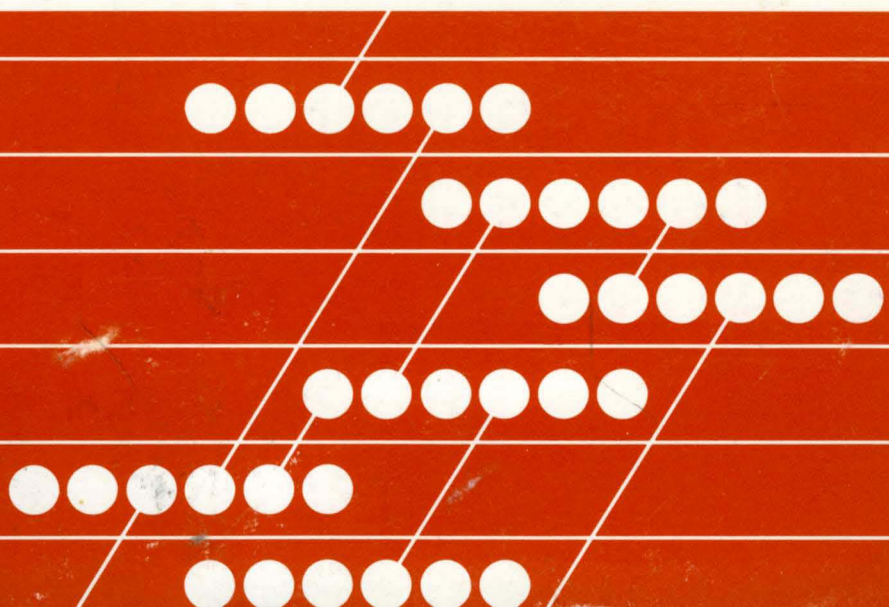


Interactive
System Productivity Facility
(ISPF) and ISPF/Program
Development Facility
Version 2 Release 2

Diagnosis

MVS



**Interactive
System Productivity Facility
(ISPF) and ISPF/Program
Development Facility
Version 2 Release 2**

Diagnosis

MVS

Publication Number
SC34-4020-1

File Number
S370/4300-39

Program Numbers
5665-317
5665-319

Second Edition (September 1985)

This edition applies to Version 2, Release 2, of the program products Interactive System Productivity Facility (ISPF) (5665-317), and Interactive System Productivity Facility/Program Development Facility (5665-319), and to all subsequent releases and modifications of both until otherwise indicated in new editions or Technical Newsletters. They are for use with OS/VS2 (MVS) Release 3.8 or later. In the Extended Architecture (XA) environment, they are for use with MVS/SP Version 2.

Changes are made periodically to the information herein. Therefore, before using this publication, consult the latest IBM System/370 and 4300 Processors Bibliography (GC20-0001) for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this document is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Dept. T46, P. O. Box 60000, Cary, NC 27511. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

PREFACE

This publication provides technical information to aid in the evaluation of problems related to the Interactive System Productivity Facility (ISPF) Program Product, Program Number 5665-319, and the ISPF/Program Development Facility (ISPF/PDF, or PDF) Program Product, Program Number 5665-317. It is intended for use by persons responsible for diagnosing and resolving program problems in an MVS environment.

Chapter 1 provides a brief introduction to ISPF and PDF capabilities and components.

Chapter 2 is a technical overview that describes:

- the ISPF task structure
- the ISPF and PDF data sets
- the member expansion and member parts list functions
- the PDF interface to the Library Management Facility (LMF)
- enqueue/dequeue logic and shared DASD operation

Chapter 3 discusses ISPF interfaces to other IBM program products.

Chapter 4 describes diagnostic aids, including test and trace modes, register usage conventions, and ABEND codes.

Related Documents

- ISPF and ISPF/PDF Version 2 for MVS General Information, GC34-4041 — Provides an overview and functional description of ISPF and PDF.
- ISPF Version 2 for MVS Dialog Management Services, SC34-4021 — Provides a detailed description of the ISPF services and related information required to develop an interactive application that runs under ISPF.
- ISPF Version 2 for MVS Dialog Management Services Examples, SC34-4022 — Supplements the above manual; provides examples of the use of various dialog management services.
- ISPF/PDF Version 2 for MVS Reference, SC34-4024 — Provides detailed information on how to use the ISPF Program Development Facility (ISPF/PDF) in MVS.
- ISPF/PDF Version 2 for MVS Services, SC34-4023 — Provides detailed information that aids in the development of dialogs that access ISPF

libraries and enable users to edit or browse data in ISPF libraries or sequential data sets.

- ISPF/PDF Version 2 for MVS Edit Macros, SC34-4018 — Provides detailed information necessary for the development of ISPF/PDF edit macro instructions.
- ISPF/PDF Version 2 for MVS Library Management, SC34-4025 — Provides detailed information about the Library Management Facility (LMF) services needed to administer and control ISPF/PDF libraries.
- ISPF and ISPF/PDF Version 2 for MVS Installation and Customization, SC34-4019 — Provides detailed information about the installation and customization of the ISPF and PDF Program Products.

CONTENTS

Chapter 1. Introduction	1
ISPF Components	3
ISPF Functional Components	3
ISPF Physical Components	5
PDF Components	5
PDF Functional Components	5
PDF Physical Components	6
Chapter 2. Technical Overview	7
ISPF Task Structure	7
ISPF and PDF Data Sets	10
Data Set Names	16
Default Names	16
Member Expansion and Member Parts List	17
The PDF Interface to the Library Management Facility	18
LMF Started Task	19
LMF Task Structure	20
LMF Data Set Usage	21
Enqueue/Dequeue Logic and Shared DASD Operation	23
Password-Protected Data Sets	23
Data Set Integrity Enqueue	23
Editor Member Name Enqueue	24
Chapter 3. ISPF and PDF Interfaces to Other Program Products	25
Hierarchical Storage Manager Interface	25
Program Control Facility Interface	25
PCF Command Authorization	25
Data Set Authorization	26
TSO/E Release 2 Interface	26
Session Manager Interface	26
TSO/TCAM Interface	26
TSO/VTAM Interface	28
Chapter 4. Diagnostic Aids	31
How to Take a Dump	32
Test and Trace Modes	32
Test Modes	32
Trace Modes	33
3270 Data Stream Trace	34
Register Linkage Conventions	34
ABEND Codes	34
Terminal Input/Output Error Codes	36
Log Files	37
The ISPF Log File	37
LMF Started Task Log Data Set	38
Resource Access Control Facility (RACF) Considerations	38
LMF Activity Log and Controls Review	39

Special Features of the ISPF Tutorial Panels	39
Hexadecimal (System) ABEND Codes	39
DAIR and Dynamic Allocation Return Codes	40
LMF Error Recovery Assistance	40
Potential Sources of LMF Errors	40
LMF Data Set Errors	40
LMF Distribution Table Errors	41
Index	43

FIGURES

1. ISPF Components	3
2. ISPF with PDF Components	5
3. ISPF Task Structure	7
4. ISPF Post/Wait Logic Connecting Control and Processing Tasks	9
5. ISPF and PDF Data Sets	11
6. PDF Interface to LMF Components	18
7. LMF Data Set Usage	21

CHAPTER 1. INTRODUCTION

The Interactive System Productivity Facility (ISPF) and the ISPF/Program Development Facility (ISPF/PDF, or PDF) are related program products for OS/VS2 MVS. You can use both of them to help you be more productive in developing interactive applications.

An interactive application that runs under ISPF is called an ISPF dialog. ISPF is a dialog manager that provides services to dialogs running under its control. Some of ISPF's services include invoking functions or displaying panels in response to a user's input, displaying error or informative messages and tutorial information, providing a split screen capability, and building and maintaining tables of user-supplied data.

ISPF Version 2 includes enhancements you can use to:

- Use the **national language support (NLS)** capability to run ISPF sessions in Danish, French, German, Hangeul, Italian, Japanese, and Spanish.
- **Write dialogs using APL2.** Dialog services can be executed from the APL2 workspace, and APL2 variables can be accessed the same way the dialog function pool is accessed.
- **Specify color and extended highlighting** in panel definitions, on terminals that have these capabilities.
- **Specify either full-screen or vertical and horizontal split-screen displays** for the 3290 terminal.
- **Use the)REINIT section**, a new section of the panel definition, to specify processing that should occur in conjunction with the redisplay of a panel.
- **Override attributes**, on a field or attribute character basis, that were specified in the)INIT,)REINIT, or)PROC sections of a panel.
- **Add statements and attributes** available on other types of panels to table display panels.
- **Sort** the rows of an existing table into a sequence specified by the dialog developer. Later additions to the table can be made so that this sequence is automatically maintained.
- **Use the Graphical Data Display Manager (GDDM) Program Product** (Program Number 5748-XXH) to format and display screens.
- **Display panels** containing one or more areas that are dynamically generated by the dialog performing the display.

- Display PF key information at the bottom of panels through use of PFSHOW.

PDF uses the advantages of interactive processing and display terminals to help you develop ISPF dialogs and other kinds of applications. For example, PDF contains a dialog test option that helps you develop and test the parts of your dialog as you create them. The edit option provides models for developing panels, messages, functions, and tables for a dialog. PDF also gives you an interface to the Document Composition Facility (SCRIPT/VS), Program Number 5748-XX9, and interfaces to standard language processors (compilers, assemblers, and a linkage editor). The online tutorial provides instruction and reference assistance.

PDF Version 2 enhancements include:

- **National language support (NLS)** for German and Japanese.
- An **edit macro language** that allows extension of the editing facilities by:
 - Accessing the data being edited
 - Accessing the current cursor position
 - Accessing edit modes and environmental information
 - Accessing external data from a library or data set
 - Invoking any dialog management service directly
- A **member parts list** that shows, for each specified source program module, the names of the modules it calls or includes, and the names of the modules that call or include it. This function can be invoked as either a foreground or background job.
- **Library access services** that allow access to ISPF libraries, sequential data sets, and partitioned data sets from user applications.

ISPF COMPONENTS

The functional and physical components of ISPF are described below.

ISPF Functional Components

Figure 1 is a functional overview of ISPF.

