 51 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 user IDs 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 Controlling access on page 53 or Controlling access on page 62. For more information about FM/IMS security, see Customizing the FM/IMS security environment on page 280.

If you provide this access then the FM/CICS primary option menu will include the functions FMN (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 Customizing File Manager on page 12. 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 Customizing File Manager IMS Component on page 242. 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 Customizing File Manager Db2 Component on page 150.

Chapter 32. 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 user IDs 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.

Note:

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 FMN3INST on page 335). The default 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.

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 user ID 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® user ID 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 user ID must have sufficient authority to issue the commands listed above to successfully perform the FM/CICS function.

Note: A user ID 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 user ID 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 user ID 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 user ID must pass the CICS® Link Security check and optional User Security check on the remote region to successfully complete the FM/CICS function.

Controlling FM/CICS processing

File Manager has the ability to read, modify and change the status of CICS® resources. If the resources are not protected in the CICS® environment then there might be a requirement to control what functions File Manager for CICS® users can perform.

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 and XFACILIT class as follows.

Activating FM/CICS resource checking

The following facility class profile is used to determine whether File Manager checks access for any given CICS® resource.

```
FILEM.CICS.RESOURCE
```

Here is an example of activating FM/CICS resource checking.

```
RDEF FACILITY FILEM.CICS.RESOURCE AUDIT(NONE)      +
        UACC(READ) OWNER(TYRONED)
SETROPTS RACLIST(FACILITY) REFRESH
```

If this profile has been defined and the user has an access of read or more then FM/CICS perform resource security checking using the XFACILIT class profiles described below.

Defining access to CICS® resources

Define XFACILIT class profiles in the form:

```
FILEM.sysplex_name.cics_applid.resource_type.resource_name
```

Where

sysplex_name

The z/OS® sysplex name.

cics_applid

The CICS® VTAM® application id for the CICS® region

resource_type

One of these values:

FILE

CICS® files

TD

CICS® transient data queues

TS

CICS® temporary storage queues

ENQ

CICS® enqueue resource name

resource_name

The CICS® file name, transient data queue name or temporary storage queue name. This level doesn't apply to the resource type ENQ.

File Manager checks the level of access as follows to determine what functions can be performed.

READ

This allows read only functions like browse, print and view to run. The user is not allowed to modify a CICS® resource.

UPDATE

This allows update functions like edit, data create, copy to, and the ability to delete TS queues and empty TD queues from the resource list displays.

CONTROL

This allows CICS® SET function processing to change the status of a resource and the ability to purge tasks with outstanding enqueues for the XFACILIT class with resource_type ENQ. If the user does not have CONTROL access then the status fields that were modifiable on the resource list panels are protected for resources they are not allowed to modify.

Note: If the XFACILIT class for CICS® files has been defined and the user is performing a File Manager function that can read or update the data set, then an additional check is performed to validate whether the user has the required level of access to the data set name associated with the CICS® file.

Examples for RACF® definitions

Case 1. Ensure all files on CICSDEV can only be accessed read

```
RDEF XFACILIT FILEM.SYSPLEXA.CICSDEV.FILE.**  AUDIT(NONE) +
               UACC(READ) OWNER(userid)
```

Case 2. Ensure all CICS® resources on CICSDEV can only be accessed read

```
RDEF XFACILIT FILEM.SYSPLEXA.CICSDEV.**  AUDIT(NONE)   +
               UACC(READ) OWNER(userid)
```

Case 3. Allow update against all CICS® resources on CICSDEV and allow SET processing to the systems programmer userid

```
RDEF XFACILIT FILEM.SYSPLEXA.CICSDEV.**  AUDIT(NONE)  +
               UACC(UPDATE) OWNER(userid)

PE FILEM.SYSPLEXA.CICSDEV.**  +
         CLASS(XFACILIT) ID(sysprog) ACC(CONTROL)
```

Case 4. Allow a specific user full access to FILE names beginning with FM

```
RDEF XFACILIT FILEM.SYSPLEXA.CICSDEV.FILE.FM* AUDIT(NONE) +
               UACC(NONE) OWNER(TYRONED)
PE FILEM.SYSPLEXA.CICSDEV.FILE.FM* +
         CLASS(XFACILIT) ID(fmuser1) ACC(CONTROL)
```

Chapter 33. Customizing the File Manager audit facility for CICS® component

You can use optionally 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.

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.

File Manager provides two different methods for controlling whether audit records are written for File Manager CICS® component. These are described in detail in Alternatives for controlling File Manager CICS auditing on page 326.

You should determine which of the two methods is appropriate for your site's requirements.

Use the checklist to determine the customization required for the File Manager CICS® audit facility.

Table 60. Checklist for audit customization, FM/CICS component. This table lists choices and decisions.

Audit customization choice	Decision (Yes|No|Not applicable)
1. Control auditing using the FMN3POPT options module	
2. Control auditing using SAF rules and a member in SYS1.PARMLIB	
3. Control auditing using SAF rules, without any changes to SYS1.PARMLIB	
4. Audit records are to be written to a data set	
5. Audit records are to be written to SMF	

For choices 1 to 3, you should answer YES for one choice only. Mark the other two choices as 'Not applicable'.

If you answer YES for choice 1, you can answer YES for one of choices 4 and 5. Mark the other choice as 'Not applicable'.

If you answer YES for choices 2 or 3, you can answer YES for one or both of choices 4 and 5. If you answer YES for both choices you are implementing dual-logging, which is only available with SAF-controlled auditing.

Once you have completed the checklist, use the table below to identify the customization that is required for each customization choice. Only complete the customization if your decision was *"YES"* in the checklist.

Table 61. Customization steps for audit customization choices. This table lists choices and related actions.

Customization choice	Sections to complete
1. Control auditing using the FMN3POPT options module	• FMN3POPT-controlled auditing on page 350
2. Control auditing using SAF rules and a member in SYS1.PARMLIB	• SAF-controlled auditing for File Manager CICS component on page 352 • Implementing SAF-rule controlled auditing on page 355
3. Control auditing using SAF rules, without any changes to SYS1.PARMLIB	• SAF-controlled auditing for File Manager CICS component on page 352 • Implementing SAF-rule controlled auditing on page 355
4. Audit records are to be written to a data set	• Audit data set configuration on page 351
5. Audit records are to be written to SMF	• Customizing File Manager to write audit records to SMF on page 79

FMN3POPT-controlled auditing

To implement FMN3POPT-controlled auditing you need to set the AUDITLOG option in the FMN0POPI macro of the FMN3POPT module.

For information about the options see File Manager options on page 383.

- If you want to produce an audit trail, specify AUDITLOG=YES in FMN3POPI macro.
- If you want to produce an audit trail and report on the changes made at the conclusion of an edit function, specify AUDITLOG=DEMAND. The job submitted will be determined by the skeleton member FMN.SFMNSLIB(FMN3FTAD). Customize the job card and JCL to specify the reporting options you require.

If audit logging is enabled, audit log records are 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 under FM/CICS).

Note:

- 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 FMN3POPT.
- 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, FMN0POPT.

Audit data set configuration

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 386 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

Any 1-8 character constant that is valid in the context of a data set name.

userid

The user ID creating the data set.

Dyymmdd

The date of the activity.

Thhmmss

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 is null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix is also 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.

Set the AUDITHLQ parameter in the FMN0POPI macro for the FMN3POPT to the required value, based on the above information and your site's requirements.

You can print the information in a File Manager CICS® audit data set using the FM/CICS Print Audit Trail utility. To do this select option 3.6 from the FM/CICS Primary Option Menu.

Note:

- 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.

SAF-controlled auditing for File Manager CICS® component

There are two methods for implementing SAF-controlled auditing for File Manager CICS® component. These are:

1. Control auditing for File Manager CICS® component using an enabling SAF Facility class rule and a member in SYS1.PARMLIB.

   To use this method complete the customization described in SAF-controlled auditing using SYS1.PARMLIB on page 352.
2. Control auditing for File Manager CICS® component using an enabling SAF Facility class rule, without any changes to SYS1.PARMLIB.

   To use this method complete the customization described in SAF-controlled auditing without SYS1.PARMLIB on page 355.

Important: If you use a security product other than RACF®, review the information in When a security product other than RACF is in use on page 87 to avoid one possible cause of S047 abends.

SAF-controlled auditing using SYS1.PARMLIB

You need to define an enabling SAF facility profile as described below:

Define SAF facility profile

```
FILEM.PARMLIB.CICS
```

and ensure all FM/CICS users to be audited have at least read access to that facility. See the example below:

Example

User PROD1 to have SAF-rule controlled auditing using SYS1.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)
```

Add member FMN3PARM to SYS1.PARMLIB (or any other library in the logical parmlib concatenation). See Defining the FMN3PARM member on page 353.

Once the above SAF rule is defined and activated, auditing for FM/CICS component users is controlled by the FMAUDIT parameter in the FMN3PARM member. See FM/CICS options specified in FMN3PARM on page 551 for more information. If audit log records are to be written to SMF, the SMF record number is specified as an FMAUDIT parameter option. See FMAUDIT on page 551, and SMF_NO on page 552.

Note: FM/CICS does not start if a user has read access to the above facility and the FMN3PARM member does not exist in the logical parmlib concatenation.

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.

Defining the FMN3PARM member

If auditing is to be controlled from parmlib (user has read access to FILEM.PARMLIB.CICS, see SAF-controlled auditing for File Manager CICS component on page 352), then member FMN3PARM 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 a FMSECRTY statement. This option is not used at present, and can be either omitted, or commented out. It has no effect.

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.
   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
2. Add member FMN3PARM to the new library, specifying the appropriate FMAUDIT parameter.
3. 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 on page 354. If the FMN3PARM member specifies any include statements (see Facilities for customizing the FMN3PARM definitions on page 553), all of the included members must also reside in the same library.

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 FM/CICS options specified in FMN3PARM on page 551.

SAF-controlled auditing without SYS1.PARMLIB

You need to define an enabling SAF facility profile as described below:

Define SAF facility profile

```
FILEM.SAFAUDIT.CICS
```

and ensure that all FM/CICS users to be audited have at least read access to that facility. See the example below.

Example

User PROD2 to have SAF-rule controlled auditing without using SYS1.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)
```

If you use this method and intend to write audit records to SMF, the required SMF number is specified in the FMN3POPT module. See Customizing File Manager to write audit records to SMF on page 79.

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 355 for more information.
2. Write the relevant SAF rules. See the examples in SAF rule examples on page 359.
3. Activate SAF auditing for a specific logon using the chosen method of activating SAF-controlled auditing. See SAF-controlled auditing for File Manager Base component on page 84.
4. Using the selected logon, test the configuration to ensure that auditing is occurring as required.
5. When testing is complete, activate SAF-controlled auditing for all FM/CICS users.

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 65: FM/CICS auditing FACILITY class resource names on page 363 and Table 66: FM/CICS auditing XFACILIT class resource names on page 363).

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 65: FM/CICS auditing FACILITY class resource names on page 363). 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 (online 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:

1. Step 1.
   - If auditing is being controlled by means of parmlib, the FMNAUDIT specification of the FMN3PARM member is used as follows.

     The FMAUDIT specification setting in the FMN3PARM member (in SYS1.PARMLIB or any other library in the logical parmlib concatenation) 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 551 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 Customizing the File Manager audit facility for CICS component on page 349.

     SAF_CTRL=YES

     SAF-rule controlled auditing is in effect. Processing continues to Step 2.
   - If auditing is being controlled using the method which does not access the parmlib concatenation, the TSO logonid has READ access to the DAF FACILITY rule FILEM.SAFAUDIT.CICS for processing to continue to Step 2 on page 357.
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 65: FM/CICS auditing FACILITY class resource names on page 363; the various outcomes are summarized in Table 62: Determination of a user's ability to write audit log records on page 358.

   Table 62. 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

Note:

a. Refers to the level of access the user has to SAF FACILITY rule 1 in Table 65: FM/CICS auditing FACILITY class resource names on page 363.
b. Refers to the level of access the user has to SAF FACILITY rule 2 in Table 65: FM/CICS auditing FACILITY class resource names on page 363.
c. Refers to the level of access the user has to SAF FACILITY rule 3 in Table 65: FM/CICS auditing FACILITY class resource names on page 363.
d. 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.

3. Step 3.

   Does the user have access to write audit records for the current function and data set?

   The XFACILIT resource names 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 63: FM/CICS function codes that can be audited using SAF on page 359 shows the function codes which are supported.

Table 63. 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.

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 64. 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 65: FM/CICS auditing FACILITY class resource names on page 363 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 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 writes audit log records for functions which access resources. Table 63: FM/CICS function codes that can be audited using SAF on page 359 shows FM/CICS function 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.CICSAPLD.CTE.UPDATE.TS01[1]
RDEF XFACILIT FILEM.AUDIT3.CICSAPLD.CTE.UPDATE.TS01 OWNER(XXXXXXXX) UACC(READ)[2]
PE FILEM.AUDIT3.CTE.UPDATE.CICSAPLD.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 user IDs 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.CICSAPLD.CDE.ALL.TD01[1]
RDEF XFACILIT FILEM.AUDIT3.CICSAPLD.CDE.ALL.TD01 OWNER(XXXXXXXX) UACC(NONE)[2]
PE FILEM.AUDIT3.CICSAPLD.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 user IDs 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.CICSAPLD.CFP.FUNCTION.FMN.CICS.KSDS[1]
RDEF XFACILIT FILEM.AUDIT3.CICSAPLD.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 user IDs 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 FACILITY and XFACILIT class resource names and details.

Table 65. 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 66. FM/CICS auditing XFACILIT class resource names

Resource Name	Purpose
FILEM.AUDIT3.cicsapplid.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).
FILEM.AUDIT3.cicsapplid.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.cicsapplid.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).

Note:

1. Use this option with caution. The size of the data set being accessed and the editing technique used will influence the number of read records logged, and this may affect 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

You can customize FM/CICS for national languages other than English.

If you are using a language other than English, you will need to perform some or all of the customization tasks listed in Table 67: Summary of steps for customizing FM/CICS for a national language on page 364.

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 67. Summary of steps for customizing FM/CICS for a national language

Step	Description
__ 1	Set the LANGUAGE option in FMN3POPT. See Setting the LANGUAGE option on page 364.
__ 2	Verify that your CICS® terminal is defined as DBCS-capable, if necessary. See Verifying that the CICS terminal is DBCS-capable on page 365.
__ 3	Translate the FM/CICS logon messages to your language. See Translating the FM/CICS logon messages on page 365.
__ 4	Provide a version of FMN3MENU (FM/CICS logon messages) for your language. See Providing a multicultural version of FMN3MENU on page 366.
__ 5	Translate the FM/CICS panels to your language. See Translating the panel text on page 367.

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

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 98.

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.

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

Setting the LANGUAGE option

FM/CICS uses the LANGUAGE option set in the FM/CICS options module, FMN3POPT. This setting is used for all FM/CICS messages and panels.

Therefore, if you are using a language other than English, you will need to set the LANGUAGE option in FMN3POPT to one of the values shown in Table 68: Keyword values for the LANGUAGE option on page 365. For example, to use French messages, specify LANGUAGE=FRENCH.

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 41 and Changing the default options on page 337 for information on how to do this.

Table 68. 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

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 following CICS® transaction and verify that the *SOSI* data area contains X'FF':

```
CECI ASSIGN
```

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 multicultural version of FMN3MENU on page 366.

Providing a multicultural 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 333 for information about FMN3INST.

To use your translated messages, see Using the translated messages and panels on page 367.

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 your translated panels, see Using the translated messages and panels on page 367.

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 332 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 103.

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 104.

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:

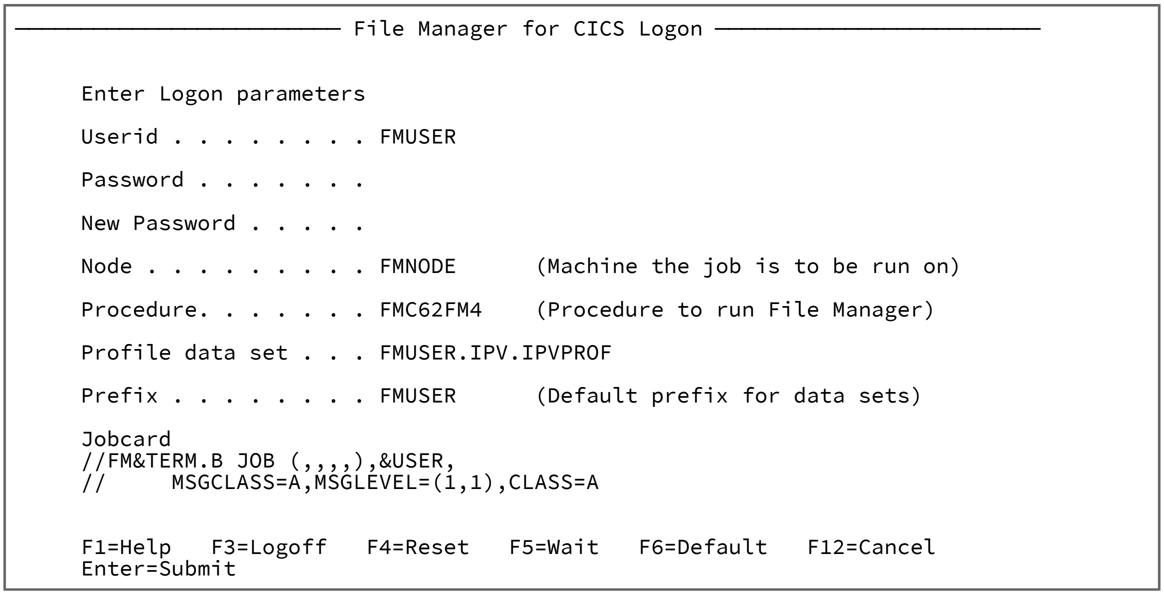
1. Copy the member FMN3MKOR from FMN.SFMNSAM1 to your own source library.
2. Change the message text in FMN3MKOR in your library.
3. Modify the FMN3UMDQ member in FMN.SFMNSAM1 to meet your site's requirements. Refer to the usermod for information about other changes you might need to make.
4. Install SMP/E usermod FMN3UMDQ.

Chapter 35. Verifying the customization of FM/CICS

After you have completed the initial installation and customization of FM/CICS, you can perform the following steps to verify your customization. You might need to refer to the File Manager for z/OS User's Guide and Reference for CICS.

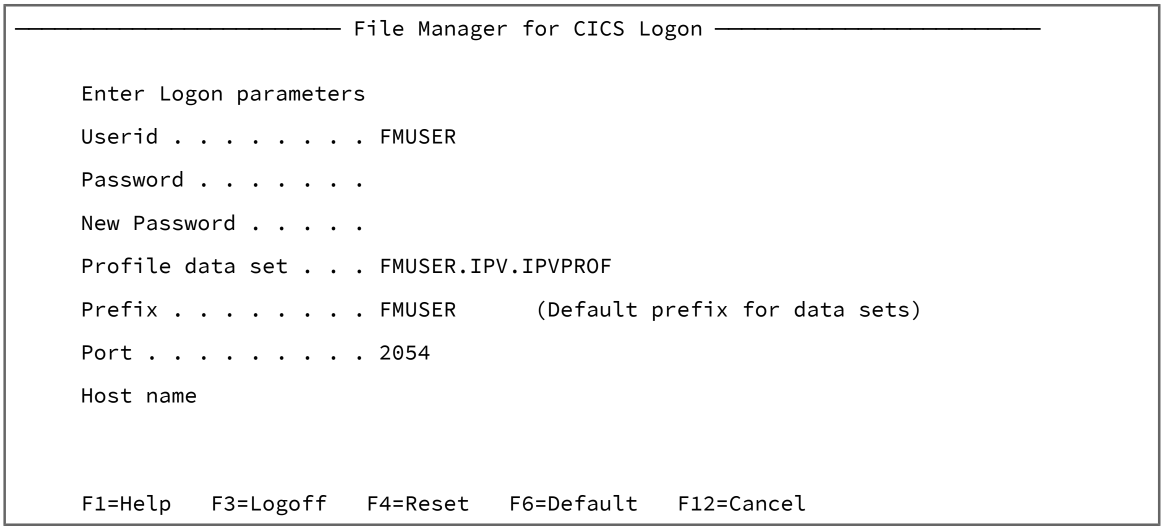
You verify the installation and customization of FM/CICS by logging on to a CICS® system where you plan to run FM/CICS. Before you can do this, you must have customized FM/CICS, your CICS® system, and, if necessary, TCP/IP.