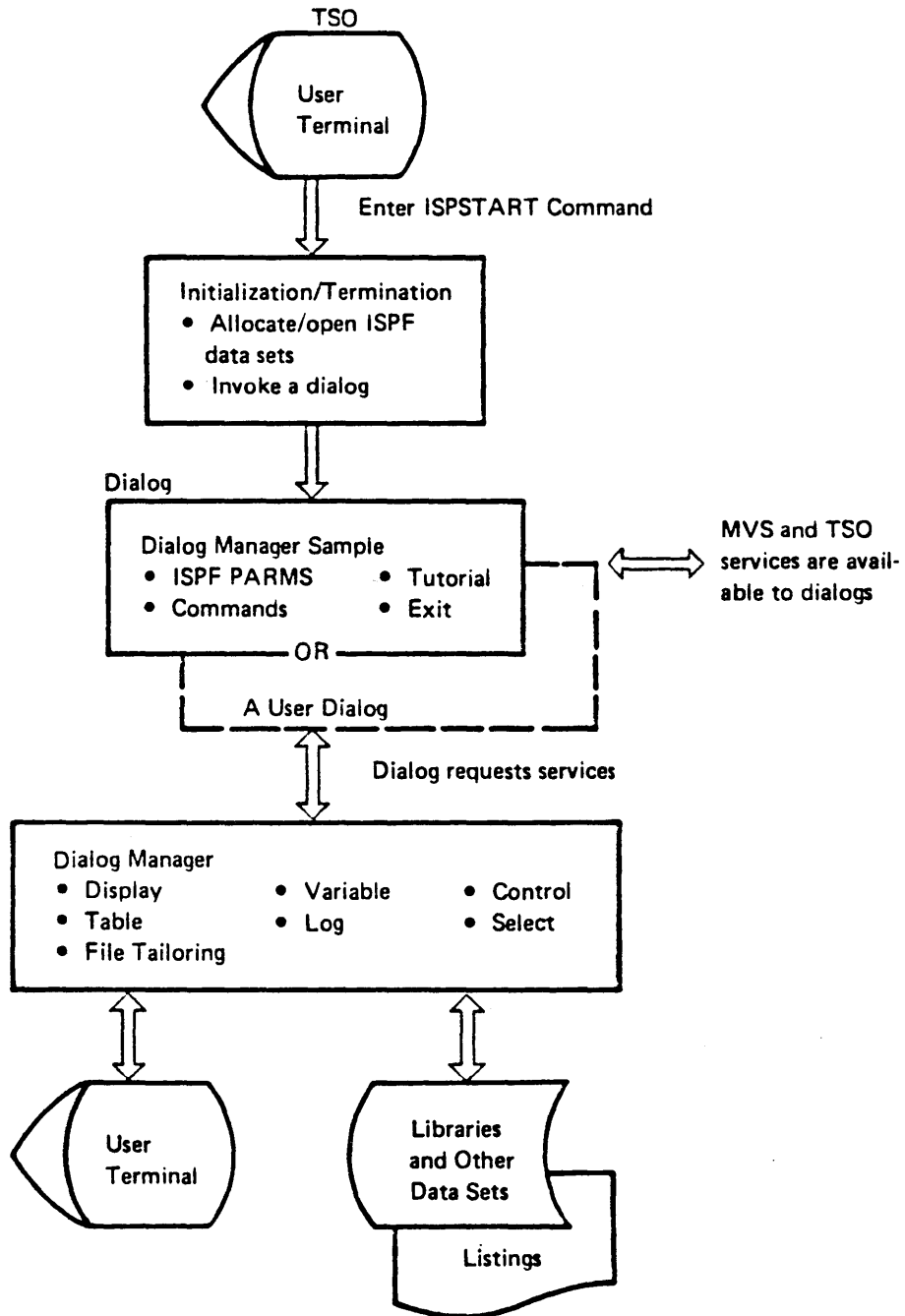


Figure 1. ISPF Components

To use ISPF the user must:

1. Log on to TSO.
2. Allocate the data sets used by ISPF.
3. Issue the ISPSTART command, and, optionally, specify the dialog or application to be invoked initially.

The ISPSTART command causes the ISPF environment to be initialized by setting up the necessary control blocks and assuring that certain files are allocated.

To specify the initial dialog or application, the user must enter the program (PGM), command (CMD), or PANEL parameter on the ISPSTART command.

4. Invoke a dialog. If the user did not invoke an application or dialog in step 3, the ISPF master application menu (ISP@MSTR) is invoked instead.

ISPF dialogs can be written as CLIST procedures, or as APL2 functions (using the APL2 Program Product, Program Number 5668-899). Dialogs can also be written in any of the high-level languages that can be used with ISPF (COBOL, FORTRAN, PASCAL, or PL/I). The initial dialog can be any one for which the user is authorized by MVS to use the data sets to be referenced by the dialog.

Dialogs can call other dialogs in a hierarchical manner.

Any dialog in control can call any of the services (for example, control, display, or table services) provided by ISPF. These services use data sets that must be allocated prior to the dialog's calling of the services. These data sets must contain all panels, procedures, messages, tables, and skeletons needed by the dialog during its execution.

In addition to using the services available in ISPF, an ISPF program dialog can use any of the MVS services. ISPF CLIST dialogs can include the normal CLIST statements, but cannot issue the ATTN or TERMIN statements. MVS authorized programs and TSO authorized commands can be executed only if TSO/E Release 2 is installed.

The user can split the screen at any time. When an additional screen is requested, a new invocation of the initially-specified dialog also begins. In split screen mode, the logical screens operate independently, and can run different dialogs or several invocations of the same dialog.

When the user terminates an ISPF session, ISPF releases all system resources it had allocated and returns control to TSO.

ISPF Physical Components

ISPF is comprised of one non-reenterable and several reenterable load modules. These load modules are built from object modules distributed on tape in SMP-installable format. The object modules must be link edited to create their respective load modules. Most of the load modules should be copied to the system link pack area (data set SYS1.LPALIB). See ISPF and ISPF/PDF for MVS Installation and Customization for more information.

PDF COMPONENTS

The following sections describe the functional and physical components of PDF.

PDF Functional Components

Figure 2 is a functional overview of ISPF and PDF.

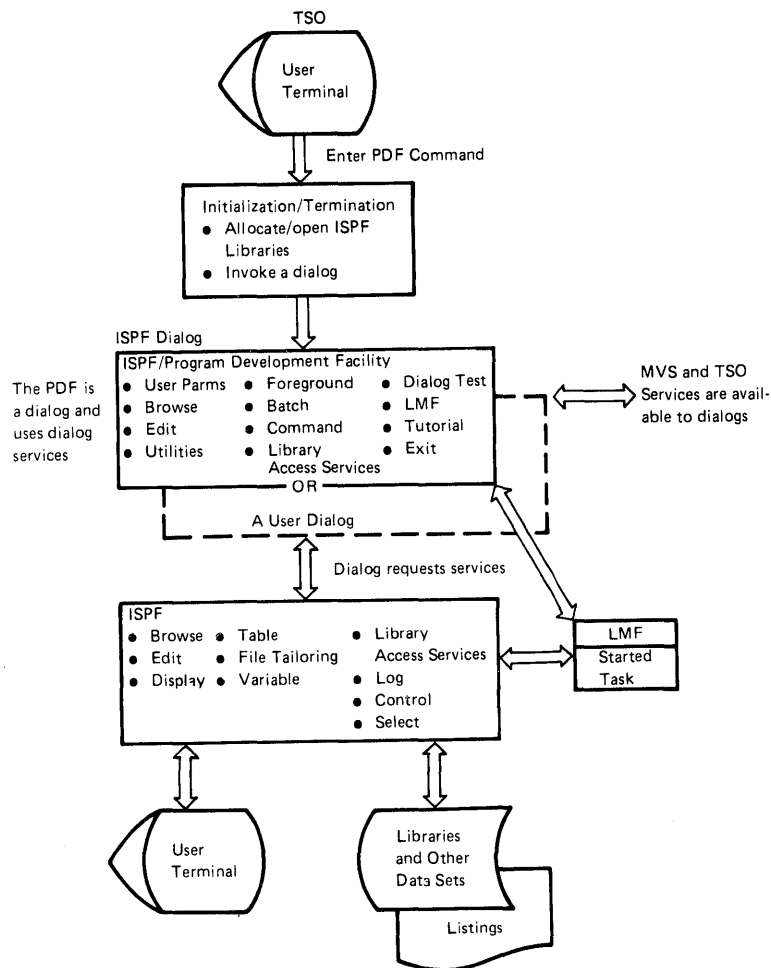


Figure 2. ISPF with PDF Components

To use PDF, the user must:

1. Log on to TSO.
2. Allocate the data sets used by PDF.
3. Issue the PDF command.

The PDF command causes the PDF environment to be initialized by setting up the necessary control blocks and assuring that certain files are allocated.

4. Invoke the PDF dialog.

The user can split the screen at any time. When an additional screen is requested, a new invocation of the initially-specified PDF dialog also begins. In split screen mode, the logical screens operate independently, and can run different dialogs or several invocations of the same dialog.

When the user terminates a PDF session, PDF releases all system resources it had allocated and returns control to TSO.

PDF Physical Components

PDF is comprised of one SVC and many reenterable load modules. These load modules are built from object modules distributed on tape in SMP-installable format. The object modules must be link edited to create their respective load modules. Certain load modules should be copied to the system link pack area (data set SYS1.LPALIB) in order to increase performance efficiency. See ISPF and ISPF/PDF for MVS Installation and Customization for more information.

CHAPTER 2. TECHNICAL OVERVIEW

This chapter describes:

- ISPF and PDF task structure
- ISPF and PDF data set usage
- Member expansion and member parts list capabilities
- The Library Management Facility (LMF) interface to PDF
- Enqueue/dequeue logic and shared DASD operation.

ISPF TASK STRUCTURE

The ISPF task structure is shown in Figure 3. The ISPF controller is attached by TSO whenever the user enters the ISPSTART command. The controller performs initialization and termination functions, and also handles display I/O (through either TCAM or VTAM) on behalf of the other ISPF programs. The controller attaches the driver for the logical screen and initiates a dialog.

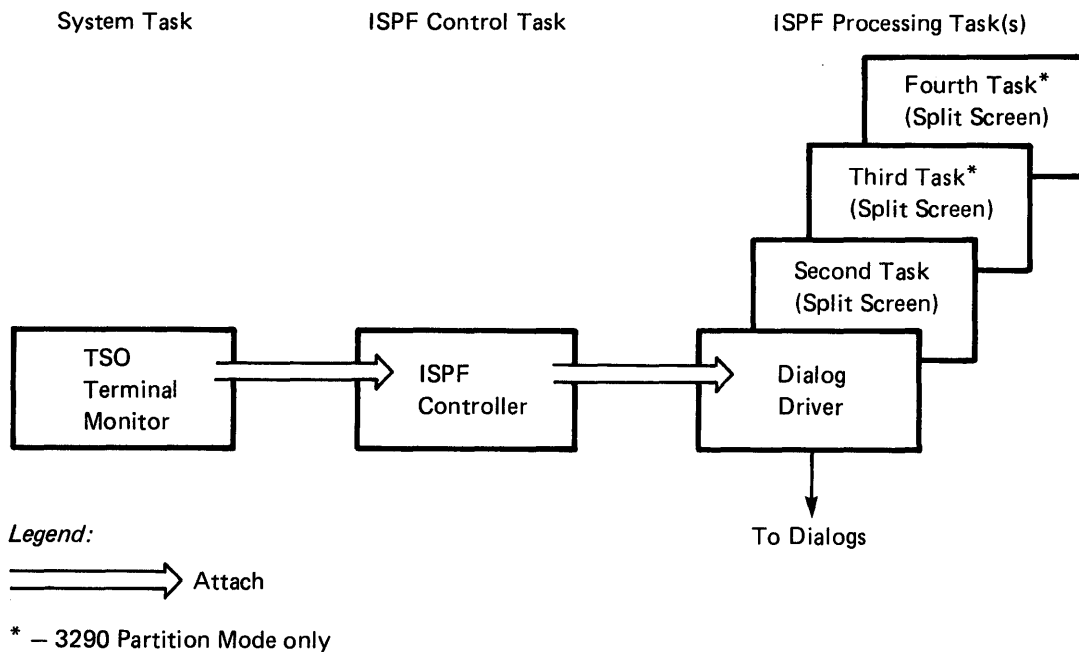


Figure 3. ISPF Task Structure

If the user enters split screen mode, the controller again attaches the driver to handle operations on the second logical screen. Thus, three tasks (up to five on a 3290 terminal) are active in split screen mode:

- ISPF control task
- ISPF processing task for screen 1
- ISPF processing task for screen 2
- ISPF 3290 processing task for screen 3
- ISPF 3290 processing task for screen 4.

The post/wait logic that connects the control task to each one of the processing tasks is shown in Figure 4. After the control task attaches a processing task, it waits for one of the following event control blocks (ECBs) to be posted:

- Display request ECB (ECB 1)
- Task completion ECB (ECB 2).

When a processing task wants the control task to perform display I/O or some other function, it posts the display request ECB (ECB 1) with one of three codes that specify the type of display request, and issues a wait on a process request ECB (ECB 3).

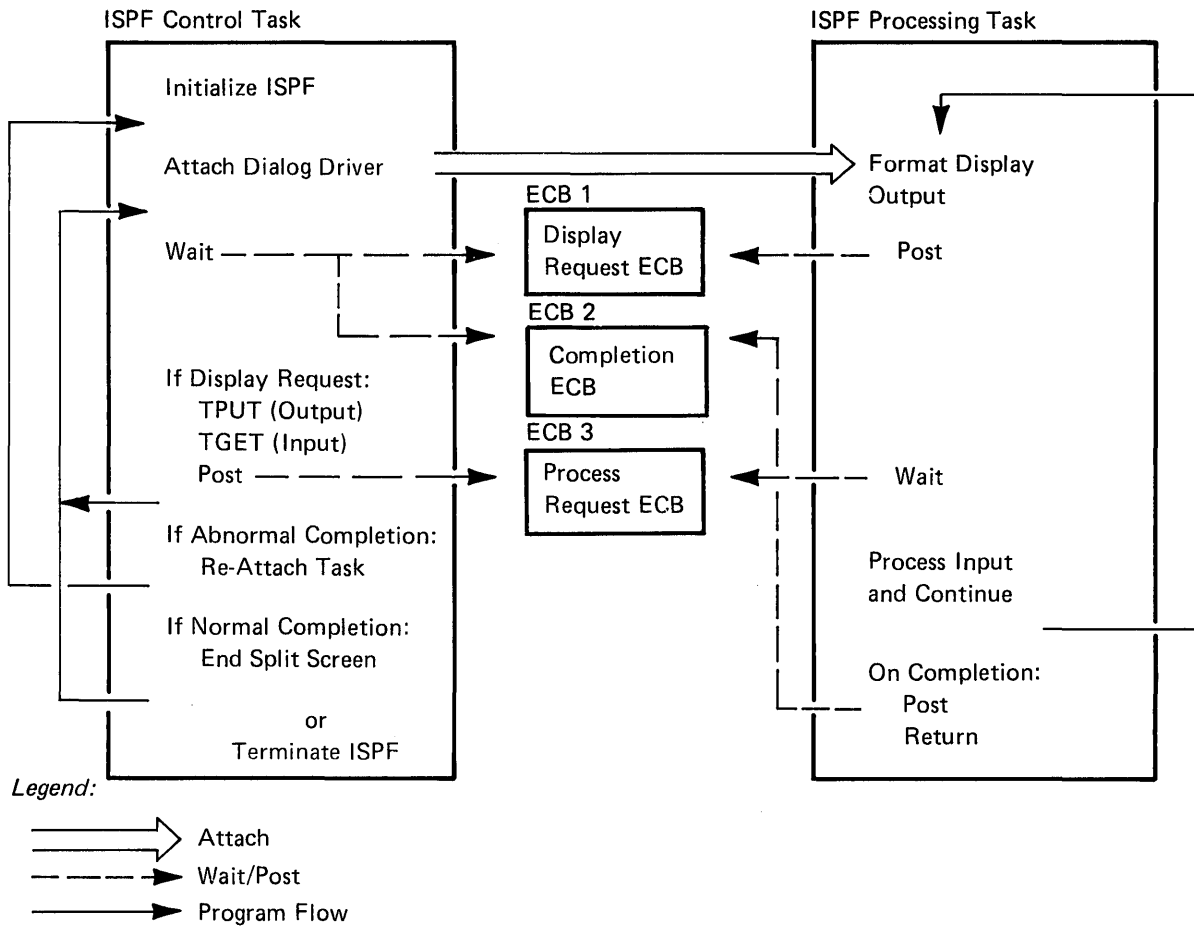


Figure 4. ISPF Post/Wait Logic Connecting Control and Processing Tasks

The three types of display requests are:

- Normal full-screen I/O - used when a processing task requires normal ISPF display I/O. In response to a display request, the control task:
 1. Writes information to the terminal (TPUT)
 2. Waits for a response from the user (TGET)
 3. Posts the process request ECB (ECB 3)
 4. Waits for a display request or completion of the processing task.

- Line I/O - used when a processing task enters standard TSO line I/O mode. In response to a line I/O request, the control task:
 1. Clears part of the screen and sets the line count

Note: In 3290 partition mode, in extended data stream mode, or when GDDM is being used, the entire screen is cleared.
 2. Posts the process request ECB (ECB 3)
 3. Waits for a display request or completion of the processing task.

- Common control interface - used when a processing task is to execute a common subroutine under the control task's task control block (TCB). In response to a common control interface request, the control task:
 1. Calls the common subroutine
 2. Posts the process request ECB (ECB 3)
 3. Waits for a display request or completion of the processing task.

When a processing task completes (either normally through an end function request, or abnormally because of an ABEND), the task completion ECB (ECB 2) is posted. For normal completion, the controller terminates split screen mode (if it was in effect), or terminates ISPF. For abnormal completion, the controller reattaches the processor task.

ISPF AND PDF DATA SETS

The functions of the data sets used by ISPF and PDF are shown in Figure 5. Some of the data sets are permanent data sets that are required for ISPF and PDF execution. Others are temporary data sets that are allocated if they are needed for a particular application. See "LMF Data Set Usage" on page 21 for a description of the PDF data sets used by the Library Management Facility (LMF).

INPUT DATA SETS	DESCRIPTION	OUTPUT DATA SETS
ISPMLIB—> (REQUIRED)	MESSAGE LIBRARY	
ISPMALT—>	DBCS MESSAGE LIBRARY	
ISPPLIB—> (REQUIRED)	PANEL LIBRARY	
ISPPALT—>	DBCS PANEL LIBRARY	
ISPSLIB—> (REQUIRED)	SKELETON LIBRARY	
ISPSALT—>	DBCS SKELETON LIBRARY	
ISPTLIB—> (REQUIRED)	TABLE LIBRARY (INPUT)	
ISPPROF—> (REQUIRED)	USER PROFILES	—> ISPPROF
	TABLE LIBRARY (OUTPUT)	—> ISPTABL
	FILE TAILORING OUTPUT	—> ISPFILE
	ISPF LOG FILE	—> ISPLOG
	ISPF LIST FILE	—> ISPLIST
ISPCTLO—> (TEMPORARY)	TEMPORARY EDIT SUBMIT DATA SET	—> ISPCTLO
ISPCTL1-4—> (TEMPORARY)	TEMPORARY CONTROL DATA SET (LOGICAL SCREEN 1-4)	—> ISPCTL1-4
ISPWRK1-4—> (TEMPORARY)	TEMPORARY WORK DATA SET (LOGICAL SCREEN 1-4)	—> ISPWRK1-4
ISPLST1-4—> (TEMPORARY)	TEMPORARY LISTING DATA SET (LOGICAL SCREEN 1-4)	—> ISPLST1-4
ISRExxxx—> (TEMPORARY)	TEMPORARY EDIT RECOVERY DATA SET	—> ISRExxxx

Figure 5. ISPF and PDF Data Sets

The data set names listed below are those used to refer to these data sets for documentation purposes. For example, "ISPCTL" is sometimes used generically to refer to ISPCTL1, ISPCTL2, ISPCTL3, or ISPCTL4 (the latter two being used with the 3290 terminal). Likewise, "ISPLIST" refers to ISPLST1, ISPLST2, ISPLST3, or ISPLST4. (ISPLST3 and ISPLST4 are used with the 3290 terminal.) Actual data set names and the file names to which they are assigned are discussed in "Data Set Names" on page 16.

The panel, message, skeleton, and table input libraries that are shipped with PDF must be concatenated ahead of the corresponding ISPF libraries. See ISPF and ISPF/PDF for MVS Installation and Customization.

The data sets have the following functions:

- ISPMLIB** This data set contains message definitions that are formatted and used primarily as error messages, information messages, or log messages.
- ISPMALT** This data set contains alternate message definitions to be used with DBCS. Users can allocate both ISPMALT and ISPMLIB at start time. See ISPF and ISPF/PDF for MVS Installation and Customization for a brief description of the alternate NLS libraries.
- ISPMUSR** This data set contains application-level message libraries that users can allocate before they invoke ISPF. This data set is used only when the LIBDEF service is issued with the DATASET keyword.
- ISPLLIB** This data set contains panel definitions that are formatted for display.
- ISPPALT** This data set contains alternate panel definitions to be used with DBCS. Users can allocate both ISPPALT and ISPLLIB at start time. See ISPF and ISPF/PDF for MVS Installation and Customization for a brief description of the alternate NLS libraries.
- ISPPUSR** This data set contains application-level panel libraries that users can allocate before they invoke ISPF. This data set is used only when the LIBDEF service is issued with the DATASET keyword.
- ISPSLIB** This data set contains file skeletons used by ISPF termination, file skeletons used by batch processing (option 5), and edit models, which are named ISRExxxx, where xxxx represents system-generated names for successive invocations of edit.
- ISPSALT** This data set contains alternate skeleton definitions to be used with DBCS. Users can allocate both ISPSALT and ISPSLIB at start time. See ISPF and ISPF/PDF for MVS

Installation and Customization for a brief description of the alternate NLS libraries.

ISPSUSR This data set contains application-level skeleton libraries that users can allocate before they invoke ISPF. This data set is used only when the LIBDEF service is issued with the DATASET keyword.