1. Log on to the CICS® system where you have implemented FM/CICS, and enter the FM/CICS transaction, FM.
2. Enter your user ID 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.



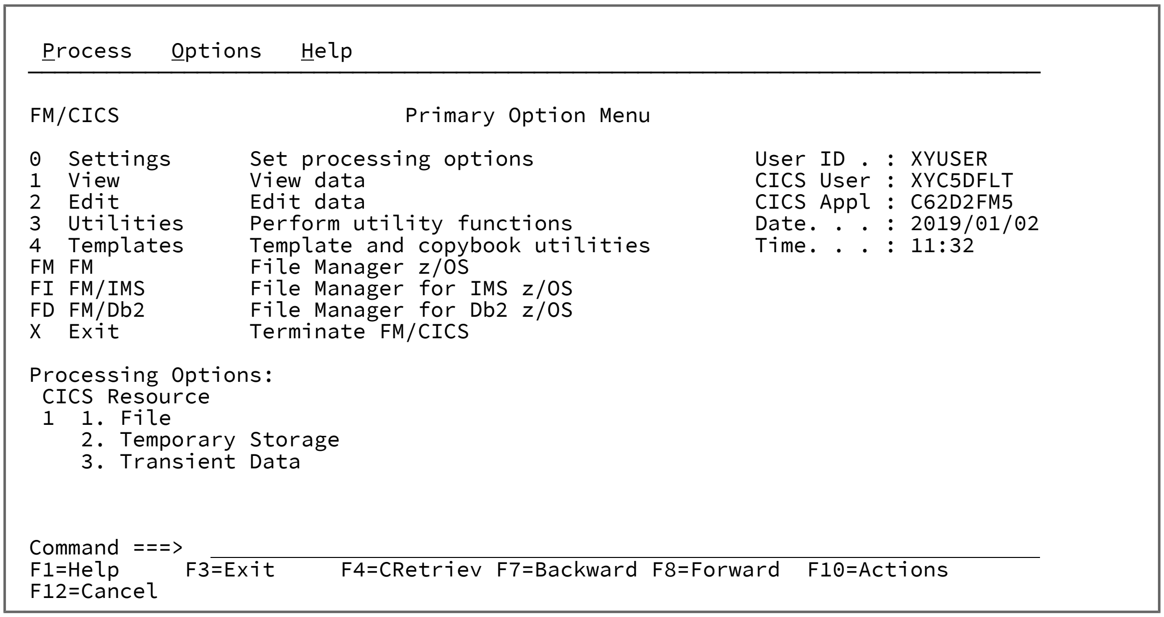
```
────────────────────────── File Manager for CICS Logon ──────────────────────────

     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 69: The FM/CICS Primary Option Menu on page 372.
    - 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 328 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.

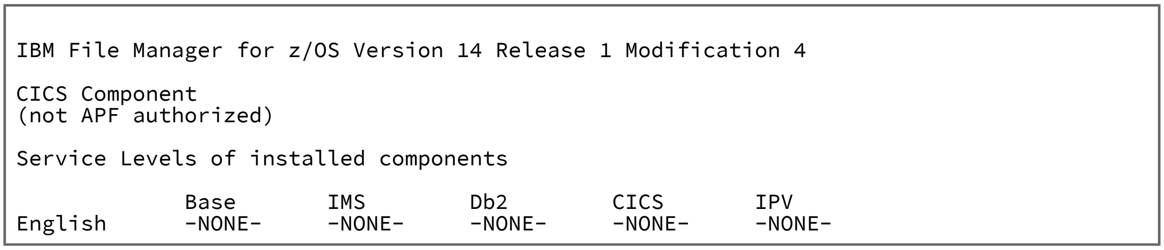
Figure 69. The FM/CICS Primary Option Menu



```
 Process   Options   Help
────────────────────────────────────────────────────────────────────────────────

FM/CICS                      Primary Option Menu

0  Settings      Set processing options                User ID . : XYUSER
1  View          View data                             CICS User : XYC5DFLT
2  Edit          Edit data                             CICS Appl : C62D2FM5
3  Utilities     Perform utility functions             Date. . . : 2019/01/02
4  Templates     Template and copybook utilities       Time. . . : 11:32
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
```

Note: The options FM, FI, and FD allow the user to invoke File Manager and FM/IMS from this primary option menu. 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 330.

6. Enter VER on the command line to display the release level and PTF level of FM/CICS and ADFz Common Components. A panel is displayed similar to this:



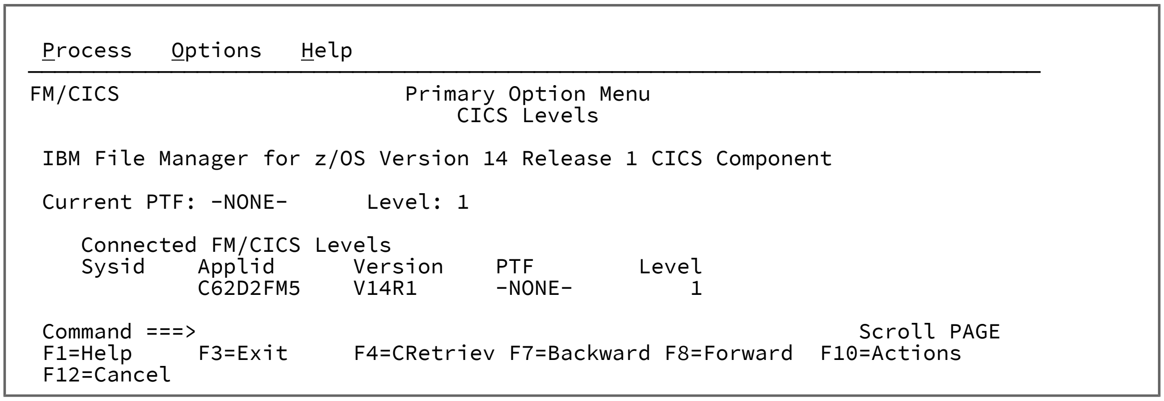
```
IBM File Manager for z/OS Version 14 Release 1 Modification 4

CICS Component
(not APF authorized)

Service Levels of installed components

             Base       IMS        Db2        CICS       IPV
English      -NONE-     -NONE-     -NONE-     -NONE-     -NONE-
```

Notes:

a. FM/CICS is always shown 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 14 Release 1 CICS Component

 Current PTF: -NONE-      Level: 1

    Connected FM/CICS Levels
    Sysid    Applid      Version    PTF        Level
             C62D2FM5    V14R1      -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.

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.

Chapter 36. Customizing ADFz Common Components for use with FM/CICS

ADFz Common Components (ADFzCC) is required for the operation of FM/CICS. There are some customization tasks you can perform after ADFzCC has been installed. Information about installing is included in the File Manager Program Directory.

The customization tasks you can perform are:

- Translate ADFzCC message text into your national language.
- Translate ADFzCC panel text into your national language.
- Change the Japanese message text.
- Change the Korean message text.

Library names after you finish installing ADFz Common Components

These topics assume you have installed ADFz Common Components into the default libraries. The default high level qualifier is IPV. Therefore after you have installed ADFzCC the names of the target and distribution libraries will be those listed in Table 69: ADFz Common Components target and distribution libraries on page 374.

If you have installed the ADFzCC Japanese component, the Japanese target and distribution libraries will be those listed in Table 70: ADFzCC Japanese target and distribution libraries on page 374. If you have installed the ADFzCC Korean component, the Korean target and distribution libraries will be those listed in Table 71: ADFzCC Korean target and distribution libraries on page 375.

Table 69. ADFz Common Components target and distribution libraries

Target library name	Distribution library name	Usage
IPV.SIPVMOD1	IPV.AIPVMOD1	ADFzCC load modules
IPV.SIPVSAM1	IPV.AIPVSAM1	Sample jobs
IPV.SIPVPENU	IPV.AIPVPENU	ADFzCC English panels
IPV.SIPVMENU	IPV.AIPVMENU	ADFzCC English messages
IPV.SIPVTENU	IPV.AIPVTENU	ADFzCC English tables

Table 70. ADFzCC Japanese target and distribution libraries

Target library name	Distribution library name	Usage
IPV.SIPVPJPN	IPV.AIPVPJPN	ADFzCC Japanese panels
IPV.SIPVMJPN	IPV.AIPVMJPN	ADFzCC Japanese messages

Table 70. ADFzCC Japanese target and distribution libraries (continued)

Target library name	Distribution library name	Usage
IPV.SIPVTJPN	IPV.AIPVTJPN	ADFzCC Japanese tables

Table 71. ADFzCC Korean target and distribution libraries

Target library name	Distribution library name	Usage
IPV.SIPVPKOR	IPV.AIPVPKOR	ADFzCC Korean panels
IPV.SIPVMKOR	IPV.AIPVMKOR	ADFzCC Korean messages
IPV.SIPVTKOR	IPV.AIPVTKOR	ADFzCC Korean tables

Translating the message text

All ADFzCC messages are provided in English.

Messages are also provided in Japanese if you have installed the Japanese component of ADFz Common Components, and in Korean if you have installed the Korean component of ADFz Common Components.

You can translate these messages into another language. (If no translated version of a particular message is available, ADFz Common Components uses the English version.)

All ADFz 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 your translated messages, add your messages library, IPV.SIPVM*yyy*, to the IPVPLIB concatenation in front of IPV.SIPVMENU, in FMN3CICB.

See Customizing the batch procedure on page 332 for information about FMN3CICB.

Translating the panel text

All ADFz Common Components 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 these panels into another language. (If no translated version of a particular panel is available, ADFz Common Components uses the English version.)

All ADFz 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 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 332 for information about FMN3CICB.

Changing the Japanese message text

If you have installed the Japanese component of ADFz Common Components, all ADFz 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 ADFz Common Components, all ADFz 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 such as IBM® Explorer for z/OS® or IBM® Developer for z/OS, 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, CICS® resources, UNIX™ System Services files, DB2® databases and tables, 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, CICS® resources, and USS files.
  - View load modules.
  - Manipulation of data stored in DB2® tables.
  - Edit and view of DB2® tables.
  - Use of wizards to Create, Copy, Export, Import, Print data, and so on.

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 a supported version of File Manager.

The ADFzCC server needs to be configured for the client to communicate with the host system. See Customizing the ADFz Common Components server on page 558.

Client requirements

File Manager plug-in for Eclipse is designed to work in an Eclipse environment such as IBM® Explorer for z/OS® Aqua or IBM® Developer for z/OS. The plug-in has been tested to work in these environments:

- IBM® Explorer for z/OS® Aqua
- IBM® Developer for z/OS® Version 14

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.

Further information about the installation requirements can be found at http://www-03.ibm.com/software/products/en/ibm-application-delivery-foundation-for-z-systems.

Host installation and configuration

For File Manager clients to communicate with your z/OS® system, you must first configure the ADFz Common Components server. For a full description of how to do this, see Customizing the ADFz Common Components server on page 558. To establish connection, File Manager plug-in users must know the IP address of your z/OS® system and the port number where the ADFzCC server is listening.

For users of the Remote Systems Explorer (RSE) view, the RSE server must be configured so that it knows the port number of the ADFzCC server running on the system. To do this, define a variable called `PD_SERVER_PORT` in your RSE server environment by adding this statement in the rse.env configuration file:

```
PD_SERVER_PORT=xxxx
```

where *xxxx* is an integer port number of the ADFzCC server.

For information about how to change your RSE environment, see *IBM® Developer for z/OS Host Configuration Guide*.

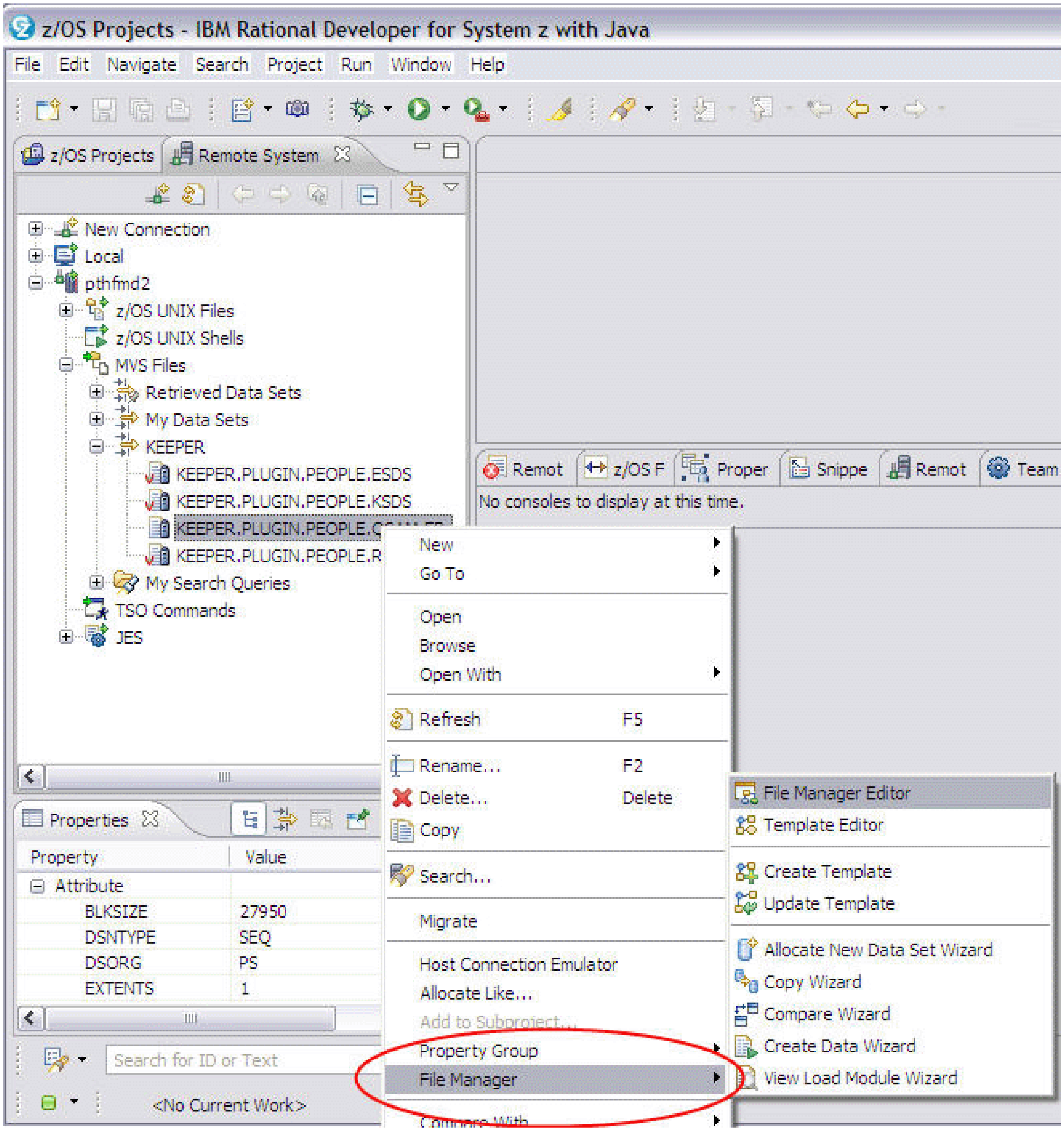
Client installation

The installation of IBM® File Manager plug-in for Eclipse depends on the target environment where you are installing the plug-in. Please visit https://developer.ibm.com/mainframe/products/downloads/ on page ______ for the latest installation instructions. Additional details about File Manager can be found at https://developer.ibm.com/mainframe/products/fm/ on page ______.

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 70: Screenshot of IBM File Manager plug-in for Eclipse on page 380).

For File Manager plug-in functions to work correctly, the host configuration steps must be completed first. See Host installation and configuration on page 379.

Figure 70. Screenshot of IBM® File Manager plug-in for Eclipse



No consoles to display at this time.

Property	Value
Attribute	
BLKSIZE	27950
DSNTYPE	SEQ
DSORG	PS
EXTENTS	1

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.

Chapter 38. Preparing for File Manager Remote Services

File Manager supports a number of services using resources accessed on a remote system via an ADFzCC server connection. When the remote ADFzCC server is configured for SSL/TLS, the local File Manager system validates the remote host's server certificate during the SSL/TLS handshake by verifying the Certificate Authority (CA) of the server's certificate is registered as trusted.

By default, File Manager searches local SITE certificates for the CA certificate of the remote system and verifies that it is trusted. Consequently, when importing a CA certificate for remote services, you should import it as a SITE certificate. For example, using RACF®:

```
RACDCERT ADD(‘hlq.ADFZCC.CA.EXPORT’) SITE TRUST WITHLABEL(‘your label’)
```

Alternatively, you can use the CERTRUST keyword of the FMN4POPT module to specify a trust store other than SITE. Note that all users of remote services need authority to access the nominated key store. See Customizing miscellaneous options in FMN4POPT on page 383 for more information about the CERTRUST keyword and the FMN4POPT module.

FMNAUTH DD usage

When using File Manager to create a remote connection through the menu option 11, the entered details are stored (in an internal format) in a file allocated to the FMNAUTH DD. If such an allocation does not pre-exist, as is normally the case, a data set is created as `Userid.FMNAUTH` and allocated to the FMNAUTH DD.

When running batch functions and specifying remote resources, the FMNAUTH DD needs to be included in JCL to provide the stored connection details.

Similarly, if there is a requirement to share remote connection details amongst users, you may pre-allocate the FMNAUTH DD in TSO/ISPF and File Manager reads the currently allocated FMNAUTH. Security access should be set appropriately for such scenario to allow READ access for trusted users to the data set referred to by FMNAUTH. Otherwise, for a user on a local system, who has connection details stored in their own FMNAUTH data set, we recommend setting the UACC for that resource to NONE if that is not already the default.

Appendix A. File Manager options

This section describes the File Manager options. The syntax described here applies to the FMN0POPI macro, which is installed by default in FMN.SFMNMAC1. 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, FMN1POPT, FMN2POPT, FMN3POPT, and FMN4POPT and apply the appropriate usermods. The options only take effect in the component where they are applied.

For information about changing File Manager miscellaneous options, see Changing the default options on page 384.

For information about changing File Manager base function options, see Changing the default options on page 41.

For information about changing FM/Db2 options, see Changing the default options on page 178.

For information about changing FM/IMS options, see Customizing the FM/IMS installation options module on page 263.

For information about changing FM/CICS options, see Changing the default options on page 337.

For File Manager plug-in for Eclipse users, the information about Base or Db2® function options applies.

Customizing miscellaneous options in FMN4POPT

Miscellaneous processing options are supplied with File Manager in the module FMN4POPT. This options module supports the following keywords:

REPOS