ISPTLIB This is the table input library. Each member contains an ISPF table.

Three tables are shipped with ISPF:

- ISPCMDS - the system command table
- ISPSPROF - the initial system profile (for new users)
- ISPPROF - the initial default application profile (for new users).

The ISRPROF table (the initial PDF application profile for new users) is shipped with PDF.

ISPTUSR This data set contains application-level table input libraries that users can allocate before they invoke ISPF. This data set is used only when the LIBDEF service is issued with the DATASET keyword.

ISPPROF There is a separate ISPPROF data set for each user. Each member of this data set contains a system profile, an application profile, or an edit profile for that user. If any one of these members is not found in the user's ISPPROF data set, the table input library (ISPTLIB) will be searched for that member. Hence, the initial profiles for new users come from ISPTLIB, but they will be stored in the user's ISPPROF data set for subsequent use.

Note: The ISPPROF data sets replace the ISPPARMS data set used with the System Productivity Facility (SPF), Program Number 5668-009, a predecessor program product. See ISPF and ISPF/PDF for MVS Installation and Customization for information on converting ISPPARMS data sets to the ISPPROF format.

ISPLLIB This data set is used as a task library when fetching load modules.

ISPLUSR This data set contains application-level LOADLIBs that users can allocate before they invoke ISPF. This data set is used only when the LIBDEF service is specified with the DATASET keyword.

ISPTABL This data set contains any output tables the user creates. ISPTABL **must** be allocated if the user invokes any PDF option that results in table creation (for example, option

3.8, the outlist utility, or option 3.9, the command table utility).

ISPTABU This data set contains application-level table output libraries that users can allocate before they invoke ISPF. This data set is used only when the LIBDEF service is specified with the DATASET keyword.

ISPLOG This data set contains log messages that keep track of significant actions during the session. The user can request the creation of a log data set by selecting option 0.2 from the primary option menu. Alternatively, setting the primary space allocation to 0 will eliminate the overhead of allocating, writing, and freeing the log data set.

You cannot use the print program provided with ISPF to print the contents of the ISPLOG data set if it contains DBCS data. You must make some changes to the ISPOPT2 panel and the ISPPFTS1 skeleton if you wish to print the information in ISPLOG. See ISPF and ISPF/PDF for MVS Installation and Customization for more information.

ISPLIST This data set contains formatted printer output requested by the user. A list data set will be allocated **only** if the user specifies one or more of the following options that write to ISPLIST:

- the PRINT command
- Edit automatic print (autolist mode)
- Print index listing (option 3.1 X)
- Print entire data set (option 3.1 L)
- Print member (option 3.1 P)
- Move/copy promote (option 3.3)
- Print data set lists (option 3.4)
- Print VTOC entries (option 3.7 P)
- Review listings (option 8.5)

You cannot use the print program provided with ISPF to print the contents of the ISPLIST data set if it contains DBCS data. You must make some changes to the ISPOPT2 panel and the ISPPFTS1 skeleton if you wish to print the information in ISPLIST. See ISPF and ISPF/PDF or MVS Installation and Customization for more information.

ISPFILU This data set contains file tailoring output.

ISPFILU This data set contains application-level file tailoring output libraries that the user can allocate before they invoke ISPF. This data set is used only when the LIBDEF service is specified with the DATASET keyword.

- ISPCTL0** This temporary data set is used with the edit submit command.
- ISPCTL1-4** These temporary data sets contain either control card images or utility output that has been formatted for display. They are allocated the first time they are required, and are deleted when a logical screen terminates. The following options and services use these data sets:
- Edit (by entering the edit command) for JCL to be submitted
 - Batch (option 5) for JCL to be submitted
 - Hardcopy utility (option 3.6) for JCL to be submitted
 - Catalog utility (option 3.4) for IDCAMS control cards
 - VTOC utility (option 3.7) for data to be displayed
 - PDF termination for JCL to be submitted
 - File tailoring processing when a temporary sequential file is used
- ISPWRK1-4** These data sets contain temporary output data used during the file tailoring process. They are allocated the first time they are required, and are deleted when either the file tailoring process or a logical screen terminates.
- ISPLST1-4** These temporary data sets contain listings that have been generated by OS utilities invoked by PDF. The listings are displayed on the screen. They are allocated the first time they are required, and deleted when a logical screen terminates. The compress utility (option 3.1) uses these data sets.
- ISRExxxx** These temporary data sets contain data that is saved during an edit session in recovery mode. They are allocated the first time data is changed in an edit session, and they remain allocated until the user terminates PDF. The data sets allow recovery processing in split-screen mode.

DATA SET NAMES

All permanent data sets must be allocated by the user before ISPF or PDF is initialized. Temporary data sets are allocated dynamically as needed by ISPF. The user must allocate optional data sets before their first use.

Default Names

The default names used by ISPF and PDF for the various data sets are:

These are the fully-qualified data set names:

ISPMLIB	-	'ISP.V2R2M0.ISPMLIB'
ISPMALT	-	'ISP.V2R2M0.ISPMKLB'
ISPPLIB	-	'ISP.V2R2M0.ISPPLIB'
ISPPALT	-	'ISP.V2R2M0.ISPPKLB'
ISPSLIB	-	'ISP.V2R2M0.ISPSLIB'
ISPSALT	-	'ISP.V2R2M0.ISPSKLB'
ISPPROF	-	(assigned by the installation or user)
ISPTABL	-	(assigned by the installation or user)
ISPFIL	-	(assigned by the installation or user)

These names are right-appended to the TSO user prefix and the userid (if the userid is different from the prefix).

ISPFLOG	-	SPFLOG1.LIST	(1)
ISPF1LIST	-	SPF1.LIST	(1)
ISPFCTL0-4	-	SPFTEMP1.CNTL	(2)
ISPFWRK1-4	-	SPFTEMP1.WORK	(2)
ISPFPLST1-4	-	SPFTEMP1.LIST	(2)
ISRExxxx	-	ISRExxxx.BACKUP	(3)

Notes:

1. These data sets can have numbers other than "1." For example:

```
SPFLOG4.LIST
SPF8.LIST
```

2. These data sets will have numbers "1" for logical screen 1, "2" for logical screen 2, and so on. For example:

```
SPFTEMP2.CNTL
```

3. These data sets will have system-generated names for successive invocations of edit. For example:

```
ISRE0003.BACKUP
```

MEMBER EXPANSION AND MEMBER PARTS LIST

For input to batch jobs, which normally require sequential input, a single member can be selected from a set of concatenated libraries and copied to a temporary sequential data set. If the selected member is packed, it can be unpacked along with all included members. A member parts list can also be generated. The following functions permit the use of concatenated ISPF libraries for source code:

ISRSCAN This module is used to select a member from the input library and copy it to a temporary sequential data set if none of the input data is packed.

ISRLEMX This module is used either to unpack and expand source code or to generate the member parts list. You must use this module to select your input member if your input member or any of its included members are packed. ISRLEMX starts with the selected member and copies it to the output data set. Any included members are expanded beginning at the line where the include was found. This function uses the common keywords COPY, %INCLUDE, @INCLUDE, and .im to trigger expansion. In addition, users can define triggers at installation time. These user-defined triggers must meet the following criteria:

- They must contain no more than 20 characters.
- They must start in a user-specified column.

Expansion will continue if an included member cannot be found in the input library, because the expansion module assumes that the compiler will get the member from another library. If a severe error is encountered, ISRLEMX will end with a return code of 20.

ISRLEMX also generates the member parts list, which is a list of all members called or included by a module, as well as all referenced modules that called or included it. The INCLUDE keywords are the same as those used for member expansion. The CALL keyword is used for called modules.

Input to the member parts list can be a single member, a truncated member name (ISRL*), or an asterisk (*), which indicates all members. These input members are referred to as primary members. In order to reduce the size of the listings for jobs that specify "*" as input, ISRLEMX can limit the search for primary members to one or more of the input data sets. To locate members specified by the CALL or INCLUDE keywords, however, ISRLEMX searches all of the input data sets.

THE PDF INTERFACE TO THE LIBRARY MANAGEMENT FACILITY

This section describes the PDF interface to the Library Management Facility. The LMF is an optional extension of the ISPF library concepts and options that allows users to control the contents of their libraries and to manage their development processes more efficiently.

Note: If you do not wish to use the LMF interface, you can bypass it by not allocating the ddname ISRCFIL when you establish the PDF session. However, if you bypass LMF, you will not be able to lock members or promote any members to controlled libraries.

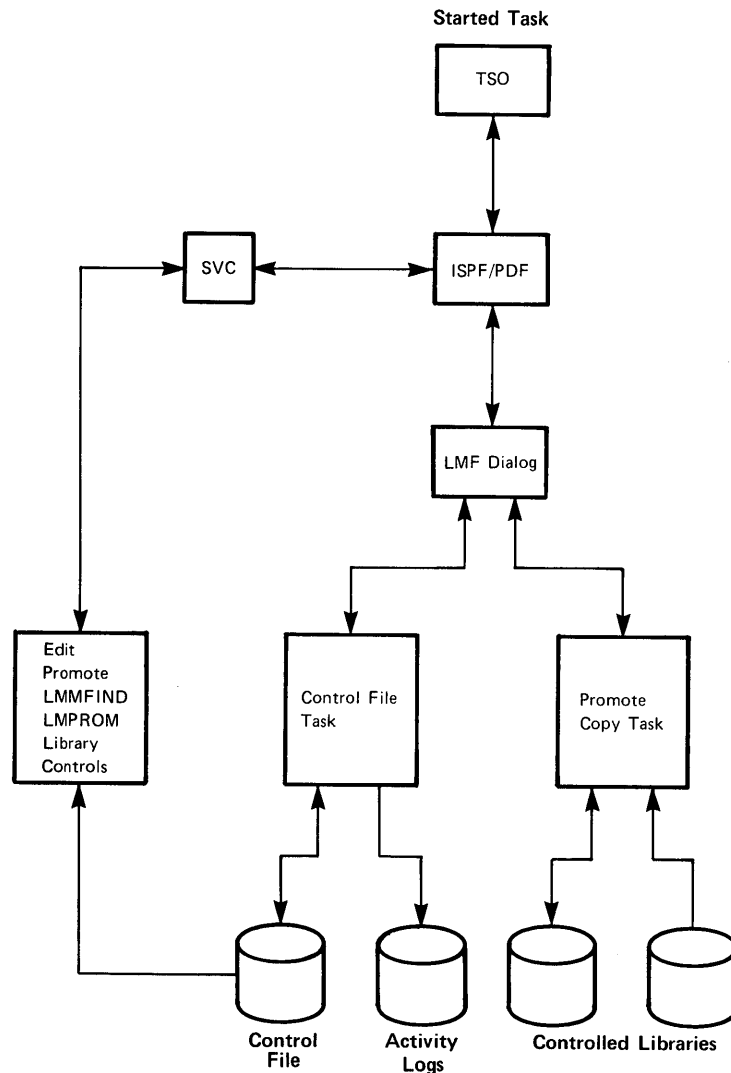


Figure 6. PDF Interface to LMF Components

The PDF interface to LMF contains:

1. Panels and programs that reside in the user's address space. In order to create, update, and review library controls, the control file must be allocated before the first invocation of LMF, since these programs perform initial verification of LMF controls.
2. A started task running PDF in batch mode. The invocation of the started task should become part of the normal IPL procedure.
3. A system extended SVC (number 109-36, module IGX00036) to transfer information between the two address spaces.