```
►►─ REPOS= template.repostry.dsn ─►◄
```

Where *template.repostry.dsn* is the name of the template repository file that is defined using File Manager ISPF option 7.7.1.

CERTRUST

```
►►─ CERTRUST= keystore-pattern ─►◄
```

Specifies the trust key store for File Manager remote services certificate validation. File Manager remote services connects to remote systems using the ADFzCC server. When the common server is configured for SSL/TLS, the local File Manager system validates the certificate of the remote host server by verifying that the Certificate Authority (CA) of the certificate is registered as trusted in the key store(s) specified by the *keystore-pattern*. Consequently, the *keystore-pattern* identifies one or more key stores that contain relevant CA certificates of all remote servers.

The *keystore-pattern* identifies the key store owner and the key store name, separated by a forward slash (/). For example:

```
CERTRUST=userid/keyring
```

The *keystore-pattern* can include asterisks (*) as wild cards. The maximum length you can specify for the value of the CERTRUST keyword is 246 bytes. CERTRUST is optional. If omitted, the default pattern used is `CERTRUST=*SITE*/*`.

The default macro statements are in FMN.SFMNSAM1(FMN0POPT).

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: If you do not want to use SMP/E you can use the sample job FMN0POPH to assemble FMN0POPT .

Changing the default options

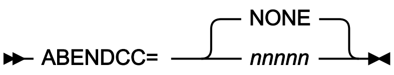
4POPT4

You can find the FMN4POPI macro statement in FMN.SFMNSAM1(FMN4POPT). Change the options as follows:

1. Copy the member FMN4POPT from FMN.SFMNSAM1 into your own source library.
2. Change the default options in your copy of the FMN4POPT member as required.
3. Modify the FMN4UMDP 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 FMN4UMDP.

Note: You can also use the sample job FMN4POPH to assemble FMN4POPT if you do not want to use SMP/E.

ABENDCC



ABENDCC

Specifies how File Manager processing resulting in a non-zero return code (including customized return codes) will be completed in batch.

NONE

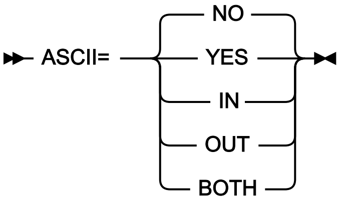
The job step will terminate with a non-zero return code. The default is NONE.

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.

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 the File Manager ASCII/EBCDIC translate table; output translation is done using the File Manager EBCDIC/ASCII translate table.

Note: Changing the ASCII option has no effect for FM/Db2, FM/IMS, or FM/CICS.

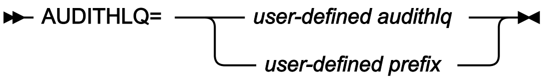
AUDDATAC

▶▶─ AUDDATAC= *xxxxxxxx* ─▶◀

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 72: The format of generated data set names on page 386.

Table 72. 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 user ID 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

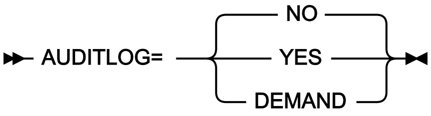
AUDITHLQ=FMNAUD.&&FUNCOD..&&USER

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= xxxxxxxx ─▶◀
```

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

```
                    ┌──── 16 ─────┐
▶▶─ AUDPQTY= ──────┴─ nnnnnnnn ──┴──▶◀
```

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

```
                    ┌───── 8 ─────┐
▶▶─ AUDSQTY= ──────┴─ nnnnnnnn ──┴──▶◀
```

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.

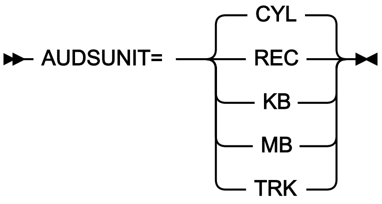
AUDSTORC

```
▶▶─ AUDSTORC= xxxxxxxx ─▶◀
```

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.

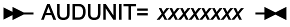
TRK

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= *xxxxxxxx* —▶◀

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= *dsn* —▶◀

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 your user ID. 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= *aux-hlq* —▶◀

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 your user ID. If you specify &&PREFIX. for any part of the qualifier, this is replaced with the user 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 user ID 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 user ID is ZZZZ then specifying

```
      FMN.&&USER..ABCD
```

results in a high-level-qualifier of

```
      FMN.ZZZZ.ABCD
```

Note:

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= *dsn* ─►◄

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.

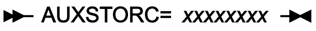
AUXMGMTC

►►─ AUXMGMTC= *xxxxxxxx* ─►◄

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.

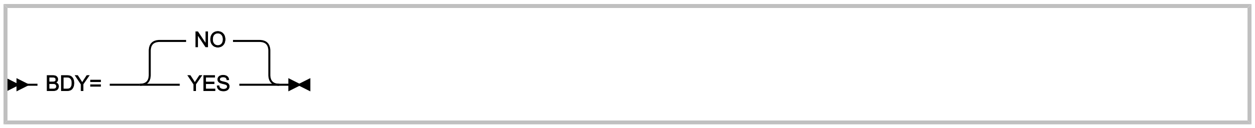
AUXSTORC



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.

BDY



BDY

When data is displayed or printed in SNGL format, this option specifies whether fields that start beyond the current record boundary should be suppressed.

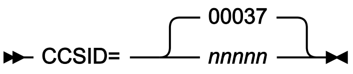
NO

Fields starting beyond the record's boundary are not suppressed.

YES

Fields starting beyond the record's boundary are suppressed.

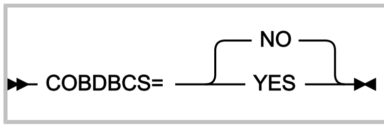
CCSID



CCSID

Specifies the CCSID to be used as the default CCSID when File Manager is run in batch. The CCSID specified should be a valid CCSID and a CCSID which is recognised by the Z/OS system the batch job will be run on. This option does not apply to FM/Db2.

COBDBCS



COBDBCS

Specifies if the **DBCS** COBOL compiler option is to be specified when compiling COBOL copybooks.

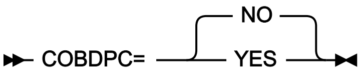
YES

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.

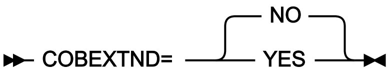
YES

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.

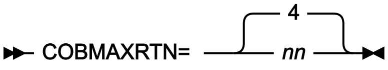
YES

The **Arith(extend)** COBOL compiler option is specified.

NO

The **Arith(extend)** COBOL compiler option is not specified.

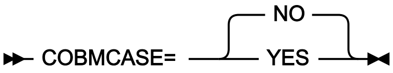
COBMAXRTN



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.

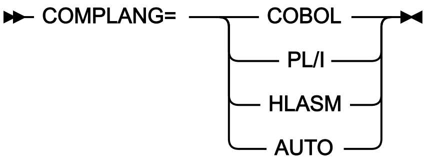
YES

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

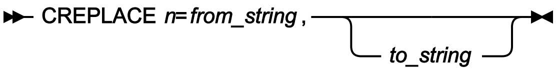
AUTO

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.

Note: The COMPLANG setting in FMN0POPT is the equivalent of the LANG parameter in batch functions.

- If COMPLANG is set it determines the installation default language for online and batch compilation.
- If COMPLANG is not specified then COBOL is the installation default for online compilation and AUTO is the default for batch compilation.
- If a value of COBOL, HLASM, PL/1, or AUTO is specified (in the **Compiler Language Selection** panel or through the LANG parameter in a batch job) it overrides the default language.

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.

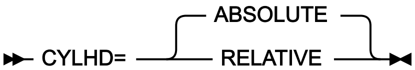
CSYSLIB*nn*

▶▶─ CSYSLIB *nn*=*dsname* ─▶◀

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

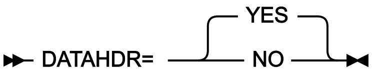
File Manager interprets cylinder-head and track-number values as actual physical addresses.

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.

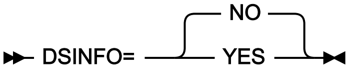
NO

Record header information is not included in the output.

Note:

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.

DSINFO



DSINFO

Specifies whether additional data set information for input and output sources should be produced in batch reports for the DSP, DSM, DSC, and FCH commands.

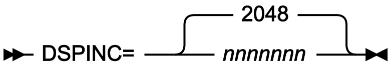
NO

No additional data set information is generated.

YES

Generates data set information including DSORG, RECFM, LRECL, BLKSIZE, and VSAM attributes including average record length, maximum record length, key offset, key length and reuse, for input and output data sets when applicable.

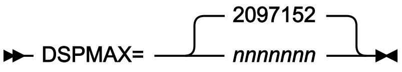
DSPINC



DSPINC

Specifies the amount (in KB) by which a clipboard will expand when it becomes full. Default is 2048 (2MB). Maximum is 4194303 (4GB).

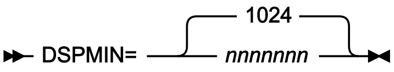
DSPMAX



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. See *z/OS® MVS™ Installation Exits*, SA22-7593, for more information about system exits.

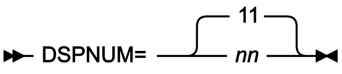
DSPMIN



DSPMIN

Specifies the starting size in KB of clipboards created by File Manager. Default is 1024 (1MB). Maximum is 4194303 (4GB).

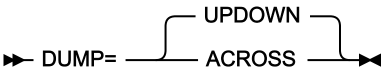
DSPNUM



DSPNUM

Specifies the maximum number of clipboards that File Manager 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).

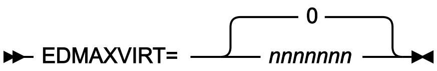
UPDOWN

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.

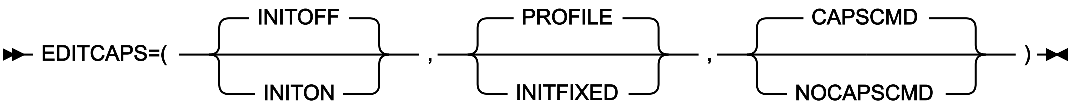
EDMAXVIRT



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 editor's virtual storage usage. Maximum is 4194303 (4GB).

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.

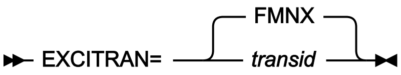
EOD

►►─ EOD= *delimiter* ─►◄

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').

EXCITRAN



EXCITRAN

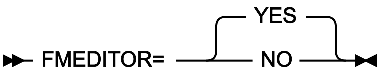
Specifies the transaction name used for the mirror transaction required for the external CICS® interface support.

transid

This value must be a valid CICS® transaction ID and must correspond to the transaction ID provided in the customized FMNCCONN job used to define the connection transaction to the CICS® regions.

For more information, see Setting up CICS access for File Manager base component and Eclipse plug-in on page 28.

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

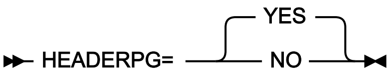
The File Manager editor is used.

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.

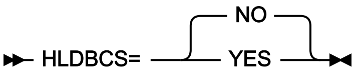
If you specify YES then the output will also show the File Manager installed components, version and PTF level. This is the same information as is produced by the $$FILEM VER control statement.

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.

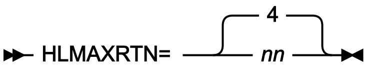
YES

The **DBCS** HLASM compiler option is specified

NO

The **DBCS** HLASM compiler option is not specified.

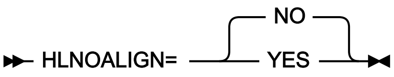
HLMAXRTN



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.

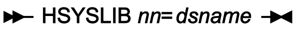
YES

The **NOALIGN** HLASM compiler option is specified.

NO

The **NOALIGN** HLASM compiler option is not specified.

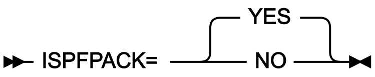
HSYSLIB*nn*



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.

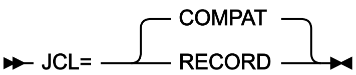
YES

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.

JCL



JCL

Defines the method by which JCL data is processed when running FCH or DSC with JCL=YES.

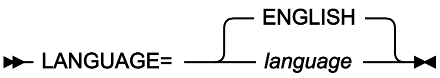
COMPAT

File Manager reformats the physical JCL records into a single logical JCL statement for processing and then builds a new set of physical JCL records.

RECORD

File Manager processes each physical JCL record as is, unless it contains a parameter field enclosed in apostrophes. If this parameter continues across multiple records, the records are flowed together to join the parameter before processing.

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

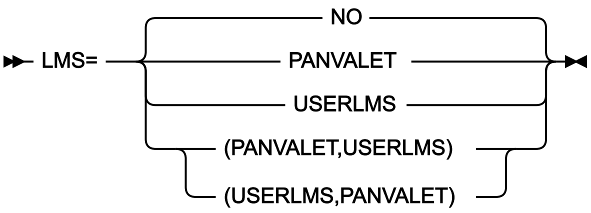
Messages will appear in the language of your choice.

Note:

1. See Table 16: Keyword values for the LANGUAGE option on page 102 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 461 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 Customizing File Manager for national languages on page 98, or Customizing FM/Db2 for national languages on page 210, or Customizing FM/IMS for national languages on page 308, or Customizing FM/CICS for national languages on page 364, 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= *xxxx* —▶◀

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

▶▶— MQREPHLQ= *mqrep-hlq* —▶◀

MQREPHLQ

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 user ID. If you specify &&PREFIX. for any part of the qualifier, this is replaced with your TSO prefix. Note that the current user ID 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 user IDs, 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 user ID is XXXX, specifying the value FMNTMQL. results in a dynamic name request for the pattern:

FMNTMQL.XXXX.*

Example 2

If the user ID 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.*

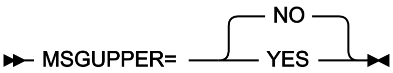
Example 3

If the user ID 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.

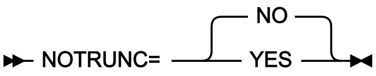
NO

No messages are translated to uppercase.

YES

All messages are translated to uppercase.

NOTRUNC



NOTRUNC

Specifies that, if the PAD option has been selected, no truncation is performed when copying or writing records to a variable-length data set.

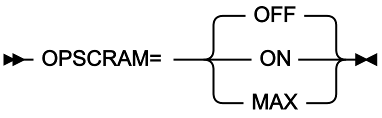
NO

Truncation performed.

YES

No truncation performed.

OPSCRAM



OPSCRAM

Specifies the optimization method used when scrambling data in fields using the data set copy function.

OFF

No optimization is performed.

ON

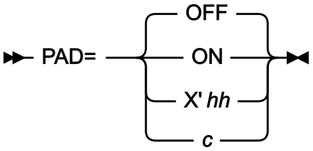
This option is designed to improve performance when scrambling large amounts of data where the Repeat or Random option has been specified for various fields.

MAX

This option is designed to provide the maximum performance improvement.

Note: If Repeat or Random has been specified for a field and a value list is not provided then duplicate alphanumeric characters within a field will yield the same scrambled character.

PAD



PAD

Specifies whether records are padded or truncated when being copied.

OFF

Records are not padded or truncated.

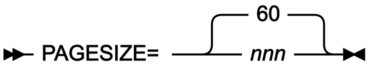
ON

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.

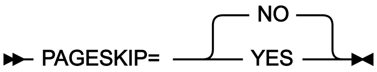
PAGESIZE



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.

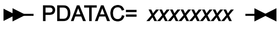
NO

Print output from each function does not begin on a new page.

YES

Print output from each function begins on a new page.

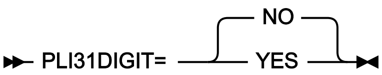
PDATAC



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.

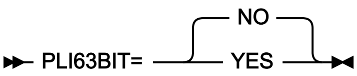
YES

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.

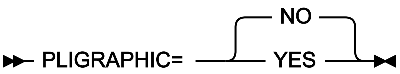
YES

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.

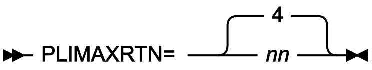
YES

The **GRAPHIC** PL/I compiler option is specified.

NO

The **GRAPHIC** PL/I compiler option is not specified.

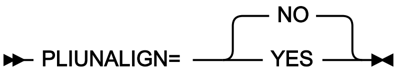
PLIMAXRTN



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= xxxxxxxx ─▶◀
```

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

```
                   ┌─ userid.FMN.LIST ─┐
▶▶─ PRINTDSN= ─────┴─────── dsn ───────┴─▶◀
```

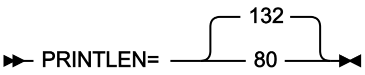
PRINTDSN

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 your user ID. If you specify &&PREFIX. as the high-level qualifier for PRINTDSN, the data set name will have a high-level qualifier of your 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 for z/OS User's Guide and Reference*.

PRINTLEN



PRINTLEN

Specifies the line length of printed output.

132

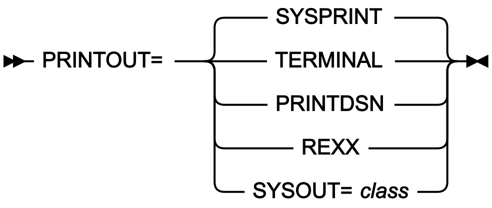
The line length of printed output is 132 characters, suitable for a line printer.

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 431.

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.

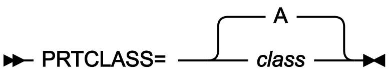
REXX

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.)

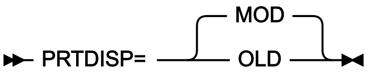
PRTCLASS



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.

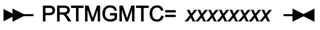
MOD

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.

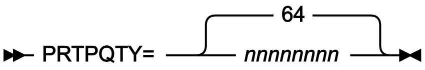
PRTMGMTC



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.

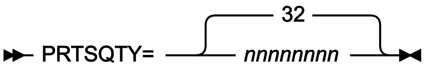
PRTPQTY



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.

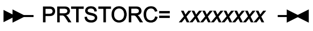
PRTSQTY



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.

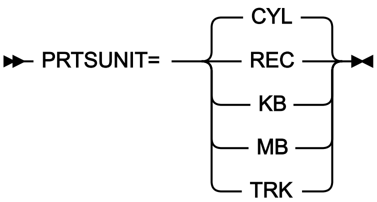
PRTSTORC



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.

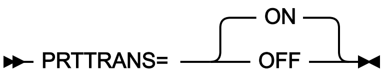
TRK

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

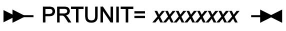
Non-printable characters are translated to blanks using a translate table. This may make printing faster.

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 48 and Changing the print and display translation tables for languages other than English on page 98.

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= xxxxxxxx ─▶◀
```

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=dsname ─▶◀
```

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= xxxxxxxx ─▶◀
```

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 PTRDATAC is eight (8) bytes. PTRDATAC is optional. If omitted the SMS data class specified in PDATAC is used, if supplied.

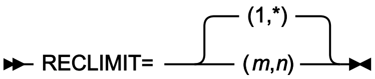
PUNIT

```
              ┌─ SYSALLDA ─┐
▶▶─ PUNIT= ───┴─ xxxxxxxx ─┴─▶◀
```

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,*)

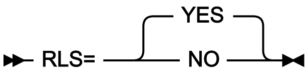
The entire record (or block) is printed.