Note: If your system is below MVS/SP 1.2, you must install a user SVC in order to make use of the LMF interface. See ISPF and ISPF/PDF for MVS Installation and Customization for more information. Otherwise, use the system extended SVC described above.

The started task executes the TSO Terminal Monitor Program, and then issues the ISPSTART command to invoke the LMF dialog. If a problem occurs during dialog initialization, a message can either be issued to the operator's console or be written to ISPLLOG (the PDF message data set), or both. The ISPLLOG data set is open for the duration of the started task and can only be viewed after the started task has terminated.

LMF Started Task

The LMF started task is a specialized dialog written to run in batch mode under PDF. This dialog queues requests and then processes them in one of two ways:

1. **Control file requests** are handled single stream. These requests cause members in the control file to be updated. If the request is to lock a member, activity logging is performed as required.
2. **Promotion requests** have four steps, as follows:
 - a. **Verification step** - This control file request checks to see if the promotion request is valid, and sets a promotion-in-progress flag in the control file for the affected member.
 - b. **Copy step** - This separate attached task runs under the dialog. It does not make use of the control file, and thus allows more than one promotion to be performed at the same time.
 - c. **Unlock step** - This control file request sets the member status to promotion complete. Activity logging, if requested, will be performed in this step.
 - d. **Return step** - This step deletes promoted members when control is returned to the user's address space.

If a system failure occurs while an LMF request is being processed, simply re-execute the request, so that any pending operation will be resolved.

LMF Task Structure

The task structure in the LMF started task is the same as that of a single-display ISPF session, with the addition of the following characteristics:

1. Since this is a started task running in batch mode, there is no terminal I/O.
2. The LMF dialog issues an EVENTS macro to wait for the SVC to post a request.
3. After processing the request, the dialog issues a cross-memory post to return data and control to the user.
4. If the user requests a promotion, the dialog first verifies that the user is authorized to make the request, and then attaches a task to perform the promotion.
5. After the control task returns from the promotion, it performs cleanup and logging functions, and deletes the controlled member from which the promotion was made.
6. After the promotion request copy(s) have been completed, the dialog resumes processing at the EVENTS macro.

LMF Data Set Usage

Figure 7 shows the functions of the LMF data sets. Some of the data sets are permanent data sets that are required to execute LMF functions. Others are temporary data sets that are allocated when they are needed for a particular application.

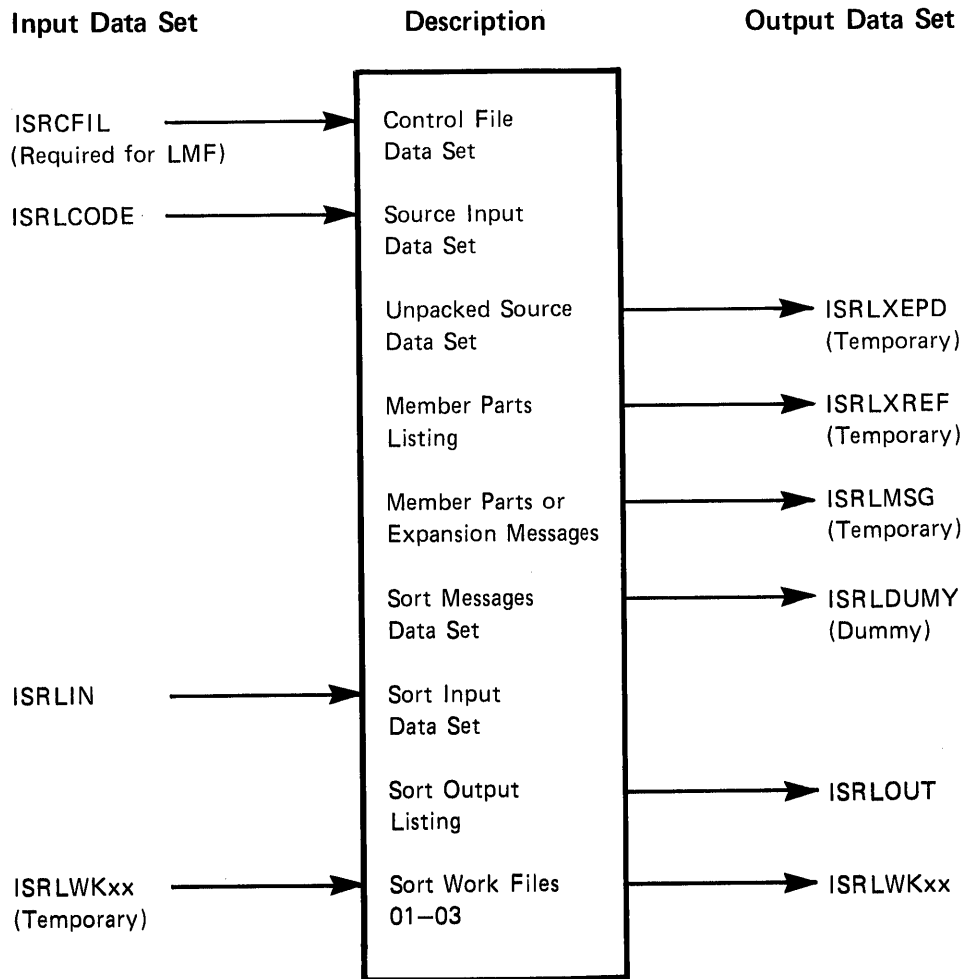


Figure 7. LMF Data Set Usage

The data sets have the following functions:

- ISRCFIL** This required data set contains the LMF control information. It must be RACF-protected for read only and allocated with DISP=SHR.
- ISRLCODE** This data set contains either the source code for member parts list (options 4.12 or 5.12) or the source code for any compiles run with PACKED DATA=YES. It can be a sequential data set or an ISPF library. If the data set is submitted as part of a batch job, it can be composed of up to nine concatenated partitioned data sets.
- ISRLEXPD** This temporary data set contains the unpacked source code for compiles for which PACKED DATA=YES was specified. All included modules will be expanded inline at the point where the include was found.
- ISRLXREF** This data set contains the report generated by the member parts list function. For batch jobs, the data set should be allocated to SYSOUT, and for foreground jobs, it should be a temporary data set.
- ISRLMSG** This file contains error messages for batch jobs running with either member parts list or member expansion. For foreground jobs, the messages are written to the output data set (ISRLEXRF or ISRLEXPD).
- LMF reviews (option 8.5) and member parts list (options 4.12 or 5.12) use the data sets listed below to sort listing data:
- ISRLDUMY** This file is the Sort/Merge message data set. Under normal circumstances, these messages do not need to be reviewed, and so the file is allocated as DD DUMMY. However, if the message "SORT FAILED" is returned from LMF reviews, member parts list, or member expansion, this file can be allocated to the terminal or to SYSOUT to provide access to the Sort/Merge messages.
- ISRLIN** This data set can be either a data set specified by the user or a temporary data set. It contains information needed to generate the requested listing, and is the input to Sort/Merge.
- ISRLOUT** This data set can be either a data set specified by the user or a temporary data set. It contains sorted information needed to generate the requested listing.
- ISRLWKxx** These temporary data sets (ISRLWK01, ISRLWK02, and ISRLWK03) are dynamically allocated for Sort/Merge if the sort cannot be performed in storage.

ENQUEUE/DEQUEUE LOGIC AND SHARED DASD OPERATION

This section describes enqueue/dequeue logic and shared DASD operation.

Password-Protected Data Sets

Password-protected data sets should not be placed on shared DASD volumes. PDF users who update password-protected data sets that reside on shared DASD should fill in the password field on data entry panels in order to avoid being prompted for a password by the open processing routine, since the volume is reserved during open processing. In addition, it is not possible to promote into data sets with write-protected passwords, or to promote from data sets with read-protected passwords.

Data Set Integrity Enqueue

The PDF editor, library utility, move/copy utility, and reset statistics utility all allocate partitioned data sets for "shared" use even though the data sets can be modified (by adding, replacing, renaming, or deleting members). This shared allocation allows more than one user to modify other members in the same data set without tying up the entire data set.

In order to maintain data set integrity, either a reserve or a DEQ macro is issued at the appropriate time. The reserve macro prevents shared DASD conflicts in accessing the volume, and also causes an enqueue to be issued. This enqueue is unconditional; if the resource is in use, the processing task is placed in the wait state until the resource is freed. The DEQ macro dequeues the resource and also causes the volume to be released, if it is a shared volume.

The sequence of events for referencing a partitioned data set member without causing an integrity exposure is:

1. Issue SPFEDIT reserve
2. If updating a load module, issue link editor reserve or enqueue
 - a. Open data set for output
 - b. Write the member (for add or replace member)
 - c. Issue STOW macro (add, replace, rename, or delete)
 - d. Close the output DCB.
3. Issue SPFEDIT DEQ
4. If updating a load module, issue link editor dequeue.

The reserve macro parameters for the SPFEDIT enqueue are:

```
Type = Systems reserve
Qname = 'SPFEDIT '
Rname = Data set name right-padded with blanks
Rname length = 44
```

The link editor enqueue is issued only for record format U data sets.

- If the volume is a shared DASD volume:

```
Type = Systems reserve
Qname = 'SYSIEWLP'
Rname = Data set name right-padded with blanks
Rname length = 44
```

- If the volume is not a shared DASD volume:

```
Type = system ENQ
Qname = 'SYSIEWLP'
Rname = Data set name right-padded with blanks
Rname length = 44
```

Editor Member Name Enqueue

When a member of a partitioned data set has been selected for editing, the PDF editor issues another ENQ that determines whether another user is currently editing the same member. This ENQ is conditional. If the member is being edited by another user, a message to that effect is displayed on the screen. Otherwise, the member is fetched for editing. The corresponding DEQ is issued when the user ends (or cancels) the edit session.

For a conditional enqueue, the ENQ/DEQ parameters are:

```
Type = System ENQ
Qname = 'SPFEDIT '
Rname (first 44 characters) = fully qualified data set name,
    right-padded with blanks
Rname (next 8 characters) = member name (blanks if RECFM=PS)
Rname length = 52
```

CHAPTER 3. ISPF AND PDF INTERFACES TO OTHER PROGRAM PRODUCTS

This chapter describes ISPF interfaces to the following program products:

- Hierarchical Storage Manager
- Program Control Facility
- TSO/E Release 2
- Session Manager
- TSO/TCAM
- TSO/VTAM.