(*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.

RLS



RLS

Specifies whether VSAM RLS support is active or not.

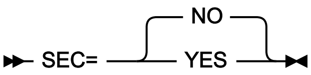
NO

VSAM data sets will be opened in non-RLS mode.

YES

VSAM data sets will be opened in RLS mode if applicable.

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.

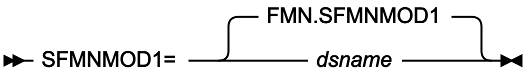
YES

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 69.

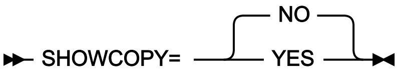
SFMNMOD1



SFMNMOD1

The data set name of the File Manager load library which is used in the FMNFTEXC and FMN1FTEX job control skeletons.

SHOWCOPY



SHOWCOPY

Specifies whether the copybook name for a record layout is made visible during template edit and for an edit, view or browse session where a copybook or template is being used.

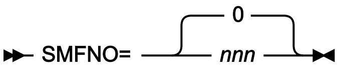
YES

The copybook name if applicable is visible.

NO

The copybook name is not visible. The default is NO.

SMFNO



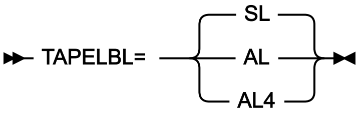
SMFNO

Specifies the SMF record type for audit logging. You can specify a value from 128 to 255, or 0. The default is 0.

Note:

1. Specifying SMFNO=0 indicates that audit logging will not create SMF records.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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 for z/OS User’s Guide and Reference*.

Note: Changing the TAPELBL option has no effect for FM/Db2, FM/IMS or FM/CICS.

TDATAC

```
▶▶— TDATAC= xxxxxxxx —▶◀
```

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= temp_hlq —▶◀
```

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 user ID. 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 user ID 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 user ID is ZZZZ then specifying

```
FMN.&&USER..ABCD
```

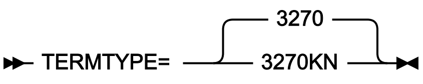
will result in a high-level-qualifier of

```
FMN.ZZZZ.ABCD
```

Note:

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

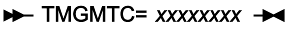
Specify 3270 for standard 3270 terminals.

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.

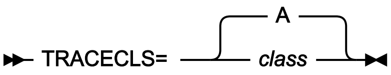
TMGMTC



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.

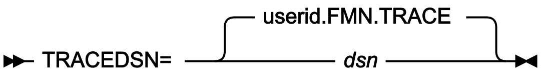
TRACECLS



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.

TRACEDSN

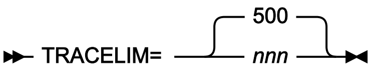


TRACEDSN

The trace data set where output is directed when the TRACEOUT print option is TRACEDSN.

You can specify any valid name for this data set. If you specify &&USER as a high-level qualifier for TRACEDSN, the data set name will have a high-level qualifier of the user's user ID. If you specify &&PREFIX as the high-level qualifier for TRACEDSN, the data set name will have a high-level qualifier of the user's TSO PREFIX.

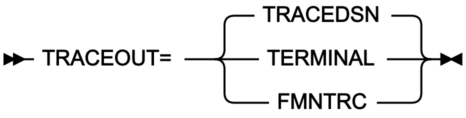
TRACELIM



TRACELIM

Specifies the number of trace entries to be retained by the background trace mechanism. In the event of abnormal termination of a File Manager function, or the DEBUG option is selected, these entries will be written to TRACEDSN. If this value is set to zero, the background trace mechanism is switched off. If a value is specified it must be zero, or in the range 500 to 9999.

TRACEOUT



TRACEOUT

Specifies where printed File Manager debug trace output should be sent.

FMNTRC

Debug trace output is sent to the current FMNTRC allocation.

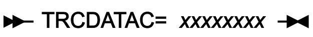
TERMINAL

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).

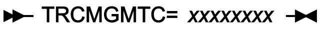
TRCDATAC



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.

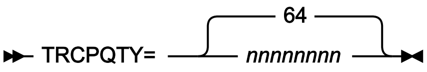
TRCMGMTC



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.

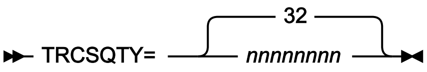
TRCPQTY



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.

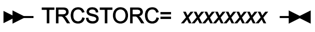
TRCSQTY



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.

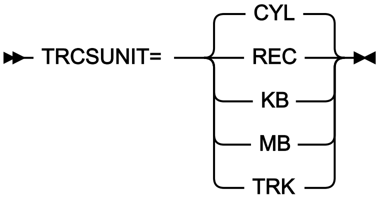
TRCSTORC



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= *xxxxxxxx* ►◄

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.

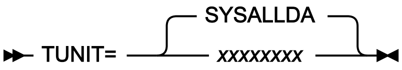
TSTORC

►► TSTORC= *xxxxxxxx* ►◄

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.

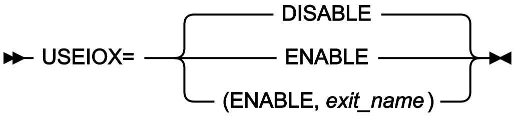
TUNIT



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 Customizing File Manager to use an I/O exit on page 120.

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.

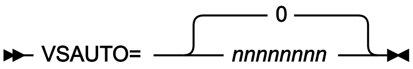
ENABLE

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.

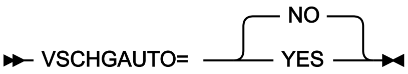
VSAUTO



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.

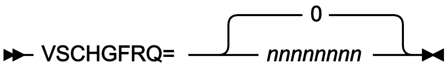
YES

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.

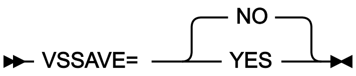
VSCHGFRQ



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.

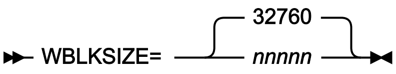
NO

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.

WBLKSIZE



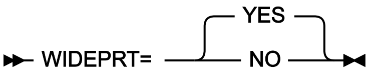
WBLKSIZE

Specifies the blocksize of the to-be-allocated (new) print output data sets (online) or for SYSPRINT allocation (in batch) if the WIDEPRT=YES option is specified.

nnnnn

For the to-be-allocated (new) print output data or the SYSPRINT allocation (in batch), File Manager uses the blocksize specified with a record format of VBA. The record size used is taken from the WLRECL option.

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.

The maximum record length/block size can be overwritten by the WBLKSIZE and WLRECL File Manager options.

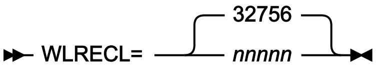
NO

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, 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, the data set's attributes cannot 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.

WLRECL



WLRECL

Specifies the record size of the to-be-allocated (new) print output data sets (online) or for SYSPRINT allocation (in batch) if the WIDEPRT=YES option is specified.

nnnnn

For the to-be-allocated (new) print output data or the SYSPRINT allocation (in batch), File Manager uses the record size specified with a record format of VBA. The blocksize used is taken from the WBLKSIZE option.

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 178 for information on how to do this.

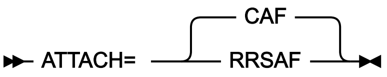
FMN2SSDM

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 Defining all Db2 systems that FM/Db2 will access in FMN2POPT (required) on page 172.

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.

RRSAF

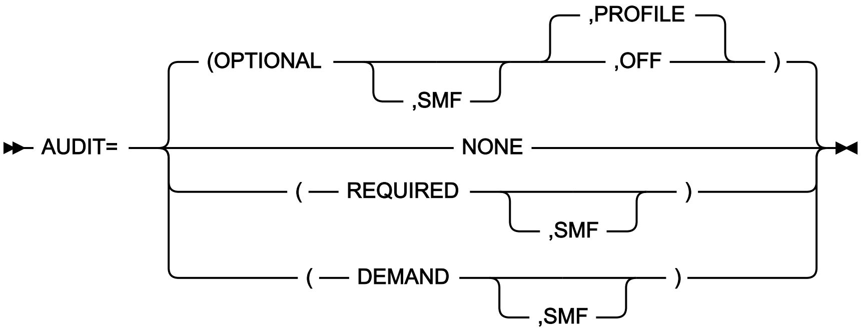
The Db2® Resource Recovery Services attachment facility (RRSAF) will be used to connect to and use the Db2® subsystem defined by SSID.

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

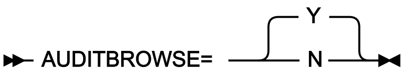
An audit trail is produced, regardless of the setting of `Create an audit trail` in the editor options. An audit report job will be submitted at the conclusion of an edit function. If **AUDIT=(DEMAND,SMF)** is specified, then the audit trail is also written to SMF.

Note:

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 on page 435 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

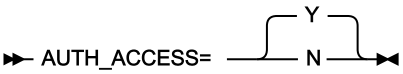
Specifies that audit log records are to be written for FM/Db2 view or browse sessions. The default is AUDITBROWSE=Y.

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.

AUTH_ACCESS



AUDITBROWSE

Specifies whether or not access to the *AUTH Db2® catalog tables is available to FM/Db2 users.

Y

Specifies that SELECT access to the *AUTH Db2® catalog tables is available.

This is the default. Specify Y if FM/Db2 users are to have access to the FM/Db2 Privileges utility, and are able to view, and possibly change, information the *AUTH Db2® catalog tables.

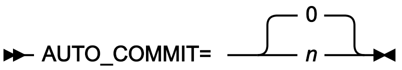
N

Specifies that no SELECT access to the *AUTH Db2® catalog tables is available.

Specify N in environments where the information in the *AUTH is considered confidential. Specify AUTH_ACCESS=N when you do not grant SELECT access on the *AUTH Db2® catalog tables to FM/Db2 users.

The FM/Db2 Privileges utility is disabled when N is specified, along with line commands (G and P) that may be issued against a list of Db2® objects in the FM/Db2 Object List utility.

AUTO_COMMIT



AUTO_COMMIT

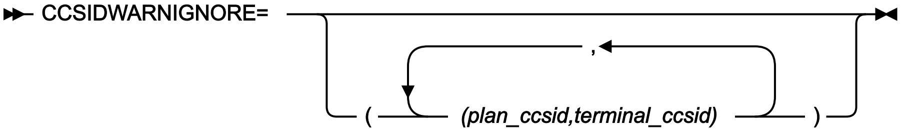
Specifies a value for the editor **Auto commit (Changes)** option that is to apply to all users accessing the Db2® subsystem specified in the SSID parameter. Allowable values are in the range 0 - 10000.

The default is 0, meaning that every FM/Db2 editor user can change the value for the **Auto commit (Changes)** option using the **Editor Options (6 of 8)** panel. Allowable values are in the range 0 - 2147483647.

Specify a positive value (*n*) to:

- Set the **Auto commit (Changes)** value for every FM/Db2 editor user to the specified value. The value appears on the **Editor Options (6 of 8)** panel as a protected field.
- Remove the ability of each FM/Db2 editor user to change the value for the **Auto commit (Changes)** option.

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.

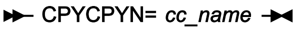
An option is provided to allow you to stop the displaying of the CCSID Warning message when there is a mismatch between the terminal CCSID and the SSID plan CCSID.

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.

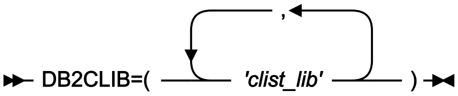
CPYCPYN



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

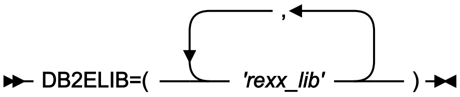


DB2CLIB

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

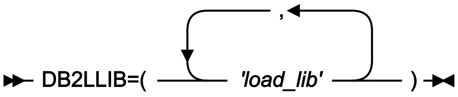


DB2ELIB

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.

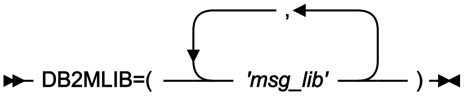
  or:
- Db2® load libraries have been added to the LINKLIST, but the libraries in the LINKLIST are for a different version or release of Db2® to the Db2® system defined by this FMN2SSDM macro.

Note:

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 152 for information about the LINKLIST and TSO logon procedures for FM/Db2.

DB2MLIB

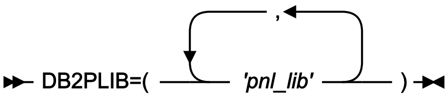


DB2MLIB

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

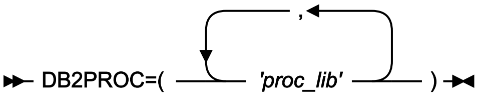


DB2PLIB

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

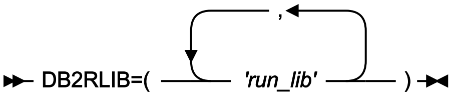


DB2PROC

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:

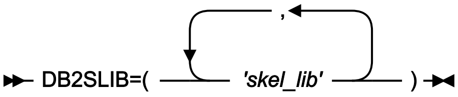
- The Db2® run libraries have not been added to the LINKLIST, and the libraries have not been allocated as part of the TSO logon procedure, or any initialization clist or exec run prior to starting FM/Db2.

  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 152 for information about the LINKLIST and TSO logon procedures for FM/Db2.

DB2SLIB

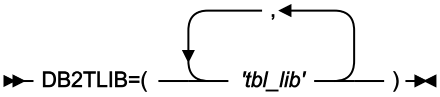


DB2SLIB

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.

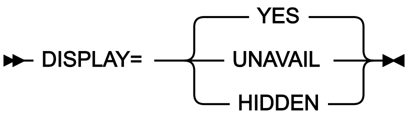
DESC

▶▶─ DESC= *'description'* ─▶◀

DESC

Is the description of the subsystem that will appear in the **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 **Db2 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.

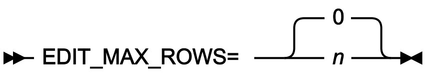
Unavailable SSIDs will not appear on the selection list if **LIST=ACTIVE** is specified in the FMN2POPI macro. However, FM/Db2 will attempt to connect to subsystems marked as UNAVAIL if **CONNECT=ANY** is specified in the FMN2POPI macro.

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.

EDIT_MAX_ROWS



EDIT_MAX_ROWS

Controls the maximum number of rows that can be loaded into any FM/Db2 editor session, and can be used to disable "large" table support.
The default is 0, meaning that:

- "Large" editor mode is available to all FM/Db2 users. If an FM/Db2 user specifies row count = 0 for an editor-related function, FM/Db2 may use a scrollable cursor to access Db2® data. This might require Db2® to build a copy of the result table in a Db2® temporary database.
- "Normal" editor mode is available to all FM/Db2 users. An FM/Db2 user can specify row count = *n*, *n*>0 for an editor-related function, FM/Db2 attempts to load up to *n* rows into the FM/Db2 editor session. There is no restriction on the value *n* that can be specified up to a maximum of 2G-1.

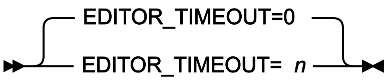
Specify a positive value (*nn*) to:

- Disable "large" editor mode. If an FM/Db2 user specifies row count = 0 for an editor-related function, FM/Db2 ignores the 0 value, the resulting FM/Db2 editor session is a "normal" edit session and FM/Db2 loads at most *nn* rows.
- Set an absolute limit on the number of rows that any FM/Db2 can process in an FM/Db2 editor session. If an FM/Db2 user specifies row count = *m*, *m*>*nn* on a function entry panel, FM/Db2 ignores the *m* value and loads at most *nn* rows into the editor.

Tip:

- Specify a large value (for example, 2,000,000,000) to disable "large" editor mode, while setting no reasonable upper limit to the number of rows that can be loaded into an FM/Db2 editor session.
- Specify a modest value (for example, 10,000) to limit TSO memory usage when using the FM/Db2 editor.

EDITOR_TIMEOUT



EDITOR_TIMEOUT

Specifies whether FM/Db2 editor sessions are subject to an inactivity time-out, and the length of time that must elapse before the inactivity time-out is triggered.

Specify 0 to disable the inactivity time-out. This is the default.

Specify a non-zero value to enable the inactivity timer. **n** is integer, **n** > 0. The value specified is interpreted as HHMMSS. Here are some examples:

```
n=30        specifies an inactivity time-out interval of 30 seconds
n=100       specifies an inactivity time-out interval of 1 minute
n=10000     specifies an inactivity time-out interval of 1 hour
n=718191    specifies an inactivity time-out interval
                 of 71 hours, 81 minutes and 91 seconds
```

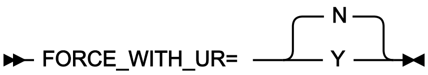
The inactivity time-out applies to the locally connected Db2® system. For example, suppose the FMN2POPT has these entries:

```
FMN2SSDM    SSID=DSNA,
            EDITOR_TIMEOUT=0
FMN2SSDM    SSID=DSNB,
            EDITOR_TIMEOUT=45
```

An FM/Db2 user connected locally to DSNA is not subject to the editor inactivity time-out. If the FM/Db2 user specifies a location value to access a Db2® object at DSNB, the resulting editor session IS NOT subject to the inactivity time-out.

An FM/Db2 used connected locally to DSNB is subject to the editor inactivity time-out. If the FM/Db2 user specifies a location value to access a Db2® object at DSNA, the resulting editor session **is** subject to the inactivity time-out.

FORCE_WITH_UR



FORCE_WITH_UR

Specifies whether FM/Db2 should always append *"WITH UR"* to any SELECT statement processed by the FM/Db2 editor. The default is N; FM/Db2 does not add the *"WITH UR"* clause to SELECT statements processed by the FM/Db2 editor.

Specify Y when there is a requirement to reduce the number of Db2® locks taken as a result of the use of the FM/Db2 editor.

Y

When Y is specified there are two effects:

- The Concurrency and Keep Locks editor options, which appear on the seventh editor options panel, are not displayed, and all FM/Db2 users are unable to specify either option.
- The *"WITH UR"* clause is automatically added to any SELECT statement processed by the File Manager Db2® editor. This includes the Browse (B), View (1), Edit (2), Basic Prototyper (4.1), Advanced Prototyper (4.2), Enter, execute and explain SQL statements (4.3) and Edit and execute SQL statements from a data set (4.4).

N

When N is specified:

- The Concurrency and Keep Locks editor options, which appear on the seventh editor options panel, are displayed, and all FM/Db2 users are able to change either option to their preference. The values selected by each user are applied to FM/Db2 editor sessions arising from the Browse (B), View (1) and Edit (2) functions.
- Select statements entered using the following functions are not affected:
  - Basic Prototyper (4.1)
  - Advanced Prototyper (4.2)
  - Enter, execute and explain SQL statements (4.3)
  - Edit and execute SQL statements from a data set (4.4)

  FM/Db2 processes the SELECT statement from these functions without any additions.

LDFDDLN

►►— LDFDDLN= *ld_name* —►◄

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 450, SLDJCL2 on page 450, SLDJCL3 on page 450, and SLDJCL4 on page 451.

For an example, see Examples of FMN2SSDM macros on page 173.

LOCATION

►►— LOCATION= *location_name* —►◄

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 online functions only.

LOCATION_NICKNAME

►►─ LOCATION_NICKNAME= *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 online functions only.

LODINDN

►►─ LODINDN= *li_name* ─►◄

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= *ev_line1* ─►◄

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 173.

OPTEVT2

►►─ OPTEVT2= *ev_line2* ─►◄

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= *ev_line3* ─►◄

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.

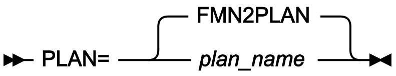
OPTEVT4

►►─ OPTEVT4= *ev_line4* ─►◄

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

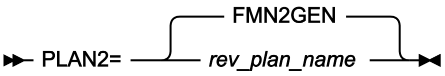


PLAN

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 jobs FMN2BN9P and FMN2BN0P. See Binding Db2 (required) on page 170.

PLAN2

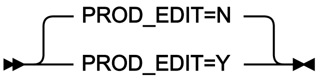


PLAN2

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 jobs, FMN2BN9P and FMN2BN0P. See Binding Db2 (required) on page 170 for information about FMN2BN9P and FMN2BN0P.

PROD_EDIT



PROD_EDIT

Specifies whether various restrictions should apply to any FM/Db2 editor sessions. These restrictions are intended to reduce the potential for locking and resource contention when accessing Db2® data. Collectively these restrictions are referred to as the *"PROD_EDIT"* (Production Edit) behavior.

Specify PROD_EDIT=N to disable the PROD_EDIT restrictions. This is the default.

Specify PROD_EDIT=Y to enable the PROD_EDIT restrictions. Specifying PROD_EDIT=Y has multiple effects, as described below:

- An inactivity time-out applies to every FM/Db2 editor session. If EDITOR_TIMEOUT=0 is also specified in the same FMN2SSDM macro invocation it is replaced with EDITOR_TIMEOUT=60. You can specify any value for EDIT_TIMEOUT except 0, and that value will not be over-ridden.
- *"Large"* editor mode is disabled for all users. If EDIT_MAX_ROWS=0 is also specified in the same FMN2SSDM macro invocation it is replaced with EDIT_MAX_ROWS=1000. You can specify any value for EDIT_MAX_ROWS except 0, and that value will not be over-ridden.
- Table locking is disabled for all users. If TABLE_LOCKING=YES is also specified in the same FMN2SSDM macro invocation it is replaced with TABLE_LOCKING=NO.
- The editor option *"Commit after data fetch"* is fixed and is selected for all FM/Db2 users.
- The editor option *"Commit when save issued"* is fixed and is selected for all FM/Db2 users.
- The editor option *"Commit when no save errors"* option is fixed and is not selected for all FM/Db2 users.

The PROD_EDIT behavior applies to the locally connected Db2® system.

For example, Suppose the FMN2POPT has these entries:

```
FMN2SSDM    SSID=DSNA,
            PROD_EDIT=N
FMN2SSDM    SSID=DSNB,
            PROD_EDIT=Y
```

An FM/Db2 user connected locally to DSNA is not subject to the PROD_EDIT behavior. If the FM/Db2 user specifies a location value to access a Db2® object at DSNB, the resulting editor session **is not** subject to the PROD_EDIT behavior.

An FM/Db2 user connected locally to DSNB is subject to the PROD_EDIT behavior. If the FM/Db2 user specifies a location value to access a Db2® object at DSNA, the resulting editor session IS subject to the PROD_EDIT behavior.

RBXWRKN

►►— RBXWRKN= *rxw_name* —►◄

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= *rgu_name* —►◄

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= *ld_line1* —▶◀

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= *ld_line2* —▶◀

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= *ld_line3* —▶◀

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.

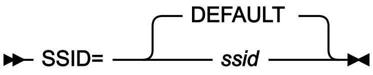
SLDJCL4

►► SLDJCL4= *ld_line4* ►◄

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** in Examples of FMN2SSDM macros on page 173.

STMJCL1

►► STMJCL1= *tm_line1* ►◄

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 173.

STMJCL2

►►─ STMJCL2= *tm_line2* ─►◄

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= *tm_line3* ─►◄

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.

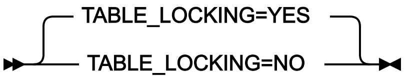
STMJCL4

►►─ STMJCL4= *tm_line4* ─►◄

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).

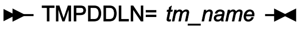
YES

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.

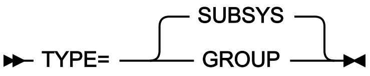
Specify a value for this keyword if you want to implement a site-specific policy for templates.

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 451, STMJCL2 on page 452,STMJCL3 on page 452, and STMJCL4 on page 452.

For an example, see Examples of FMN2SSDM macros on page 173.

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= *up_name* ─▶◀

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.

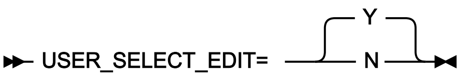
UNLUNLN

▶▶─ UNLUNLN= *uu_name* ─▶◀

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.

USER_SELECT_EDIT



USER_SELECT_EDIT

Specifies whether a user can select the Edit option for the *“Arbitrary SQL SELECT Statements”* system option.

Y

Specifies that a user may select the Edit option for the *“Arbitrary SQL SELECT Statements”* system option. The default is USER_SELECT_EDIT=Y.

N

Specifies that a user may not select the Edit option for the *“Arbitrary SQL SELECT Statements”* system option.

Specify USER_SELECT_EDIT=N when it considered necessary to prevent users accessing the FM/Db2 editor - in edit mode - when processing the result table for any user-specified SELECT statements. The FM/Db2 editor - in either browse or view mode - is still used to display the result table from any user-specified SELECT statement. Since browse and view modes do not allow changes to be saved to Db2®, users are unable to make changes to Db2® data.

Note that this setting is applied when connecting to the local Db2® system only. Consider the following example:

In the FMN2POPT,

- FMN2SSDM for Db2® system DSNA has USER_SELECT_EDIT=N, the CURRENT SERVER special register for DSNA is LOCN_A.
- FMN2SSDM for Db2® system DSNB has USER_SELECT_EDIT=Y, the CURRENT SERVER special register for DSNB is LOCN_B.

An FM/Db2 user locally connected to Db2® system DSNA can issue a statement like:

```
SELECT * FROM DSN81010.EMP,
```

but can only browse or view the result table.

The same user at DSNA can issue a statement like:

```
SELECT * FROM LOCN_B.DSN81010.EMP,
```

but can only view or browse the result table.

An FM/Db2 user locally connected to Db2® system DSNB can issue a statement like:

```
SELECT * FROM DSN81010.EMP,
```

and can browse, view or edit the result table.

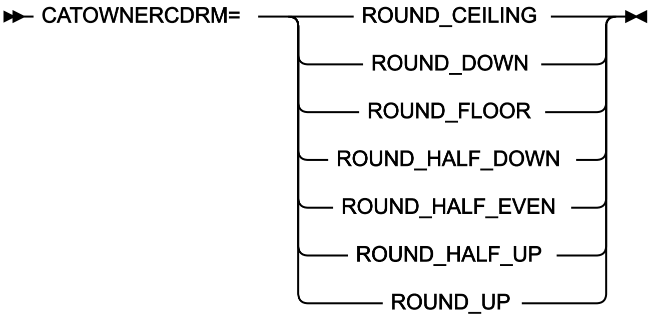
The same user at DSNB can issue a statement like:

```
SELECT * FROM LOCN_A.DSN81010.EMP
```

and can browse, view or edit the result table.

FMN2POPI

CATOWNERCDRM



CATOWNERCDRM

Specifies the current DECFLOAT rounding mode (*CDRM*) 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.

The 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.

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 the required CDRM before creating the views.
  The CDRM can be set using:

  ```
  SET CURRENT DECFLOAT ROUNDING MODE = value
  ```

  See the Db2® SQL Reference manual for a detailed explanation.
- 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).

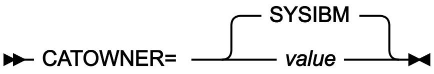
Important information about CDRM

The 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 (or later), regardless of whether or not there are DECFLOAT columns defined in the view.

CATOWNER



CATOWNER

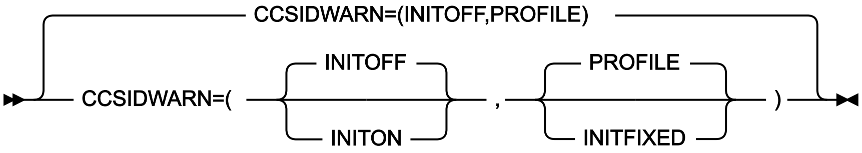
Specifies to FM/Db2 the owner of every Db2® catalog table or view accessed by FM/Db2.

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: 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 159.

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.

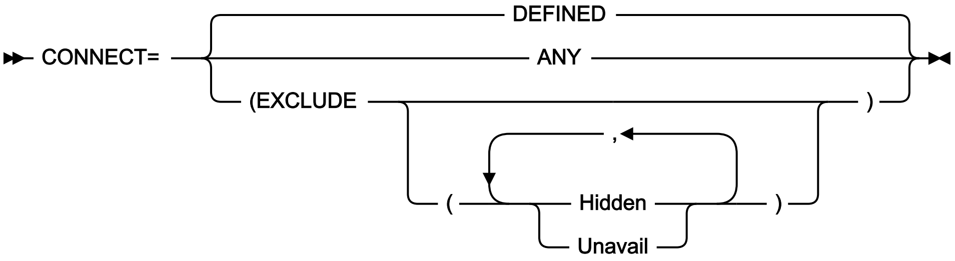
PROFILE, INITFIXED

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

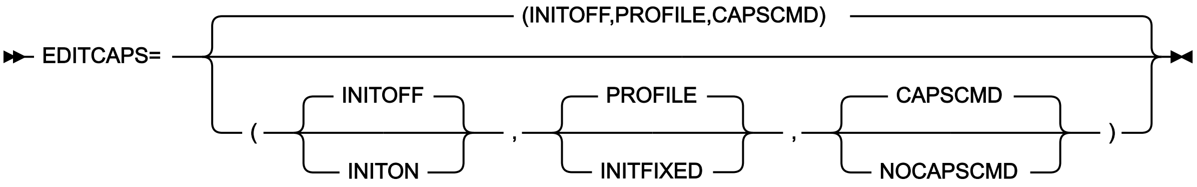
Will allow FM/Db2 to attempt to connect to any specified SSID, whether defined in the FMN2SSDM macro or not. FM/Db2 will only attempt connection to an SSID (either Db2® subsystem or data sharing group) that is active on the local z/OS® system.

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.

Note:

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= *module_name* —▶◀

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 16: Keyword values for the LANGUAGE option on page 102, for doptop34mod.

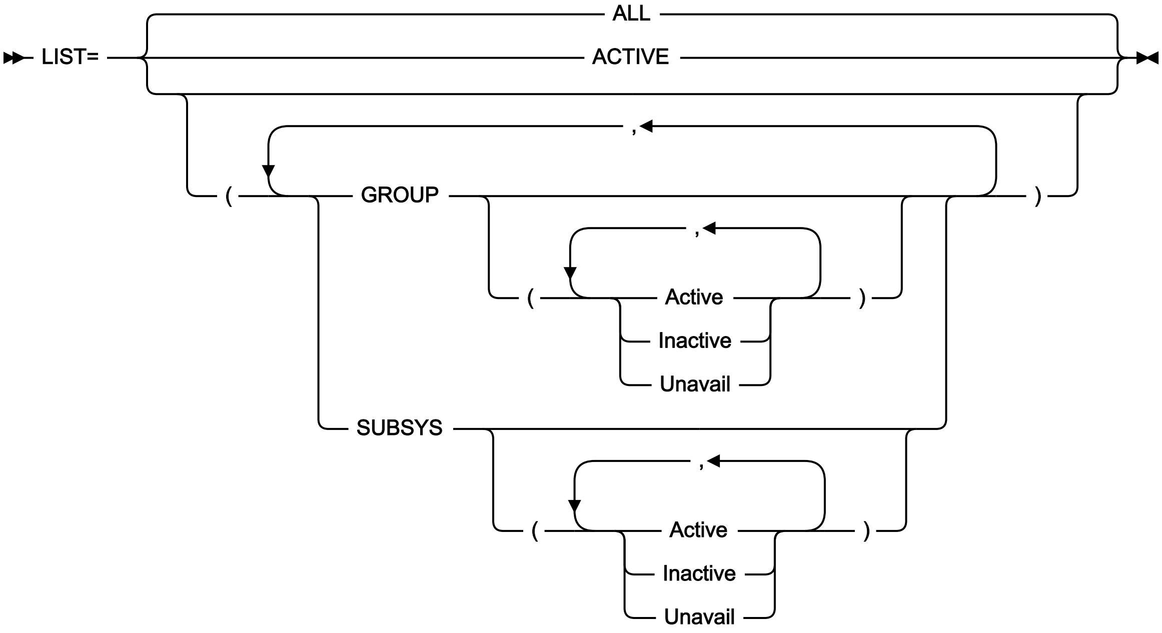
Note:

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, uppercase English, Japanese, and Korean. These modules are FMN2DENU, FMN2DENP, FMN2DJPN, and FMN2DKOR respectively.

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 the following option to ensure that the Japanese version of the data module is loaded.

```
OP34MOD=FMN2DJPN
```

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')      ❶
SUBSYS SUBNAME(DFA2) INITRTN(DSN3INI) INITPARM('DSN3EPX,-DFA2,S,DFA2') ❷
SUBSYS SUBNAME(DFS2) INITRTN(DSN3INI) INITPARM('DSN3EPX,-DFS2,S,DFG2') ❸
```

❶ defines a Db2® subsystem DFB2 that is not part of a Db2® data sharing group.

❷ defines a Db2® subsystem DFA2 that is also part of a Db2® data sharing group. The data sharing group is also called DFA2.

❸ defines a Db2® subsystem DFS2 that is also part of a Db2® data sharing group. The data sharing group name is DFG2.

❶ generates a single item on the FM/Db2 subsystem selection list. The corresponding FMN2SSDM macro entry should specify:

```
FMN2SSDM  SSID=DFB2,TYPE=SUBSYS, ...
```

❷ and ❸ generate two items each on the FM/Db2 subsystem selection list. For ❷ the corresponding FMN2SSDM macro entries are:

```
FMN2SSDM  SSID=DFA2,TYPE=SUBSYS, ...
FMN2SSDM  SSID=DFA2,TYPE=GROUP, ...
```

For ❸ 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 ❷ and ❸ 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, ❷), 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= *xxxxxxxx* —▸◂

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.

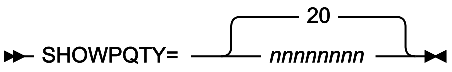
SHOWMGMTC

▸▸— SHOWMGMTC= *xxxxxxxx* —▸◂

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.

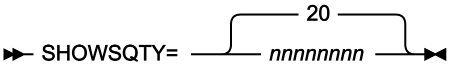
SHOWPQTY



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.

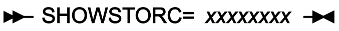
SHOWSQTY



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.

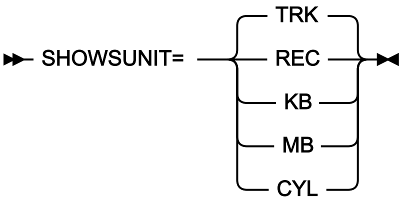
SHOWSTORC



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.

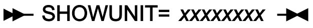
TRK

Track of a direct access storage device (DASD).

CYL

Cylinder of a DASD.

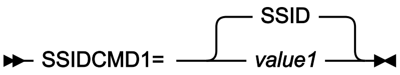
SHOWUNIT



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.

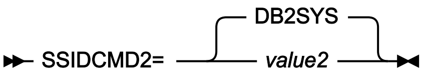
SSID

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 263.

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
- PADS

For the parameters that both macro statements support:

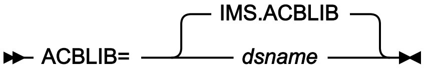
- The syntax of the parameter on the FMN1POPD macro statement is the same as the syntax of the parameter on the FMN1POPI macro statement.
- Each parameter is optional on both macro statements.
- Only the defaults are different. The default for the FMN1POPI parameter is the value specified on the FMN1POPD 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.

ACBLIB



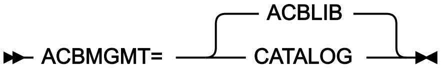
ACBLIB

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 the value on the FMN1POPD macro statement (if one is specified), or IMS.ACBLIB.

If you do not want the user to override the value you specify for this parameter, set UACBLIB=N.

ACBMGMT



ACBMGMT

Specifies how the active application control blocks (ACBs) are managed, as indicated by the ACBMGMT parameter in the CATALOG section of the DFSDFxxx PROCLIB member.

ACBLIB

The active ACBs are managed by your installation in ACBLIB data sets.

CATALOG

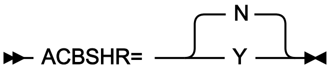
The active ACBs are managed by IMS™ in the IMS directory data sets of the IMS catalog.

FM/IMS retrieves the DBDs and PSBs from the IMS™ catalog through the IMS catalog API. FM/IMS provides dynamic PSB support by using the IMS Catalog Populate utility to add DOPT PSBs to the IMS catalog.

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 ACBLIB (otherwise).

The user cannot override the value you specify for this parameter.

ACBSHR



ACBSHR

Specifies whether the application control blocks (ACBs) that this IMS™ subsystem uses are also used by other IMS™ subsystems.

N

Only this IMS™ subsystem uses the ACBs.

Y

This IMS™ subsystem and other IMS subsystems use the ACBs.

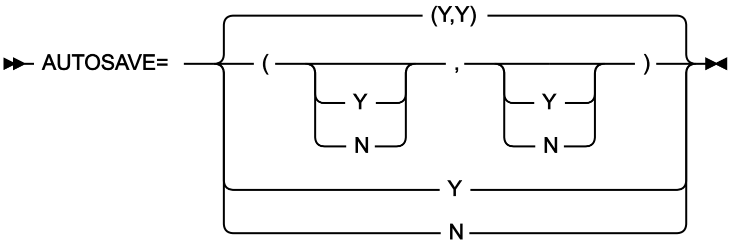
When IMS™ management of ACBs is enabled (ACBMGMT=CATALOG), the ACBs are in the IMS directory data sets of the IMS catalog. Set ACBSHR to Y when multiple IMS subsystems use the same IMS catalog.

When the ACBs are managed by your installation (ACBMGMT=ACBLIB), the ACBs are in ACBLIB data sets. Set ACBSHR to Y when multiple IMS subsystems use the same ACBLIB data sets.

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.

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

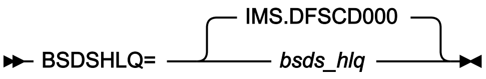
The automatic save function is not set on during an Edit.

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.

BSDSHLQ



BSDSHLQ

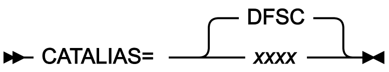
Specifies the high level qualifier for the IMS™ bootstrap data set. The qualifier '.BSDS' is appended to the specified qualifier to form the full name of the IMS™ bootstrap data set.

When IMS™ management of ACBs is enabled (ACBMGMT=CATALOG), the specified high-level qualifier is passed to the IMS™ catalog API (DFS3CATQ) and the IMS™ Catalog Library Builder utility (DFS3LU00).

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value on the FMN1POPD macro statement (if one is specified), or IMS.DFSCD000.

The user cannot override the value you specify for this parameter.

CATALIAS



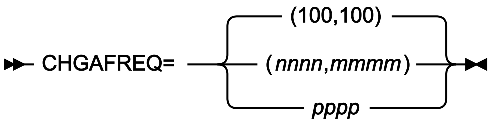
CATALIAS

The alias for the IMS™ catalog as specified on the ALIAS parameter in the CATALOG section of the DFSDFxxx PROCLIB member.

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 DFSC (otherwise).

The user cannot override the value you specify for this parameter.