HIERARCHICAL STORAGE MANAGER INTERFACE

When ISPF allocates a cataloged data set, it issues a LOCATE system service. If the data set is under the control of the Hierarchical Storage Manager (HSM) Program Product (Program Number 5740-XRB), the volume "MIGRAT" is normally returned. In order to cause a "recall" and get the real volume serial, ISPF sets the third bit in the third byte (CAMPOTN3) of the parameter list (CAMLIST) that is passed to LOCATE. ISPF sets this bit for all ISPF-issued locates, whether or not HSM is installed.

PROGRAM CONTROL FACILITY INTERFACE

This section describes the ISPF interface to the Program Control Facility (PCF) Program Product (Program Number 5798-CLW).

PCF Command Authorization

ISPF uses PCF command authorization by means of a PCF interface that uses IKJSCAN. Before calling IKJSCAN, ISPF places 'SPF' in the primary command field of the Environmental Control Table (ECT). This action indicates to PCF that the command being scanned is not a subcommand, and that command authorization should be performed. If IKJSCAN returns with RC=52 (indicating that the user is not authorized to issue the command), ISPF displays an error message.

Data Set Authorization

PCF data set read-only authorization is based on the assumption that programs that are to update a data set will specify a disposition of OLD when first allocating the data sets. However, the PDF editor, library utility, move/copy utility, and reset statistics utility all allocate partitioned data sets with a disposition of SHARE and then issue a RESERVE to serialize the updating of the members.

To inform PCF that a SHARE allocation is being performed for updating purposes, PDF sets the first byte of the ECTSCMD field in the ECT to '01'X. PCF tests for this flag and performs the authorization check as though the data set were allocated with a disposition of OLD.

TSO/E RELEASE 2 INTERFACE

ISPF uses two interfaces to communicate with TSO/E:

1. The first interface assists in the invocation of authorized commands and authorized programs under ISPF by checking for the presence of the command name or program name in either the TSO authorized command list (IKJEFTE2) or the authorized program list (IKJEFTE8). If the name is found in either list, the TSO authorized program interface module, IJKEFASR, is called to invoke the authorized command or program.
2. The second TSO interface provides access to all CLIST variables. The TSO IKJCT440 routine is called to create and update variables, retrieve values of variables, and query variable names.

SESSION MANAGER INTERFACE

Modules ISPSC93 and ISPSC94 are ISPF exit routines for SVC 93 (IGC0009C) and SVC 94 (IGC0009D). The Session Manager exit routines IKTTMPX1 and IKTTMPX2 reference these modules. The ISPF exits provide a better interface between ISPF and the Session Manager, but they are not required for ISPF operation with the Session Manager Program Product (Program Number 5740-XE2). The ISPF exits are not installed if the Session Manager is not installed.

Refer to ISPF and ISPF/PDF for MVS Installation and Customization for information on the installation of these modules.

TSO/TCAM INTERFACE

This section describes modifications to the TSO/TCAM message handler that are required to run ISPF on a TSO/TCAM system. Refer to ISPF and ISPF/PDF for MVS Installation and Customization for more information about these modifications.

During ISPF operation, certain non-ISPF-generated messages, such as broadcast messages from the system operator or from other TSO users, can be sent to the terminal. TCAM queues these messages and displays them when the user presses any one of the interrupt keys (the ENTER key or one of the PF keys). That same interrupt, however, generally causes ISPF to generate full-screen output. Unless that output is intercepted, it is sent to the terminal, where it causes an immediate overlay of the broadcast message.

The ISPF modifications to the standard TSO/TCAM message handler are designed to correct this problem. When ISPF full-screen output is followed by non-ISPF line output, the following sequence of events occurs:

1. The line message is displayed with high intensity, and the audible alarm (if one has been installed) is sounded.
2. A simulated attention (SIMATTN) condition is generated, causing three asterisks (***) to be displayed on the bottom line of the screen. The SIMATTN condition prevents further output from being sent to the terminal until the user presses an interrupt key.
3. When ISPF full-screen output follows the line message, a simulated PA2 interrupt is generated, causing ISPF to re-display the entire screen contents.

Note: Since ISPF does not know about the line message, the screen image can be garbled until the re-display occurs.

The FULLSCR EXPFLS=YES message handler macro provides the program logic for these functions. This macro generates code that distinguishes ISPF-generated output from non-ISPF-generated output by analyzing the signal string that begins every ISPF-generated full-screen output message.

There are two types of ISPF signal strings:

1. ISPF full-screen signal string:

HEX '11 5D7F 11 xxxx

2. ISPF exit full-screen signal string:

HEX '11 5D7E 11 xxxx

Where:

HEX 11 is 3270 set buffer address (SBA) orders
HEX 5D7F represents screen location 1919 (decimal)
HEX 5D7E represents screen location 1918 (decimal)
xxxx represents the actual buffer address desired
by ISPF.

Note: The occurrence of two SBAs in a row is a valid but meaningless sequence. Therefore, it is assumed that non-ISPF-generated messages will contain neither of these signal strings.

In addition to using these signal strings to distinguish ISPF-generated output, the message handler macros reset the TSO/TCAM line count whenever they encounter a signal string, as follows:

- For the full-screen signal string, the line count is set to 2, causing any subsequent line message (such as a broadcast) to be displayed on line 3 of the screen.
- For the exit full-screen signal string, the line count is set to the position indicated by the second SBA (xxxx). This setting allows ISPF to control the starting location for subsequent line messages, such as a 'SUBMIT' command confirmation message or the beginning of a foreground processing session.

The FULLSCR message handler macros use the IEDQFSCR option byte to keep track of the message sequence. This byte has the following format:

IEDQFSCR BITS	Meaning
-----	-----
...1	reserved
.... 1...	EXPFLS re-display flag
.... .1..	EXPFLS broadcast flag
.... ..1.	EXPFLS full-screen flag
.... ...1	reserved

The option byte must be initialized to zero in the assembly of the TCAM message control program (MCP).

TSO/VTAM INTERFACE

This section describes TSO/VTAM system services that are used to run ISPF on a TSO/VTAM system. Refer to ISPF and ISPF/PDF for MVS Installation and Customization for information on the modifications necessary for running ISPF with TSO/VTAM.

TSO/VTAM generally handles non-ISPF-generated messages, which are described in "TSO/TCAM Interface" on page 26. In contrast to TSO/TCAM, TSO/VTAM requires no terminal access method modifications. However, ISPF must, when entering or leaving full-screen mode, send an appropriate signal to TSO/VTAM.

Module ISPMIN issues a "STFSMODE ON,INITIAL=YES" system service before the first ISPF full-screen TPUT. This system service is also used to determine the terminal access method. A nonzero return code indicates that ISPF is running under TSO/TCAM.

When an ISPF module requires entry into normal TSO line I/O mode, control first passes to module ISPMLI. ISPMLI issues the "STLINENO MODE=OFF,LINENO=xx" system service to set the line number. The requesting module can then issue line I/O TPUTS and TGETS to the terminal, or can attach a program that will issue the commands. To return to ISPF full-screen operations, the module requests full-screen output by calling module ISPCDI. Module ISPMCO assumes control and issues a full-screen TPUT, causing TSO/VTAM to generate a simulated PA2 interrupt. ISPMCO handles a PA2 interrupt by first issuing a "STFSMODE ON" system service and then re-displaying the complete screen.

When the ISPF session is terminated, module ISPPFT issues a "STFSMODE OFF" system service.

A major operating difference between TSO/VTAM and TSO/TCAM becomes apparent when a non-ISPF-generated message is sent to the display. In this situation, TSO/VTAM clears the screen before displaying the message if the full-screen mode has not been turned off by a "STFSMODE" or "STLINENO" system service.

CHAPTER 4. DIAGNOSTIC AIDS

This chapter provides information to assist you in diagnosing problems within ISPF and ISPF/PDF. The following topics are covered:

- How to take a dump
- Test and trace modes
- 3270 data stream trace
- Register linkage conventions
- ABEND codes
- Terminal input/output error codes
- Log files
- Special features of the ISPF tutorial panels
- Potential sources of LMF errors

HOW TO TAKE A DUMP

If an ISPF subtask ABENDs, a dump can be taken by pressing ENTER after the ABEND message appears, provided that a SYSUDUMP, SYSDUMP, or SYSABEND data set has been allocated, and ISPF has been started under test mode.

Two dumps are generated: one for the ISPF subtask and one for the main task. If you have specified SYSDUMP (machine-readable to DASD), you must also specify DISP=MOD in order to keep the second dump (the main task) from overwriting the first one (the ISPF subtask).

If you direct the output from SYSUDUMP or SYSABEND to a data set, you must specify DISP=MOD.

TEST AND TRACE MODES

ISPF's test and trace modes provide special processing actions for debugging a dialog.

If you have installed PDF, consider using the dialog test option of that facility instead of the testing modes described here. See ISPF/PDF for MVS Reference for more information.

When you are in test mode and you want to find a particular message ID, look at the bottom line of the tutorial panel. This line displays the current panel name, the previous panel name, and the previous message ID.

To control the operational mode while testing a dialog, you can specify any one of four mutually-exclusive keyword parameters on the ISPSTART or PDF commands:

- TEST - Test mode
- TESTX - Extended test mode
- TRACE - Trace mode
- TRACEX - Extended trace mode

Test Modes

In **test mode**, ISPF operates differently from normal mode in the following ways:

1. Panel and message definitions are re-fetched from the panel and message libraries whenever a panel name or message id is specified in an ISPF service. (In normal mode, the most-recently-accessed panel definitions are retained in virtual storage and a table of TTRs returned from BLDL macros is kept in virtual storage for frequently-used message, panel, skeleton, and table members.) If

you have modified the panel or message library, use test mode to ensure that you are accessing the latest version of each panel or message during your test runs.

Note: A new extent on a DASD may be caused by link editing a module, or by using an editor to modify a panel, message, or skeleton. If a new extent is caused, you can access the modification only by first terminating and then re-invoking ISPF.

2. Tutorial panels are displayed with the current panel name, the previous panel name, and the previous message id on the bottom line of the display screen. This information helps you to identify the panel's position in the tutorial hierarchy, and also identifies the message number if an error occurs.
3. Screen printouts, obtained by specifying the PRINT, PRINT-HI, PRINTL, or PRINTLHI commands, include line numbers, current panel name, and message id.
4. In PDF, the index listing (option 3.1) for a partitioned data set includes TTR data for each mention of the data set.
5. If a dialog function is operating in the CANCEL error mode (the default), the panel that is displayed when an error occurs will allow you to force the dialog to continue in spite of the error. However, results from that point on will be unpredictable, and may result in an ABEND.
6. Except for the situation described in item 5, any ISPF-detected error, ABEND, or program interrupt forces an ABEND of ISPF. The user can also force an ABEND by entering ABEND or CRASH in the command line of any panel.
7. Pressing the PA1 key causes an immediate exit from ISPF.