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:

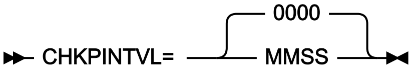
- Increments a count by 1 each time a segment is updated during a Change All operation or inserted during a Repeat All operation.
- Issues a checkpoint when the count is equal to the Change All/Repeat All checkpoint frequency.

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.

CHKPINTVL



CHKPINTVL

Specifies the time interval between checkpoints when the Edit/Browse BMP is waiting for a response from the user. The time interval is represented as zoned decimal digits of the form MMSS, where MM is minutes and SS is seconds.

The specified time interval must not be greater than 60 minutes and SS must be in the range 00 to 59.

If you specify 0000, then the BMP does not issue checkpoints when waiting for a response from the user.

CHKPINTVL equal to a non zero value is incompatible with each of these settings:

- EDITFREQ BMP value not equal to 1
- AUTOSAVE BMP value equal to N
- UAUTOSAV BMP value equal to Y

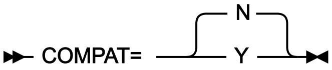
If you specify CHKPINTVL equal to a nonzero value and one or more of these incompatible settings, FM/IMS uses your CHKPINTVL setting and ignores the incompatible settings, using these compatible settings instead:

- EDITFREQ BMP value equal to 1
- AUTOSAVE BMP value equal to Y
- UAUTOSAV BMP value equal to N

The CHKPINTVL 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 "0000" otherwise.

The user cannot override the value you specify for this parameter.

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 FMNIMSIN input of batch functions that run in DLI mode.

Y

The IMSID parameter in the FMNIMSIN 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 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 the new version of FM/IMS.

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.

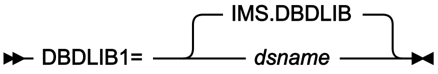
If you specify COMPAT=Y, JCL generated by versions of FM/IMS earlier than version 8 should run without modification.

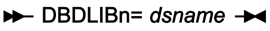
However, COMPAT=N is recommended.

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.

DBDLIB*n*





DBDLIB*n*

Where *n* is a number in the range 1 to 6, specifies the names of the load libraries that contain the database definitions (DBDs) that FM/IMS and IMS™ are to use.

The specified data sets are used when the ACBs are managed by your installation (ACBMGMT=ACBLIB).

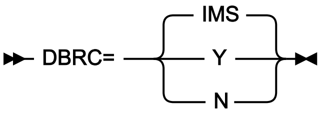
These parameters are optional. But if you specify a DBDLIB parameter, you must also specify the DBDLIB parameters that precede it. So, for example, if you specify the DBDLIB3 parameter, you must also specify the DBDLIB1 parameter and the DBDLIB2 parameter.

If no DBDLIB parameters are specified on the FMN1POPI macro statement, FM/IMS uses the DBDLIB values on the FMN1POPD macro statement (if specified), otherwise it uses IMS™.DBDLIB.

If you do not want the user to override the values you specify for these parameters, set UDBDLIB=N.

Note: When IMS™ management of ACBs is enabled (ACBMGMT=CATALOG) these parameters are not used, because FM/IMS and IMS™ get the DBDs from the IMS™ catalog that the subsystem uses.

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.

The DBRC parameter is used when the ACBs are managed by your installation (ACBMGMT=ACBLIB) and the database being accessed is not a HALDB or a HALDB partitioned secondary index.

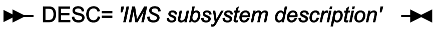
This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or IMS™ (otherwise).

If you do not want the user to override the value you specify for this parameter, set UDBRC=N.

Notes:

- When the database being accessed is a HALDB, DBRC is used unless the database is the IMS™ catalog and the REGCATLG parameter is set to N.
- When the database being accessed is a HALDB partitioned secondary index, DBRC is used unless the database is the IMS™ catalog's secondary index and the REGCATLG parameter is set to N.
- When IMS™ management of ACBs is enabled (ACBMGMT=CATALOG) DBRC is used unless the REGCATLG parameter is set to N.

DESC



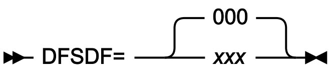
DESC

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.

DFSDF



DFSDF

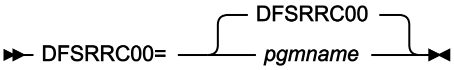
Specifies the 3-character suffix of the DFSDFxxx member of the IMS™ PROCLIB data set that contains the settings and attributes of the IMS™ catalog. This member is used by functions that run in DLI mode when IMS™ management of ACBs is enabled (ACBMGMT=CATALOG) and the IMS™ Catalog Definition exit routine (DFS3CDX0) is not used.

When the member is used, ensure that it has a CATALOG section in which the CATALOG parameter is set to YES, the ACBMGMT parameter is set to CATALOG and the ALIAS parameter is set to the alias for the IMS catalog. If the IMS catalog database and its secondary index are not registered with DBRC (REGCATLG=N), ensure that their names are specified on the UNREGCATLG parameter in the DATABASE section of the member.

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 000 (otherwise).

The user cannot override the value you specify for this parameter.

DFSRRC00



DFSRRC00

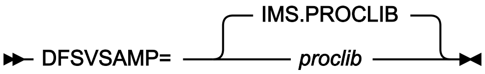
Specifies the name of the program that FM/IMS attaches to invoke IMS™. This will usually be the IMS™ region controller program, DFSRRC00.

If a different program is to be used, it must be in one of the RESLIB data sets. For the RESLIB parameters see RESLIBn on page 503.

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

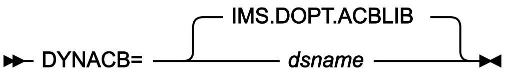
Specifies the name of the data set that contains the parameters defining the IMS™ buffer sub pools for VSAM and OSAM data sets. When an FM/IMS function is run in DLI mode, the specified data set is allocated to the DFSVSAMP DD.

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 on the FMN1POPD macro statement (if one is specified), 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 ACBLIB data set into which the ACB maintenance utility generates the DOPT PSBs.

This data set is used by functions that use dynamic PSBs to access databases in BMP mode. If IMS™ management of ACBs is enabled (ACBMGMT=CATALOG) it is also used by functions that use dynamic PSBs to access databases in DLI mode.

When the ACBs are managed by your installation (ACBMGMT=ACBLIB), the specified data set must be concatenated with the primary ACBLIB data set in the IMS execution JCL (see Providing a DOPT ACBLIB data set on page 262).

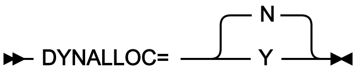
When IMS management of ACBs is enabled (ACBMGMT=CATALOG), the specified data set does not have to be concatenated with the primary ACBLIB data set. Instead, the specified data set is used as input for the IMS Catalog Populate utility (DFS3PU00), which adds the DOPT PSBs to the IMS catalog.

If APAR PH17975 has been applied or XDOPTLB=Y has been specified, you must specify a data set that only FM/IMS uses.

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:

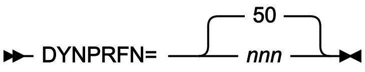
- The Database Data Set Specification panel is displayed in the FM/IMS dialog when a DLI region type and a non-HALDB database is specified on the Entry panel.
- The JCL generated by FM/IMS dialogs includes DD statements for non-HALDB database data sets.
- In an Edit or Browse, FM/IMS allocates the database 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.

DYNPRFN



DYNPRFN

Specifies the number of dynamic PSBs required by concurrent FM/IMS BMP users.

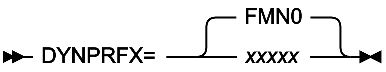
The number must be in the range 1 to 999.

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 261.)

DYNPRFX



DYNPRFX

Specifies the first 1 - 5 characters of the dynamic PSB names used by FM/IMS functions.

This parameter is used by functions that run in BMP mode and also by functions that run in DLI mode.

For BMP mode, the dynamic PSB names are obtained by appending a 3-digit number to the value you specify in this parameter. The number can be in the range 001 to the number specified in the DYNPRFN parameter. For example, if you specify DYNPRFX=FMN and DYNPRFN=3, FM/IMS uses FMN001, FMN002, and FMN003 for the dynamic PSB names.

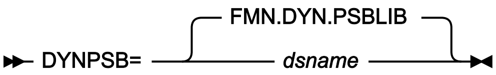
Note: If you want to run functions that use dynamic PSBs in BMP mode, you need to specify APPLCTN macro statements for each dynamic PSB name that FM/IMS is to use. (See Customizing IMS to support the use of dynamic PSBs in BMP mode on page 261.)

For DLI mode, the dynamic PSB names are obtained by appending '000' to the value you specify in this parameter. For example, if you specify DYNPRFX=FMN, FM/IMS uses FMN000 for the dynamic PSB name.

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.

DYNPSB



DYNPSB

Specifies the name of the PDSE program library in which FM/IMS stores the dynamic PSBs that it builds.

The DYNPSB parameter is used when PADS=Y is specified on the FMN1POPD macro statement.

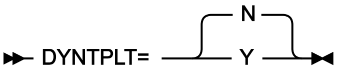
If the IMS™ subsystem supports the use of dynamic PSBs in BMP mode, use of this library should be restricted to this subsystem.

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 FMN.DYN.PSBLIB (otherwise).

The user cannot override the value you specify for this parameter.

Note: When the PADS parameter is set to N, the dynamic PSBs that FM/IMS builds are stored in temporary data sets with system generated names.

DYNTPLT



DYNTPLT

Specifies whether the dynamic generation of FM/IMS templates is enabled.

N

FM/IMS functions that have requested a new or an existing view or criteria set don't attempt to generate a template for the database that the function is accessing.

Y

FM/IMS functions that have requested a new or an existing view or criteria set attempt to generate a template for the database that the function is accessing.

Functions generate the template from field definitions in the DBDs. Therefore, it is a requirement that application-defined fields must have been added to your DBDs.

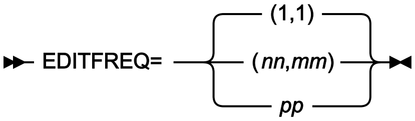
If application-defined fields have not been added to your DBDs, set DYNTPLT=N, otherwise attempts to generate a template for the database would fail.

When ACBs are managed by IMS™ (ACBMGMT=CATALOG), functions get the field definitions from the IMS™ catalog . When ACBs are managed by your installation (ACBMGMT=ACBLIB), functions get the field definitions from the DBD libraries for the subsystem.

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).

Users 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.

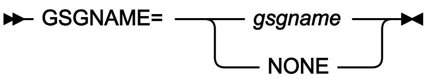
The valid range is 1 to 99.

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.

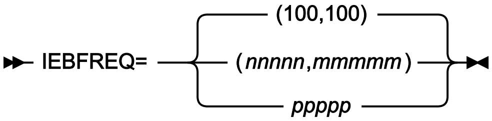
Otherwise, do not specify a value for GSGNAME.

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.

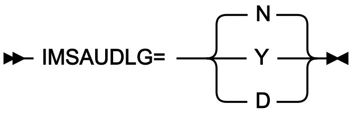
The valid range is 1 to 99999.

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

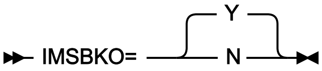
Audit logging is enforced during edit, and at the conclusion of the edit session an audit report job is submitted to report on the changes. This job can be customized by changing member FMN1FTAD from FMN.SFMNSLIB to specify the job card and the reporting options you require.

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

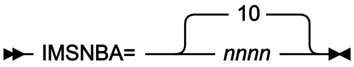
A dynamic backout is not performed when an IMS™ pseudoabend occurs in an FM/IMS function running in DLI mode.

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

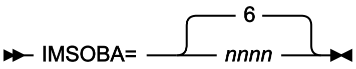
Specifies the number of Fast Path database buffers to be made available in the Common Service Area (CSA) when a Fast Path region is activated. The number specified is used by FM/IMS functions when they access Fast Path databases.

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

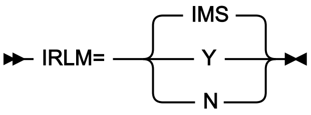
Specifies the number of additional page-fixed buffers to be made available to a Fast Path region if the normal allotment (specified or defaulted in IMSNBA) is used. The number specified is used by FM/IMS functions when they access Fast Path databases.

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= *user_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= *xxxxxxxx* —▶◀

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.

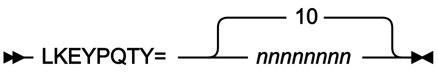
LKEYMGMTC

▶▶— LKEYMGMTC= *xxxxxxxx* —▶◀

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.

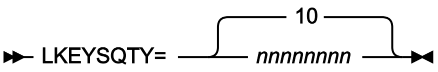
LKEYPQTY



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.

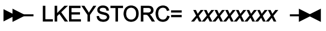
LKEYSQTY



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.

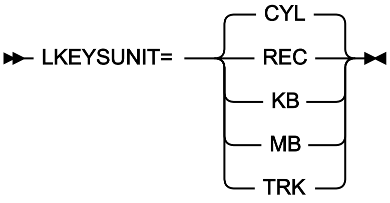
LKEYSTORC



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.

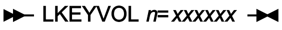
TRK

Track of a direct access storage device (DASD).

CYL

Cylinder of a DASD.

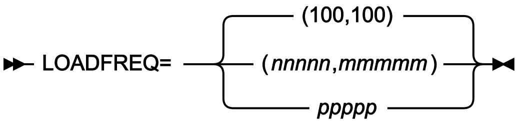
LKEYVOL*n*



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.

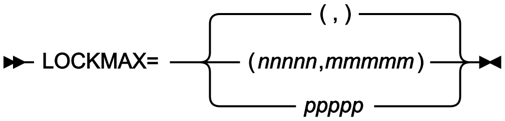
The valid range is 1 to 99999.

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.

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= xxxxxxxx ─►◄
```

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.

LOGDSN

```
                   ┌─ &&UID..FMNLOG.&&YYMMDD..&&HHMMSS..&&DBNAME. ─┐
►►─ LOGDSN= ───────┴──────────────── name-pattern ─────────────────┴─►◄
```

LOGDSN

Specifies the name pattern that is used by FM/IMS functions to generate IMS™ log data set names.

The name pattern can consist of up to five qualifiers. For each qualifier you can specify either the required character string or a symbol. The data set name is generated by replacing the symbols in the name pattern with their runtime values.

For the remainder of this parameter description entry, "the function" refers to the FM/IMS function that is to use the IMS™ log.

The parameter supports these symbols:

&&DBNAME.

The name of the primary database that the function accesses.

&&PREFIX.

Either the user's TSO prefix (if the name is being generated in a TSO environment) or the value that is specified in the TSOPREFIX parameter (otherwise). The value might be null.

&&FUNCOD.

The code for the function. The following are the codes for the functions that might use an IMS™ log:

IE

Edit

IB

Browse

DIB

Initialize

IXB

Extract

ILB

Load

IPR

Print

IEB

Batch Edit

IBB

Batch Browse

&&HHMMSS.

Thhmmss where hhmmss is the time of day that the IMS™ log data set name is generated expressed in hours (HH), minutes (MM), and seconds (SS).

&&SSID.

The name of the IMS™ subsystem that the function is to access.

&&UID.

Either &&PREFIX. (if its value is not null) or the User ID (otherwise).

&&USER.

User ID

&&YYMMDD.

Dyymmdd where yymmdd is the date that the IMS™ log data set name is generated expressed as a 2-digit year (YY), month (MM), and day (DD).

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value that is specified on the FMN1POPD macro statement (if specified on that statement), or &&UID..FMNLOG.&&YYMMDD..&&HHMMSS..&&DBNAME. (otherwise).

If you do not want users to override the value you specify for this parameter, set ULOGDSN=N.

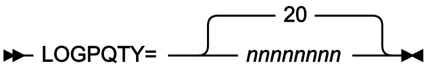
LOGMGMTC