Note: If an ISPF subtask ABENDs, a dump can be taken by pressing ENTER after the ABEND message appears, provided that a SYSUDUMP, SYSMDUMP, or SYSABEND data set has been allocated.

In **extended test mode**, ISPF operates the same way it does in test mode, except that all messages written to the ISPF log file are also displayed at the terminal.

Trace Modes

In **trace mode**, ISPF operates the same way it does in test mode, except that a message is written to the ISPF log file whenever any ISPF service is invoked from any one of the following applications:

- a CLIST
- an APL2 function

- a program dialog (through the CALL ISPEXEC service)
- an ISPF service that has detected an error (even if CONTROL ERRORS RETURN has been issued)

Note: Only CLIST, APL2, and CALL ISPEXEC service requests are recorded; program module requests for service made through CALL ISPLINK are not recorded in the log file.

In **extended trace mode**, ISPF operates the same way it does in trace mode, except that all messages written to the ISPF log file (including the trace messages) are also displayed at the terminal.

3270 DATA STREAM TRACE

The ISPTRACE CLIST found in data set ISP.V2R2M0.ISPSLIB prints the 3270 data stream for all full-screen writes and reads sent to the terminal. ISPTRACE writes the traced data to a data set named userid.TRACE.TESTLIST, which can either be browsed or printed. To obtain a trace of the 3290 buffers, edit ISPTRACE to increase the length of the buffer data that is written to the output data set. See the comments at the beginning of the CLIST for additional information.

REGISTER LINKAGE CONVENTIONS

When an ISPF program passes control to a separately-compiled ISPF program, the following linkage conventions are generally observed:

Register	Contents
1	Points to parameter list; each address in the list, in turn, points to a parameter.
2 - 12	Never used.
13	Contains the address of a 72-byte save area.
14	Contains the return address.
15	When a program is entered, contains its entry address; upon return to the invoking program, contains its return code.

ABEND CODES

ABENDs of the ISPF controller and processor tasks are controlled by STAE and STAI exit routines and by ISPF execution modes that are set using the ISPSTART TEST parameter (described in ISPF Dialog Management Services).

Under normal conditions (that is, when processor and controller dumps have not been requested by specifying the ISPSTART TEST command),

- When a processor task abends,
 1. No dump is taken

2. The controller reattaches the processor main driver (ISPPMD)
 3. The primary option menu is redisplayed for that logical screen
- When the controller task abends,
 1. No dump is taken
 2. ISPF terminates
 3. Control returns to TSO

The controller and processor tasks will issue the ABEND system service and allow dumps under certain situations. The ISPF modules that issue ABENDs, and their associated codes and reasons, are listed below:

- Various common subroutines - system code = 0C1 - In several ISPF common subroutines, parameter verification failure results in the execution of an instruction that has an invalid operation code (X'00'). This attempted execution causes an 0C1 ABEND, which indicates that the caller has passed an invalid value for a parameter that is required for continued execution of the subroutine.
- ABEND codes 111 or 222 - To produce these ABENDs the user must request processor dumps by entering the ISPSTART TEST command, and must also enter one of the following commands in the first eight bytes of the first input field on a logical screen:
 - ABEND - Terminates ISPF with code 111.
 - CRASH - Terminates ISPF with code 222 and prevents task termination from closing the edit backup/recovery data sets.
- ABEND code 997 (or X'3E5') - TPUT returned other than 0 or 8. A message (described in "Terminal Input/Output Error Codes.") will be displayed, and an attempt will be made to redisplay the full screen. If the redisplay fails twice, this ABEND will be issued.
- ABEND code 998 (or X'3E6') - Terminates the processing task for a split-screen situation.
- ABEND code 999 (or '3E7') - This ABEND is issued under the control task.
- ABEND code 999 (or X'3E7') - This ABEND is issued when a second error occurs while another error is already being processed.
- ABEND code 999 (or X'3E7') - This ABEND is issued when errors are encountered during the processor task initialization.
- ABEND code 999 (or X'3E7') - This ABEND is issued when a processor task initialization error occurs on initial entry.

TERMINAL INPUT/OUTPUT ERROR CODES

Below is a list of terminal I/O error messages and codes. The module that detects the error is shown in parentheses.

1. ** ISPF screen output error - code = nn - tput rc = xxxx **, where xxxx is a TPUT return code other than 0 or 8.
 - 41 TPUT return code not equal to 0 or 8
2. ** ISPF screen input error - code = nn **, where nn values are:
 - 21 TGET return code other than 0, 4, or 8
 - 22 Input stream size greater than input buffer size or 0
 - 23 Unknown attention identifier (AID)
 - 24 Invalid input AID
 - 25 Input stream size invalid for input AID
 - 26 Input cursor location not within physical screen
 - 28 First byte of input buffer field not an SBA (invalid input data)
 - 31 Byte preceding the physical screen field is past the end of the physical screen (input data from invalid screen position)
 - 32 Byte preceding the physical screen field is not an input attribute (input data from invalid screen position)
 - 33 Physical screen field not defined on panel (input data from invalid screen position)
 - 51 Physical screen field attribute not found in logical screen
 - 52 Byte preceding logical screen field is not an input attribute
 - 55 Physical screen size is greater than corresponding logical screen size

Notes:

1. The physical screen size is determined by ISPF during initialization.
2. The input buffer size is a variable based on the physical screen size.
3. The logical screen is the same size as the physical screen, and is the size that the processor task uses for screen I/O. When the 3290 is running in 62 x 160 partition mode, the SPLITV command makes the

logical screen width equal to 80. When a 3278 Model 5 is running in standard mode, the logical screen size is 24 x 80.

4. Only part of the logical screen appears on the physical screen when ISPF is running in split-screen mode. When the 3290 is running in 62 x 160 partition mode, the entire logical screen may be visible, depending on the position of the horizontal split line.
5. An input buffer field extends from an SBA to either the next SBA or the end of the input buffer.
6. A physical screen field extends from the location indicated in the input buffer SBA to the location of the next attribute byte in the physical screen.

LOG FILES

This section describes the ISPF log file, the LMF started task log data set, and the LMF activity log and controls review.

The ISPF Log File

The messages written to ISPLOG, the ISPF log file, contain the same message information that is presented on the terminal screen. In addition, if an ABEND occurs, failure data is collected and written to ISPLOG in the format shown below.

```
SUBTASK ABEND - VS 03.8  ISP 2.1.0000  ABEND CODE = 0C4  PSW FF850001 60104208
               - NAME MYPROG  EPA 1041F0  ISPSUBS EPA 10D358  ISPTBLS EPA 104C30

REGISTERS AT  - R0 00A1FBC0  R1 001158A0  R2 00115784  R3 00000000
ENTRY TO     - R4 00A1E614  R5 00000002  R6 00115780  R7 0010D358
ABEND        - R8 FFFFFFFF  R9 00105810  R10 00000000  R11 0000000C
              - R12 601041F6  R13 0010421C  R14 0010421C  R15 001041F0
```

The ISPLOG ABEND failure data includes:

- Operating system identification
- ISPF version identification
- ABEND code
- Program status word (PSW)
- Name and entry point address (EPA) of failing program

- EPA of the module ISPSUBS
- EPA of the module ISPTBLS
- Register contents at the time of failure

LMF Started Task Log Data Set

The LMF address space maintains a log data set, ISPLLOG, that records all errors occurring under the LMF started task. This data set is specified in the start procedure ISPF LMF.

Messages from ISPLLOG are not generally returned to the user. Instead, the user receives an error message and a request to ask the system programmer to provide access to the data set. ISPLLOG remains open for the duration of the started task, and therefore can be viewed only after the started task has terminated.

The messages that are written to the log data set diagnose the following types of errors:

- Control file errors
- Internal programming errors
- Activity log errors
- Put, Close, and Stow errors

Messages are written to the log data set in a two-step process. First, the message number is recorded. Second, the actual error is recorded. Here is an example of a message written to the log data set.

```
ISRLI065 - 'ISRLI004' IS THE MESSAGE NUMBER OF THE FOLLOWING MESSAGE:
READ FAILED ON MEMBER 'member name'.
```

Resource Access Control Facility (RACF) Considerations

The Resource Access Control Facility (RACF) Program Product, Program Number 5740-XXH, is used with LMF to ensure the integrity of the control file, the controlled data sets, and the activity logs.

You can get unpredictable results if you do not follow these steps when you use RACF:

1. Create the controls for the ISPF library
2. Allocate the data set
3. RACF-protect the library
 - a. The started task ID should have update authority
 - b. Set the universal access authority to read only

- c. To prevent accidental updates, modify the owner's authority to read only
4. Activate the controls on the library.

Because the controlled libraries are partitioned data sets, you will occasionally have to compress data sets. The suggested method is to create a background job with the started task userid and password. This job should allocate the desired libraries with a disposition of OLD. The operating system's normal data set contention process will ensure the serialization of the update.

LMF Activity Log and Controls Review

The control file allocated in the user's address space provides some of the input for the activity log and all the input for the controls review. Certain situations can cause problems with reviews. For instance, the activity log is a sequential file, which does not permit concurrent reads and updates; therefore, while an activity log is being formatted, you cannot promote or lock members, and an interrupt may occur. The same interrupt may occur during normal processing if a user allocates the activity log by specifying a BROWSE, EDIT, or TSO ALLOCATE command.

The review programs have an interface to an external sort program. The sort program you use must have the same interface as OS Sort/Merge. To prevent the user from seeing output from the sort invocation, it is written to the ISRLDUMMY file. If the user receives a "Sort failed" message and does not understand why, the ISRLDUMMY file can be reallocated to the terminal or to a data set so the user can look at the sort messages.

SPECIAL FEATURES OF THE ISPF TUTORIAL PANELS

This section describes diagnostic features that have been added to the ISPF tutorial panels.

Hexadecimal (System) ABEND Codes

The ISPF tutorial panels contain a list of the common system ABEND codes. These panels can be accessed by selecting option A1 of the tutorial index.

DAIR and Dynamic Allocation Return Codes

The ISPF tutorial panels contain a list of the TSO DAIR routine return codes and dynamic allocation return codes. The panels can be accessed by selecting option A (tutorial Appendix A) of the tutorial table of contents selection panel.

LMF Error Recovery Assistance

The ISPF tutorial panels are equipped with a diagnostic aid to help users recover from complex errors in LMF. Normally, when an error occurs, a short error message appears on the screen. To receive a more detailed message, the user can enter the "HELP" command.

When a complex error occurs, the short error message appears on the screen, and the user can request a longer message by entering the "HELP" command. In addition, if the user enters "HELP" once again, a message HELP panel appears, providing information on how to recover from the specific error situation that generated the message. For further assistance, the user can access the ISPF tutorial from the message HELP panel.