```
▶▶— LOGMGMTC= xxxxxxxx —▶◀
```

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.

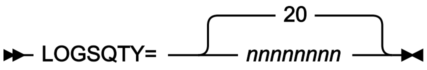
LOGPQTY



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.

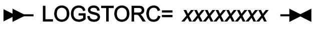
LOGSQTY



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.

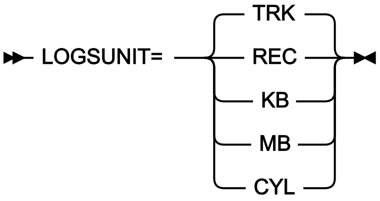
LOGSTORC



LOGSTORC

Specifies the default SMS storage class that FM/IMS uses when allocating the IMS™ log. The maximum length is 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.

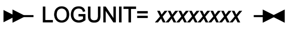
TRK

Track of a direct access storage device (DASD).

CYL

Cylinder of a DASD.

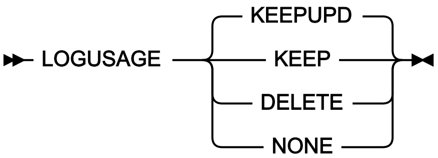
LOGUNIT



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.

LOGUSAGE



LOGUSAGE

Specifies whether FM/IMS functions which are using PSBs that have update intent use an IMS™ log when they are run in DLI mode and, if they do use a log, whether the log is kept when the function ends.

KEEP

Use an IMS™ log. Keep the log when the function ends.

KEEPUPD

Use an IMS™ log. Keep the log if the function updates databases or does not end normally.

DELETE

Use an IMS™ log. Keep the log only if the function does not end normally.

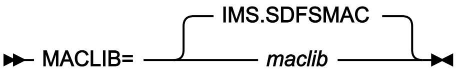
NONE

Do not use an IMS™ log.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value that is specified on the FMN1POPD macro statement (if specified on that statement), or KEEPUPD (otherwise).

If you do not want the user to override the value you specify for this parameter, set ULOGUSAG=N.

MACLIB



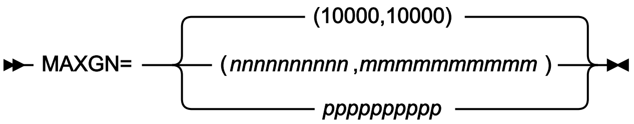
MACLIB

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 on the FMN1POPD macro statement (if one is specified), 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.

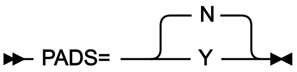
If you specify 0 for MAXGN there is no limit on the number of Get Next calls that a Find/Change command can issue.

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.

PADS



PADS

Specifies whether Program Access to Data Sets (PADS) is used in the environments in which FM/IMS is run.

N

RACF® PADS is not used in the environments in which FM/IMS is run.

Y

RACF® PADS is used in the environments in which FM/IMS is run.

When PADS=N is specified, FM/IMS functions store dynamic PSBs in temporary data sets with system-generated names.

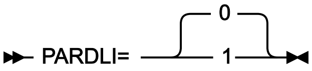
When PADS=Y is specified, FM/IMS functions store dynamic PSBs in the dynamic PSB library specified on the DYNPSB parameter.

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.

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.

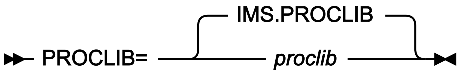
This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or 0 (otherwise).

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 PARDLI considerations on page 266.

PROCLIB



PROCLIB

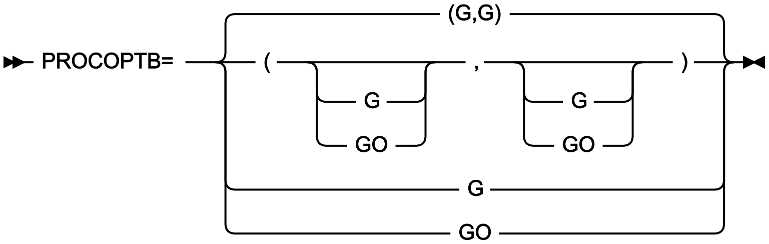
Specifies the name of the IMS™ PROCLIB data set that contains the required DFSDFxxx member (see DFSDF on page 476).

The specified data set is allocated to the PROCLIB DD when an FM/IMS function is run in DLI mode and IMS™ management of ACBs is enabled (ACBMGMT=CATALOG).

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value on the FMN1POPD macro statement (if one is specified), or IMS.PROCLIB.

The user cannot override the value you specify for this parameter.

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

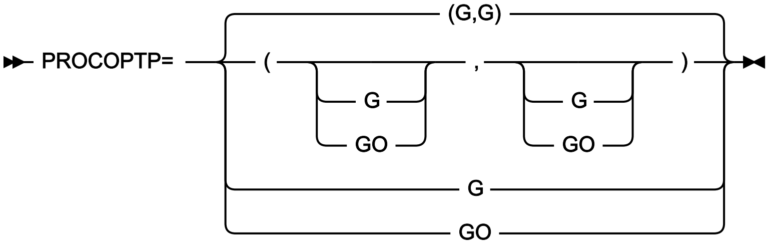
The Browse reads without integrity

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

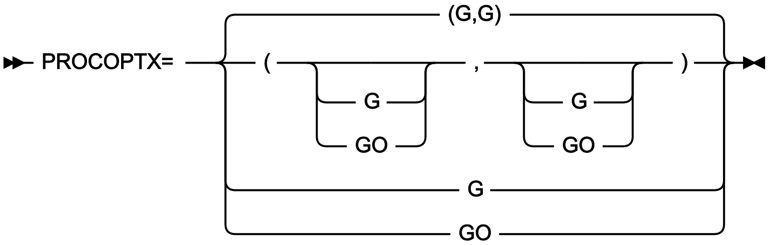
The Print reads without integrity

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

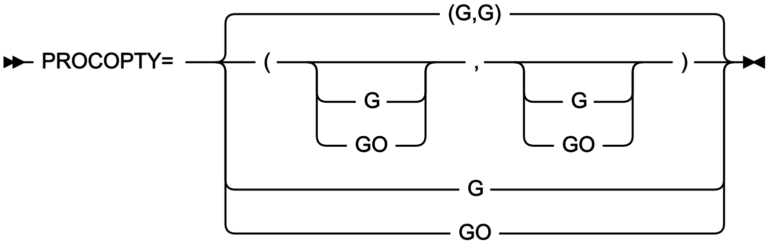
The Extract reads without integrity

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

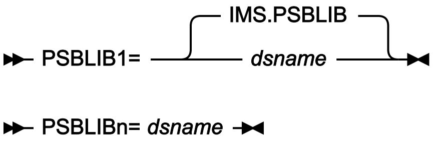
The Batch Browse reads without integrity.

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 load libraries that contain the program specification blocks (PSBs) that FM/IMS and IMS™ are to use.

The specified data sets are used when the ACBs are managed by your installation (ACBMGMT=ACBLIB).

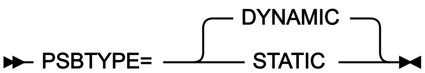
These parameters are optional. But if you specify a PSBLIB parameter, you must also specify the PSBLIB parameters that precede it. So, for example, if you specify the PSBLIB3 parameter, you must also specify the PSBLIB1 parameter and the PSBLIB2 parameter.

If no PSBLIB parameters are specified on the FMN1POPI macro statement, FM/IMS uses either the PSBLIB values on the FMN1POPD macro statement (if specified), or IMS.PSBLIB.

If you do not want the user to override the values you specify for these parameters, set UPSBLIB=N.

Note: When IMS™ management of ACBs is enabled (ACBMGMT=CATALOG), these parameters are not used, as FM/IMS and IMS™ get the PSBs from the IMS™ catalog that the subsystem uses.

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

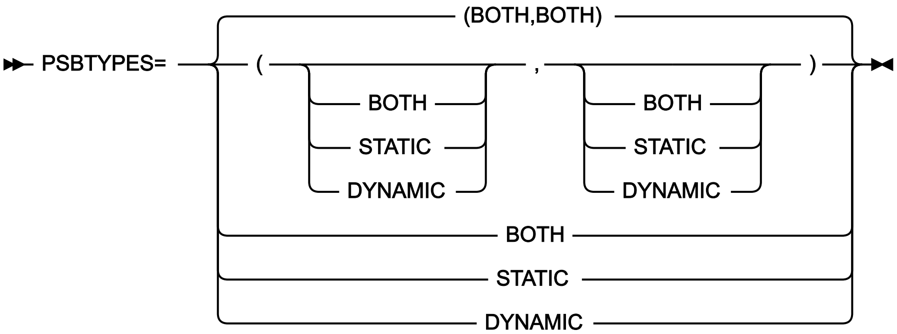
The batch functions use an existing PSB that the user specifies in the JCL.

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

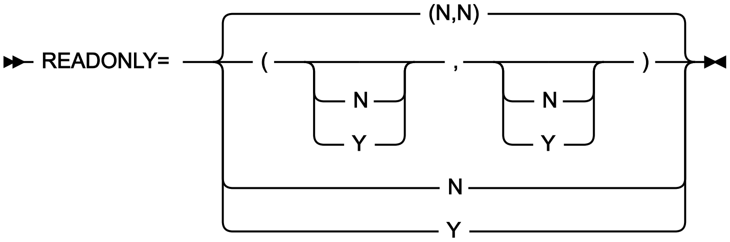
Only dynamic PSBs can be used.

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.

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

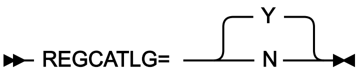
The IMS™ subsystem is to be defined as Read Only.

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.

REGCATLG



REGCATLG

Specifies whether the IMS™ catalog is registered with Database Recovery Control (DBRC).

Y

The IMS™ catalog is registered with DBRC. The IMS catalog partitions are defined with the DBRC utility (DSPURX00). The catalog partition definitions are stored in the RECON data sets.

N

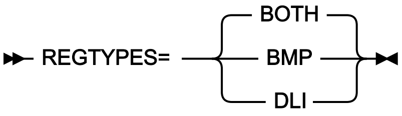
The IMS™ catalog is not registered with DBRC. The IMS catalog partitions are defined with the IMS catalog partition definition utility (DFS3UCD00). The catalog partition definitions are stored in the catalog partition definition data set.

This parameter is used by FM/IMS functions run in DLI mode when IMS™ management of ACBs is enabled (ACBMGMT=CATALOG) or the database being accessed is the IMS catalog or its secondary index. When REGCATLG=Y, functions use DBRC and database data set names are obtained from the RECON data sets.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value on the FMN1POPD macro statement (if one is specified), or Y.

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.

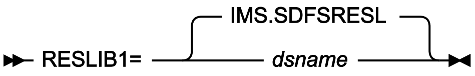
DLI

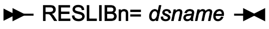
Databases in the IMS™ subsystem can be accessed in DLI mode only.

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.

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.

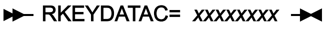
These parameters are optional. But if you specify a RESLIB parameter, you must also specify the RESLIB parameters that precede it. So, for example, if you specify the RESLIB3 parameter, you must also specify the RESLIB1 parameter and the RESLIB2 parameter.

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= xxxxxxxx ─▸◂
```

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

```
                    ┌──── 10 ────┐
▸▸─ RKEYPQTY= ──────┴─ nnnnnnnn ─┴──▸◂
```

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

```
                    ┌──── 10 ────┐
▸▸─ RKEYSQTY= ──────┴─ nnnnnnnn ─┴──▸◂
```

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.

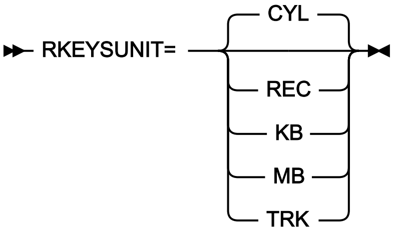
RKEYSTORC

```
▸▸─ RKEYSTORC= xxxxxxxx ─▸◂
```

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.

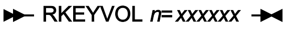
TRK

Track of a direct access storage device (DASD).

CYL

Cylinder of a DASD.

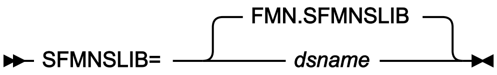
RKEYVOL*n*



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= xxxx ─▶◀
```

SSID

Specifies the 1 to 4-character IMS™ subsystem identifier.

This parameter is required, and can only be specified on the FMN1POPI macro statement.

TIMEOUTI

```
                   ┌── 0000 ──┐
▶▶─ TIMEOUTI= ─────┴── HHMM ──┴──▶◀
```

TIMEOUTI

Specifies the time interval that a user has to respond before an Edit/Browse BMP is timed out. The time interval is represented as zoned decimal digits of the form HHMM, where HH is hours and MM is minutes.

The specified time interval must not be greater than 24 hours and MM must be in the range 00 to 59.

If you specify 0000, then the BMP is not timed out.

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 "0000" otherwise.

The user cannot override the value you specify for this parameter.

TMINAME

```
▶▶─ TMINAME= 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.

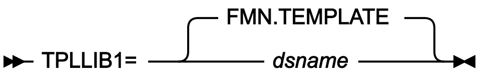
This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value specified on the FMN1POPD macro statement (if specified on that statement), or a null value (otherwise).

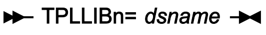
When the activity is tracked by RSR and a null value is passed to the IMS™ region controller, IMS™ uses:

- 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*

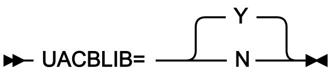
Where n is a number in the range 1 to 6, specifies the names of the data sets containing the templates that FM/IMS functions are to use.

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.

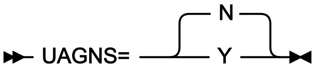
N

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).

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.

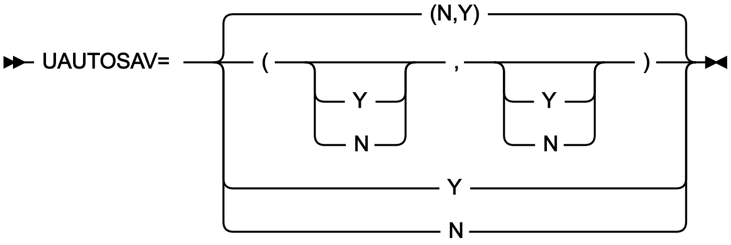
Y

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.

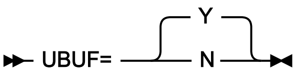
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

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).

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

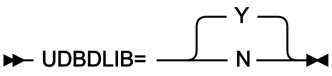
The user cannot specify the BUF parameter.

If you specify UBUF=N:

- The **BUF** parameter field on the DLI Mode Parameters panel is protected.
- IMSBUF= statements in the FMNIMSIN 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).

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

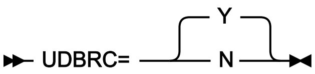
The user cannot override the DBDLIBn data set parameters.

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

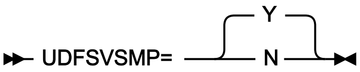
The user cannot override the DBRC parameter.

If you specify UDBRC=N:

- The **DBRC** option field on the DLI Mode Parameters panel is protected.
- DBRC= statements in the FMNIMSIN 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

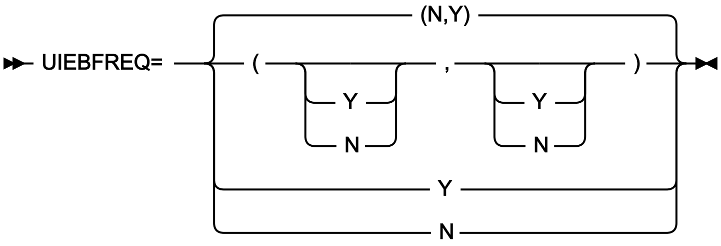
The user cannot override the DFSVSAMP and VSMPMEM parameters.

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 FMNIMSIN 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).

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:

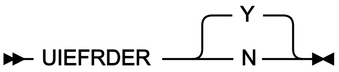
- The **Batch Edit** checkpoint frequency field on the BMP Mode Options panel is protected.
- CHKPFREQ= statements in the FMNIMSIN input of Batch Edit jobs run in BMP mode are ignored.
- No CHKPFREQ= statements are included in the JCL generated by the Batch Edit dialog when the Batch Edit is to run in 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 FMNIMSIN 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).

UIEFRDER



UIEFRDER

Specifies whether FM/IMS batch functions which are using PSBs that have update intent, use the IEFRDER DD specified in the JCL when they are run in DLI mode.

Y

Use the IEFRDER DD.

N

Do not use the IEFRDER DD.

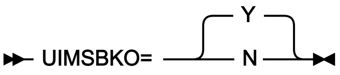
If you specify UIEFRDER=N, then batch functions which are using PSBs that have update intent:

- Check whether the IEFRDER DD was allocated by the job step and, if it was, frees this allocation.
- Dynamically allocates an IEFRDER DD according to the LOG parameters specified in the FMNIMSIN input (when the parameters are specified and are not fixed) or the subsystem defaults for the LOG parameters (otherwise).

If you specify UIEFRDER=Y, then batch functions which are using PSBs that have update intent use the IEFRDER DD allocated by the job step (when the IEFRDER DD is allocated) or dynamically allocate an IEFRDER DD according to the LOG parameters specified in the FMNIMSIN input (when the parameters are specified and are not fixed) or the subsystem defaults for the LOG parameters (otherwise).

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value that is specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

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

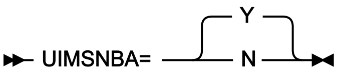
The user cannot override the IMSBKO parameter.

If you specify UIMSBKO=N:

- The **Dynamic backout** option field on the DLI Mode Parameters panel is protected.
- IMSBKO= statements in the FMNIMSIN 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

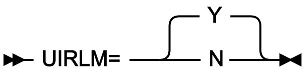
The user cannot override the IMSNBA and IMSOBA parameters.

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 FMNIMSIN 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).

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

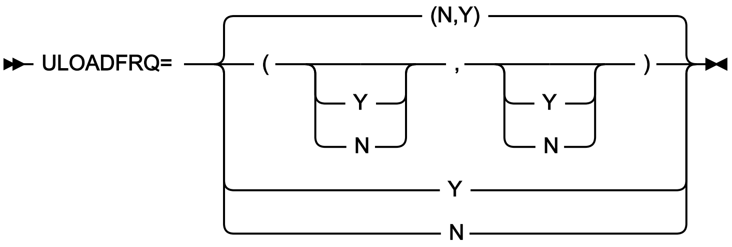
The user cannot override the IRLM and IRLMNAME parameters.

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 FMNIMSIN 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:

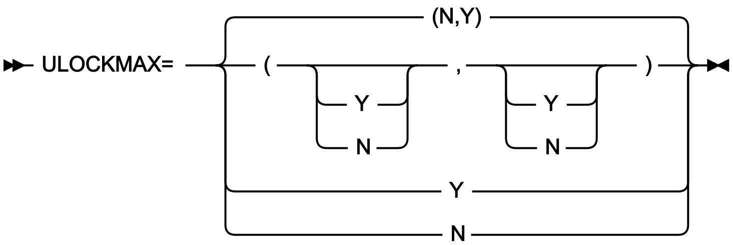
- The **Load** checkpoint frequency field on the BMP Mode Options panel is protected.
- CHKPFREQ= statements in the FMNIMSIN input of Load jobs run in BMP mode are ignored.
- No CHKPFREQ= statements are included in the JCL generated by the Load dialog when the Load is to run in BMP mode.

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 FMNIMSIN 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).

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 FMNIMSIN 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.

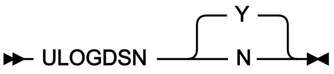
If you specify ULOCKMAX=N for DLI mode:

- The **LOCKMAX** parameter field on the DLI Mode Parameters panel is protected.
- LOCKMAX= statements in the FMNIMSIN 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).

ULOGDSN



ULOGDSN

Specifies whether the user can override the name pattern that is specified in the LOGDSN parameter.

Y

The user can override the LOGDSN parameter.

N

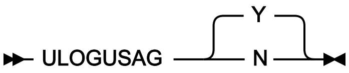
The user cannot override the LOGDSN parameter.

If you specify ULOGDSN=N:

- The **IMS log Data set name pattern** field on the **DLI Mode Data Sets 2** panel is protected.
- LOGDSN= statements in the FMNIMSIN input of batch jobs are ignored.
- No LOGDSN= 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 that is specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

ULOGUSAG



ULOGUSAG

Specifies whether the user can override the value that is specified for the LOGUSAGE parameter.

Y

The user can override the LOGUSAGE parameter.

N

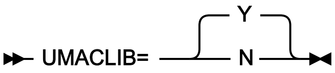
The user cannot override the LOGUSAGE parameter.

If you specify ULOGUSAG=N:

- The **IMS log option** field on the **DLI Mode Parameters** panel is protected.
- LOGUSAGE= statements in the FMNIMSIN input of batch jobs are ignored.
- No LOGUSAGE= 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 that is specified on the FMN1POPD macro statement (if specified on that statement), or Y (otherwise).

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

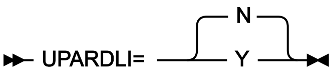
The user cannot override the MACLIB parameter.

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 FMNIMSIN 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).

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

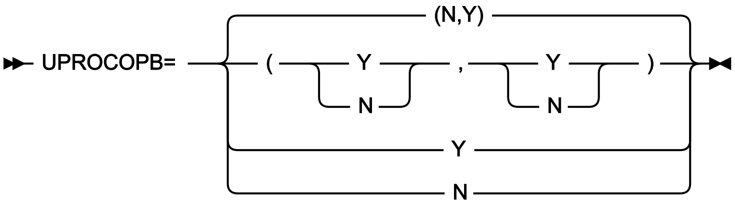
The user can override the PARDLI parameter.

If you specify UPARDLI=N:

- The **PARDLI** option field on the BMP Mode Parameters panel is protected.
- PARDLI= statements in the FMNIMSIN 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).

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.

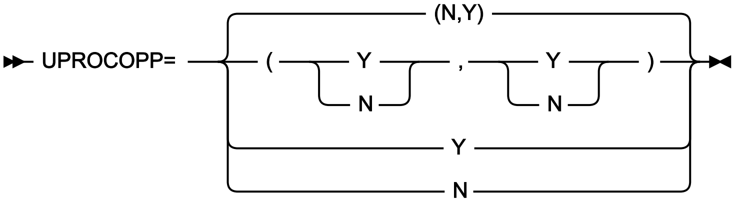
When you specify two values in parentheses, separated by a comma, the first value is used for BMP mode and the second value is used for DLI mode. When you specify a single value, this is used for both BMP and DLI modes.

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:

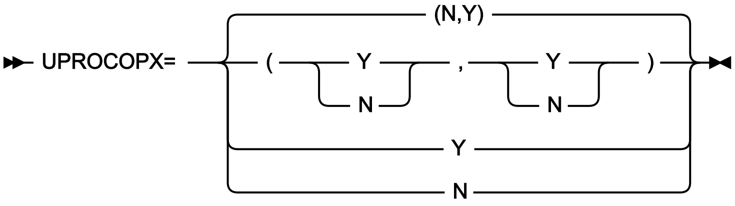
- The **Print** PSB Processing Option field on the BMP Mode Options panel is protected.
- PROCOPT= statements in the FMNIMSIN input of Print jobs run in BMP mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Print dialog when the Print is to run in BMP mode.

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 FMNIMSIN 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:

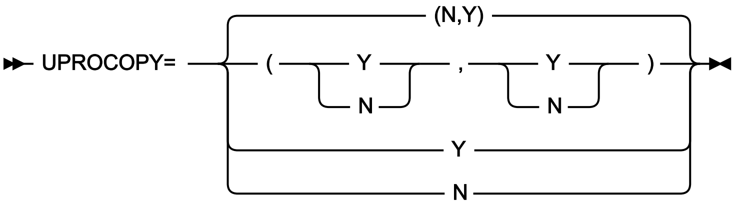
- The **Extract** PSB Processing Option field on the BMP Mode Options panel is protected.
- PROCOPT= statements in the FMNIMSIN input of Extract jobs run in BMP mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Extract dialog when the Extract is to run in BMP mode.

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 FMNIMSIN 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:

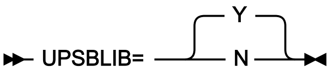
- The **Batch Browse** PSB Processing Option field on the BMP Mode Options panel is protected.
- PROCOPT= statements in the FMNIMSIN input of Batch Browse jobs run in BMP mode are ignored.
- No PROCOPT= statements are included in the JCL generated by the Batch Browse dialog when the Batch Browse is to 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 FMNIMSIN 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

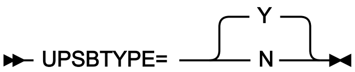
The user cannot override the PSBLIBn data set parameters.

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.

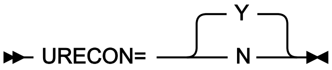
This parameter can only be specified on the FMN1POPD statement.

If you specify UPSBTYPE=N:

- PSBTYPE= statements in the FMNIMSIN 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

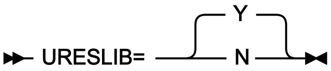
The user cannot specify RECON data sets.

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 FMNIMSIN 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).

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

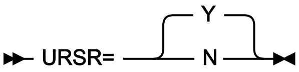
The user cannot override the RESLIBn data set parameters.

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 FMNIMSIN 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).

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

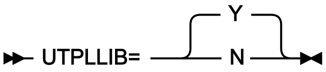
The user cannot override the GSGNAME and TMINAME parameters.

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 FMNIMSIN 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

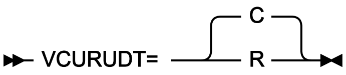
The user cannot override the TPLLIBn data set parameters.

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).

VCURUDT



VCURUDT

Specifies the usage rules that apply to views and criteria sets when the function generates a template for the database.

C

A view or criteria set can be used with a database only if it provides one or more layouts for each segment in the database. If it does provide one or more layouts for each segment in the database, the function uses a composite view or criteria set that is created by adding the view or criteria set’s selection information to the template that was generated for the database.

R

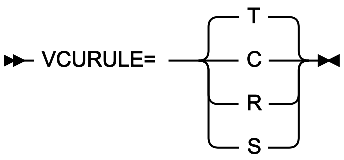
A view or criteria set can be used with a database only if it has the same fields as the template that was generated for the database. If it does have the same fields as the generated template, the function uses the view or criteria set that is specified by the user.

VCURUDT=C is the default setting and the recommended setting.

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 C (otherwise).

The user cannot override the value you specify for this parameter.

VCURULE



VCURULE

Specifies the view or criteria set usage rule.

C

A view or criteria set can be used with a database only if it provides one or more layouts for each segment in the database. If it does provide one or more layouts for each segment in the database, the function uses a composite view or criteria set that is created by adding the view or criteria set's selection information to the static template for the database.

R

A view or criteria set can be used with a database only if it has the same fields as the static template for the database. If it does have the same fields as the static template for the database, the function uses the view or criteria set that is specified by the user.

S

A view or criteria set can be used with a database only if it provides one or more layouts for each segment in the database. If it does provide one or more layouts for each segment in the database, the function uses the view or criteria set specified by the user.

The static template for the database is not used.

T

A view or criteria set can be used with a database only if these two conditions are true:

- The view or criteria set provides one or more layouts for each segment in the database.
- The view or criteria set was created for the database OR the view or criteria set and the static template for the database have the same fields.

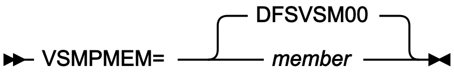
If the view or criteria set satisfies these conditions, the function uses the view or criteria set that is specified by the user.

Note: VCURULE=T is the default setting, but VCURULE=C is the recommended setting.

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

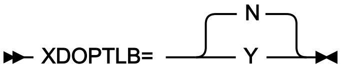
If the data set specified in the DFSVSAMP parameter is a PDS or PDSE, VSMPMEM specifies the member of that data set which contains the parameters defining the IMS™ buffer subpools for OSAM and VSAM data sets.

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



Note: APAR PH17975 makes setting the XDOPTLB parameter unnecessary. The following sections describe how XDOPTLB works when APAR PH17975 is not applied, and explain a check that you might need to perform when APAR PH17975 is applied.

XDOPTLB parameter when APAR PH17975 is not applied

This section describes the effect of the XDOPTLB parameter when APAR PH17975 has *not* been applied.

XDOPTLB

Specifies whether the ACB Maintenance utility deletes all PSBs and DBDs from the DOPT ACBLIB data set and makes all of their space available for reuse, before building blocks for the dynamic PSB.

N

FM/IMS deletes the PSB from the DOPT ACBLIB data set after the BMP terminates. No other modules are deleted from the DOPT ACBLIB data set and no space is made available for reuse.

Y

FM/IMS executes the ACB Maintenance utility with BUILD PSB=ALL specified on the SYSIN control statement. With this control statement specified, the utility deletes all PSBs and DBDs (and any other modules) from the DOPT ACBLIB data set and makes all of their space available for reuse, before building control blocks for the dynamic PSB.

If this option is selected, you must specify a DOPT ACBLIB that is only used by FM/IMS.

This parameter is optional. If it is not specified on the FMN1POPI macro statement, FM/IMS uses either the value on the FMN1POPD macro statement (if one is specified), or N.

The user cannot override the value you specify for this parameter.

XDOPTLB parameter when APAR PH17975 is applied

When APAR PH17975 has been applied, FM/IMS always follows the behavior for XDOPTLB=Y. Therefore, if XDOPTLB was previously set to Y, no action is required now.

If XDOPTLB was previously set to N, check whether the DOPT ACBLIB data set specified in the DYNACB parameter is used by any other product.

If the DOPT ACBLIB data set is used only by FM/IMS, no action is required now. If housekeeping procedures were in place for the DOPT ACBLIB they will no longer be required.

If a product other than FM/IMS also uses the DOPT ACBLIB data set, do the following steps:

1. Provide a new DOPT ACBLIB data set that only FM/IMS will use.
2. Change the DYNACB parameter on the FMN1POPI macro statement, to specify the name of the new DOPT ACBLIB data set.
3. Reassemble and link-edit the FM/IMS installation options module.
4. If the ACBs are managed by your installation (ACBMGMT=ACBLIB), concatenate the new DOPT ACBLIB data set with the ACBLIB DD in the IMS™ execution JCL (see Providing a DOPT ACBLIB data set on page 262).

XKEYDATAC

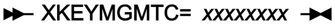
▶▶— XKEYDATAC= *xxxxxxxx* —▶◀

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.

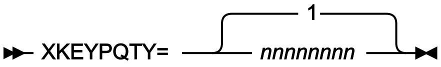
XKEYMGMTC



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.

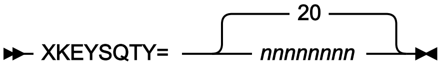
XKEYPQTY



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.

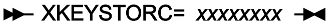
XKEYSQTY



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.

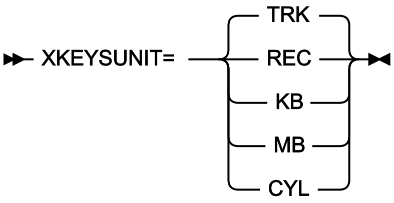
XKEYSTORC



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= *xxxxxxxx* ─▸◂

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= agn_id ─▶◀
```

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= ─┬────────────────┬─▶◀
           └─ AGNdescription ─┘
```

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= nnnn ─▶◀
```

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

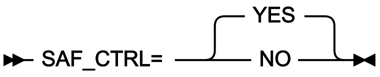
This section describes the File Manager options specified in parmlib member FMN0PARM. You can modify these File Manager options to suit your requirements.

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

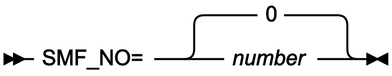


SAF_CTRL

Specifies whether File Manager should use SAF to control all aspects of File Manager audit logging. See Preparing to customize File Manager on page 13 and Customizing File Manager to write audit records to SMF on page 79 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.

SMF_NO



SMF_NO

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= *data_set_name* ▶◀

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= 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 z/OS® 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 73. 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 characters 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.

If you used Method 1 to add member FMN0PARM to the logical parmlib concatenation (see Defining the FMN0PARM member on page 85 for more information), be careful to avoid any conflicts between included member names and existing member names in 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

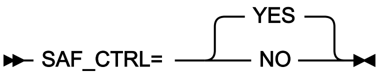
This section describes the FM/IMS options specified in parmlib member FMN1PARM. You can modify these FM/IMS options to suit your requirements.

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

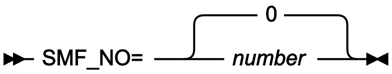


SAF_CTRL

Specifies whether FM/IMS should use SAF to control all aspects of FM/IMS audit logging. See Preparing to customize FM/IMS on page 243 and Customizing the File Manager audit facility for IMS component on page 295 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.

SMF_NO

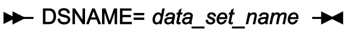


SMF_NO

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= *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 z/OS® 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 74. 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 characters 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.

If you used Method 1 to add member FMN1PARM to the logical parmlib concatenation (see Defining the FMN1PARM member on page 299 for more information), be careful to avoid any conflicts between included member names and existing member names in 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

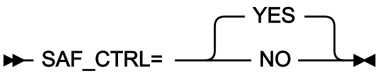
This section describes the FM/Db2 options specified in parmlib member FMN2PARM. You can modify these FM/Db2 options to suit your requirements.

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

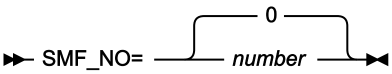


SAF_CTRL

Specifies whether FM/Db2 should use SAF to control all aspects of FM/Db2 audit logging. See Preparing to customize FM/Db2 on page 151 and SAF-controlled auditing for File Manager Db2 component on page 189 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.

SMF_NO



SMF_NO

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= *data_set_name* ─►◄

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= *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 z/OS® 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 75. 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 characters 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.

If you used Method 1 to add member FMN2PARM to the logical parmlib concatenation (see Defining the FMN2PARM member on page 190 for more information), be careful to avoid any conflicts between included member names and existing member names in 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

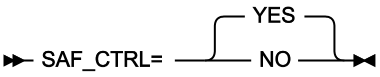
This section describes the FM/CICS options specified in parmlib member FMN3PARM. You can modify these FM/CICS options to suit your requirements.

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

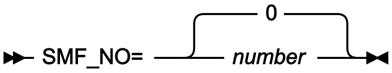


SAF_CTRL

Specifies whether FM/CICS should use SAF to control all aspects of FM/CICS audit logging. See Preparing to customize FM/CICS on page 324 and Customizing the File Manager audit facility for CICS component on page 349 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.

SMF_NO



SMF_NO

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= *data_set_name* ►◄

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= *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 z/OS® 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 76. 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 characters 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.

If you used Method 1 to add member FMN3PARM to the logical parmlib concatenation (see Defining the FMN3PARM member on page 353 for more information), be careful to avoid any conflicts between included member names and existing member names in 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 ADFz Common Components server

These topics describe how to customize the ADFzCC server.

Do this after you have installed File Manager if you plan to use either:

- FM/CICS
- File Manager Service Provider for z/OS® Connect
- File Manager Eclipse plug-in (in Eclipse, CICS Explorer®, or IBM® Developer for z/OS® (IDz))

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.

When a user runs the FM transaction using the FM/CICS interface or uses the Eclipse plug-in, a connection request will be received by the server from the client. The server will then verify the credentials of the client (user ID and password, or passphrase) and if valid, create a new FM process for that user. The user ID of the newly created process will be changed to that of the client and all further communication will occur between the client and the newly created FM process using TCP/IP. The server will then wait for another connection request from a new client.

ADFz Common Components server configuration

The default name of the ADFzCC server is IPVSRV. 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 CONFIG DD statement in the server procedure (IPVSRV). The data set defined, or concatenated, to this DD statement must be sequential, or a member from a PDS or PDSE.

Sample configuration data is provided in the FMN.SFMNSAM1 member FMNSAMCS. The sample configuration data needs customizing to your environment.

The File Manager sample IPV configuration member FMNSAMCS specifies SPAWN_PROGRAM=FMNCSEP. This program allows File Manager to run authorized and ensures that FM/CICS and the File Manager Eclipse plug-in can:

- Perform auditing
- Access tapes
- Use remote services

High level qualifiers for both File Manager and the ADFzCC server should be adjusted according to your site installation details. To simplify the creation of this configuration data, a sample REXX exec is supplied in the FMN.SFMNEXEC library, called FMNCSTCS. Running this exec from TSO prompts for the various data set names required, and produces a usable configuration for the ADFzCC server.

Refer to the *ADFz Common Components Customization Guide and User Guide* for any additional customization which may be required.

Additional security considerations

All the libraries listed in the SPAWN_STEPLIB statement 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.

Note: The user ID the IPVSRV procedure runs under must also have an OMVS RACF® segment defined.

See *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 plug-in 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. You should consider what settings are appropriate for your site and your use of ADFzCC server.

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
          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 useable.

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 562.

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 on page 568 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 565) 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.

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 consists of three parts:

- Record header
- Data item reference section
- Data section

Part 1 - record header

Table 77: Audit record: Header on page 570 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 77. Audit record: Header

Offsets		Type	Len	Name (Dim)	Description
Dec	Hex				
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)

Table 77. Audit record: Header

(continued)

Offsets		Type	Len	Name (Dim)	Description
Dec	Hex				
					'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
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 78: Audit record: Data item reference header on page 572 shows the information contained in the data item reference header.

Table 78. 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 78: Audit record: Data item reference header on page 572) specifies the number of elements in the array.

Table 79: Audit record: Data item reference section on page 572 shows the information contained in a data item reference array element.

Table 79. 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 82: Data items common to all audit records on page 574.
2	(2)	CHARACTER	1	LOG_DATATYPE	Data type code. See Table 83: Data item types common to all audit records on page 574.
3	(3)	CHARACTER	1	*	Reserved/filler
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 80: Audit record: Data section on page 573 shows the information contained in the data section:

Table 80. 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 81: Audit record: continued record on page 573.

Table 81. 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 77: Audit record: Header on page 570)
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 82. 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
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
Note: 1. This data item is not included in any FM/IMS audit records except for the Template record (Audit record type T).		

Table 83. 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 84. 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 85. 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.

Table 85. File Manager (base) 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.
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 86. 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 82: Data items common to all audit records on page 574

FM/Db2 audit records

This section describes the FM/Db2 audit records.

FM/Db2 audit data items

Table 87: Data items that appear in FM/Db2 audit log records on page 577 shows the data items that appear in FM/Db2 audit log records.

Table 87. Data items that appear in FM/Db2 audit log records

Data item identifier	Description	Usual length
14	Input/Output indicator	1
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 88: FM/Db2 audit record types on page 578 shows the FM/Db2 audit record types. Each record type has a single character code.

Table 88. 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® objects.
O	Original record or row. An 'O' record is typically written for an update (change) operation, and is followed by an 'R' record.
Q	Contains the text for an SQL statement. This is specific to FM/Db2.
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/Db2 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	Contains the text of a Db2® command. This is specific to FM/Db2.

Data items specific to FM/Db2 audit records

Table 89. Data items specific to FM/Db2 audit records

Audit Record Type	Data items
A	2031, 2032
D	2031, 2032
E	2041, 2042, 2043

Table 89. Data items specific to FM/Db2 audit records (continued)

Audit Record Type	Data items
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 82: Data items common to all audit records on page 574
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 90. 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

Table 90. Data items that appear in FM/IMS audit log records (continued)

Data item identifier	Description	Type	Length
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 90. 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 91. 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.

Note: Data items that are common to all FM/IMS audit records (4, 3, 92) are not listed.

Table 92. Data items specific to FM/IMS audit records

Audit Record Type	Data items
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.

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 93. 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

Table 93. Data items that appear in FM/CICS audit log records (continued)

Data item identifier	Description	Usual length
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 94. 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.
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 95. 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 82: Data items common to all audit records on page 574

Appendix H. Support resources

Use these resources to find product details, fixes, and support.

Search knowledge bases

- Download Program Directories from the IBM Publications Center:

Program Directory	Publication number
IBM Application Delivery Foundation for z/OS	GI13-4538
Application Delivery Foundation for z/OS Common Components	GI10-8969
Fault Analyzer	GI10-8967
File Manager	GI10-8968

- Get up-to-date details about installing, customizing, and using the product from IBM Documentation:

Product documentation	Publication number
Application Delivery Foundation for z/OS Common Components Customization Guide and User Guide	SC27-9050
Fault Analyzer User's Guide and Reference	SC27-9040
File Manager Customization Guide	SC27-9041
File Manager User’s Guide and Reference	SC27-9042
File Manager User’s Guide and Reference for DB2	SC27-9043
File Manager User’s Guide and Reference for IMS	SC27-9044
File Manager User’s Guide and Reference for CICS	SC27-9045

Get fixes

Find a fix for your product:

- Latest PTFs for ADFz Common Components
- Latest PTFs for Fault Analyzer for z/OS®
- Latest PTFs for File Manager for z/OS®
- Latest PTFs for z/OS® Debugger
- Latest PTFs for IBM Developer for z/OS Enterprise Edition
- Latest PTFs for Application Performance Analyzer for z/OS®

Collect diagnostic data

Before you contact Support, collect the data that can help to diagnose the problem. These documents explain what you'll need:

- Collect troubleshooting data for Application Performance Analyzer for z/OS®
- Mustgather: Read first for problems encountered with z/OS® Debugger
- Collect troubleshooting data for Fault Analyzer for z/OS®
- Collect troubleshooting data for File Manager for z/OS®
- Mustgather: Read first for problems encountered with Enterprise COBOL for z/OS®
- Mustgather: Read first for problems encountered with Enterprise PL/I for z/OS®

Be ready to answer these questions:

- What software versions are you running?
- Do you have logs, traces, and messages related to the problem?
- Can you re-create the problem? If so, how do you re-create the problem?
- Did you make hardware, operating system, or networking software changes?
- Do you have a workaround for the problem?

Contact Support

Open a case, chat with Support, or connect to resources and communities through https://www.ibm.com/mysupport.

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, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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 document at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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.

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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).
Portions of this code are derived from IBM Corp. and/or

HCL Ltd. sample programs.

Programming interface information

This documentation describes intended Programming Interfaces that allow the customer to write programs to obtain the services of File Manager.

Trademarks

IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM® or other companies. A current list of IBM® trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM® website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM®.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM®.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM® reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM®, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM® MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies,” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Programming interface information

This Customization Guide documents information that is **not** intended to be used as Programming Interfaces of File Manager.

Index