POTENTIAL SOURCES OF LMF ERRORS

This section discusses two potential sources of errors in the use and operation of LMF.

LMF Data Set Errors

If the promotion process yields unusual results, you must first check to be certain that there was no user error. Second, make sure that the contents of the control file are consistent with the contents of the controlled data sets, especially if you suspect that data sets have been damaged or back-leveled.

Warning: Do not attempt to edit the control file — you can damage or destroy its contents.

For example, suppose that damage to the controlled library ISPFPROJ.ENTRY.PLS causes member MEM1 to be lost. A user who subsequently tries to promote MEM1 will be informed that the member does not exist. If the user next attempts to promote a new MEM1 into ISPFPROJ.ENTRY.PLS, he will be told that the member has not been edit locked. If the user then tries to perform an edit lock, he will once again be informed that MEM1 does not exist. In order to correct this situation, the library administrator, in whose name the data set should be RACF-protected, must copy in a new version of the deleted member.

LMF Distribution Table Errors

LMF uses distribution tables to control the location of members that are promoted into more than one hierarchy. However, the range of control they offer can cause some confusion. For example, an inexperienced user might perform an edit lock on a member, and might then attempt to change its version code before re-promoting the member. Doing so can produce undesirable results that may not be discovered until a later time. Users should not attempt to change version codes unless they have a great deal of experience using distribution tables.

It may happen that several different versions of a member are edit-locked throughout a distributed hierarchy. To access all versions locked with the user's id, regardless of version code, the user can perform a promote free from PDF option 3.3.

INDEX

A

ABEND

- controller task 35
- processor task 34

ABEND codes

- issued by ISPF modules 35

C

- common control interface display request 10

compiler source code

- ISRLCODE 22
- ISRLEXPD 22
- member expansion 17
- member selection 17

- components of ISPF 3

- components of PDF 5

- conditional enqueue 24

- control card images

- ISPCTL 15

- control file requests

- handled by LMF started task 19

- controller task ABEND 35

conventions

- register linkage 34

- cross reference data

- ISRLXREF 22

D

DAIR

- See Dynamic Allocation Interface Routine

- data set errors

- in promotion process 40

- data set integrity enqueue 23

- data set names 16

- default 16

- NLS 16
 - generic 12
 - ISPF and PDF 10
 - system-generated 16
- data sets
 - DBCS 11
 - ISPF 10
 - Library Management Facility 21
 - NLS 11
 - password-protected 23
 - PDF 10
- data stream trace
 - 3270 34
- DBCS
 - See double-byte character set (DBCS)
- DEQ macro 23
- dequeue macro 23
- diagnostic aids
 - ABEND codes 34
 - extended test mode 33
 - extended trace mode 34
 - LMF distribution tables 41
 - log files 37
 - register linkage conventions 34
 - terminal I/O error codes 36
 - test mode 32
 - trace mode 33
 - tutorial panels 39
 - ABEND codes 39
 - DAIR return codes 40
 - LMF error recovery 40
 - 3270 data stream trace 34
- dialog test
 - PDF option 32
- display I/O 10
- display requests
 - common control interface 10
 - line I/O 10
 - normal full-screen I/O 10
- distribution tables
 - in LMF 41
- documents
 - related iii
- double-byte character set (DBCS)
 - printing ISPLIST 14
 - printing ISPLOG 14
- dump
 - how to take 32

requested by ISPSTART TEST 34
Dynamic Allocation Interface Routine
(DAIR)
ISPF tutorial panels 40
return codes 40

E

edit models
 ISPSLIB 12
editor member name enqueue 24
ENQ macro 23
enqueue macro 23
enqueue/dequeue logic
 integrity exposure 23
error messages and codes 36
error recovery assistance
 LMF
 ISPF tutorial panels 40
extended test mode
 TESTX 32
extended trace mode
 TRACEX 34
external register linkage
 conventions 34

F

failure data
 ISPLOG 37
file skeletons
 ISPSLIB 12
finding message ID 32
functional components
 ISPF 3
 PDF 5

H

hexadecimal ABEND codes
 ISPF tutorial panels 39
Hierarchical Storage Manager (HSM)
 ISPF interface 25
how to use ISPF 4
how to use PDF 6

HSM
 See Hierarchical Storage Manager

I

I/O data sets 10
I/O error messages and codes 36
input/output error messages and
 codes 36
integrity exposure
 enqueue/dequeue logic 23
interfaces
 Hierarchical Storage Manager 25
 Library Management Facility 18
 Program Control Facility 25
 Session Manager 26
 TSO/E Release 2 26
 TSO/TCAM 26
 TSO/VTAM 28
ISPCTLO
 used with edit submit command 15
ISPCTL1-4
 control card images 15
ISPF
 components
 functional 3
 physical 5
 how to use 4
 register linkage conventions 34
ISPF log file
 ISPLOG 37
ISPF modules
 ABEND codes 35
ISPF post/wait logic 8
ISPF task structure 7
ISPF tutorial panels
 DAIR return codes 40
 hexadecimal ABEND codes 39
 LMF error recovery assistance 40
 special features 39
 system ABEND codes 39
ISPFILU
 LIBDEF file tailoring output 14
ISPLIST
 printer output 14
 printing DBCS data 14
ISPLOG
 failure data format 37
 ISPF log file 37
 log messages 14
 PDF message data set 19

printing DBCS data 14
 ISPLST1-4
 OS utility listings 15
 ISPLUSR
 LIBDEF LOADLIBs 13
 ISPMALT
 DBCS message library 12
 ISPMLIB
 message library 12
 ISPMUSR
 LIBDEF message library 12
 ISPPALT
 DBCS panel library 12
 ISPPLIB
 panel definitions 12
 ISPPROF
 user profiles 13
 ISPPUSR
 LIBDEF panel library 12
 ISPSALT
 DBCS skeleton library 12
 ISPSLIB
 file skeletons 12
 ISPSTART TEST
 command to request dump 34
 ISPSUSR
 LIBDEF skeleton library 13
 ISPTABL
 output table library 13
 ISPTABU
 LIBDEF table output library 14
 ISPTLIB
 table library 13
 ISPTRACE
 CLIST for 3270 data stream 34
 ISPTUSR
 LIBDEF table input library 13
 ISPWRK1-4
 temporary output data sets 15
 ISRCFIL 22
 ISRLCODE 22
 ISRLDUMY 22
 ISRLEMX
 used with packed input data 17
 ISRLEXPD 22
 ISRLIN 22
 ISRLMSG 22
 ISRLOUT 22
 ISRLWKxx 22
 ISRLWK01 22
 ISRLWK02 22
 ISRLWK03 22

ISRLXREF 22
 ISRSCAN
 used with unpacked input data 17

L

LIBDEF
 ISPFILU 14
 ISPLUSR 13
 ISPMUSR 12
 ISPPUSR 12
 ISPSUSR 13
 ISPTABU 14
 ISPTUSR 13
 Library Management Facility (LMF)
 data sets 21
 distribution tables 41
 error recovery assistance 40
 PDF interface 18
 promotion process 40
 started task 19
 task structure 20
 line I/O display request 10
 linkage conventions 34
 LMF
 See Library Management Facility
 log messages
 ISPLOG 14

M

member expansion 17
 messages 22
 member expansion triggers 17
 member parts list 17
 generated by ISRLEMX 17
 input 17
 messages 22
 member parts list source code
 ISRLCODE 22
 message ID
 how to find 32
 message library
 ISPMALT 12
 ISPMLIB 12

N

names of data sets
 generic 12
 ISPF and PDF 10
normal full-screen I/O display
 request 10

O

options that use ISPCTL 15
output table library
 ISPTABL 13
output, printer
 ISPLIST 14

P

packed input data
 ISRLEMX 17
packed source code
 ISRLCODE 22
 member expansion 17
panel definitions
 ISPPLIB 12
panel library
 ISPPALT 12
parts of ISPF 3
parts of PDF 5
password-protected data sets 23
PCF
 See Program Control Facility
PDF
 components
 functional 5
 physical 6
 how to use 6
PDF option for dialog test 32
permanent data sets 10
physical components
 ISPF 5
 PDF 6
post/wait logic

ISPF 8
printer output
 ISPLIST 14
processor task ABEND 34
Program Control Facility (PCF)
 command authorization 25
 data set authorization 26
 ISPF interface 25
promotion process
 data set errors 40
promotion requests
 handled by LMF started task 19
publications
 related iii

R

register linkage conventions
 external 34
related documents iii
reserve macro 23
return codes
 DAIR 40

S

Session Manager
 ISPF interface 26
skeleton library
 ISPSALT 12
sort data sets
 ISRLIN 22
 ISRLOUT 22
split-screen capability
 ISPF 4
 PDF 6
split-screen mode
 task structure 8
started task
 Library Management Facility 19
SYSABEND 32
SYSMDUMP 32
system ABEND codes
 on ISPF tutorial panels 39
SYSUDUMP 32

T

table input library
 ISPTLIB 13
taking a dump 32
task structure
 ISPF 7
 LMF started task 20
temporary data sets 10
temporary output data sets
 ISPWRK1-4 15
terminal input/output error codes 36
TEST
 ISPSTART parameter 32
test mode 32
testing a dialog 32
TEXTX
 ISPSTART parameter 32
TRACE
 ISPSTART parameter 32
trace modes 33
TRACEX
 ISPSTART parameter 32
triggers
 member expansion 17
 user-defined 17
TSO Session Manager
 ISPF interface 26
TSO/E Release 2
 ISPF interface 26

TSO/TCAM
 ISPF interface 26
TSO/VTAM
 ISPF interface 28
tutorial panel
 displays message ID 32
tutorial panels
 special features 39
types of display requests 10

U

unconditional enqueue 23
unpacked input data
 ISRSCAN 17
unpacked source code
 ISRLEXPD 22
user profiles
 ISPPROF 13
user-defined triggers 17

Numerics

3270 data stream trace 34
3290
 in 62 x 160 partition mode 37

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. It will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

What is your occupation? _____

Number of latest Technical Newsletter (if any) concerning this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Reader's Comment Form

Cut or Fold Along Line

Fold and tape

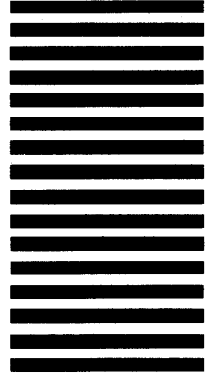
Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department T45, Information Development
P. O. Box 60000
Cary, North Carolina 27511

Fold and tape

Please Do Not Staple

Fold and tape

ISPF and ISPF/PDF for MVS

Diagnosis

Printed in U.S.A.

SC34-4020-1



This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. It will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

What is your occupation? _____

Number of latest Technical Newsletter (if any) concerning this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Reader's Comment Form

Cut or Fold Along Line

Fold and tape

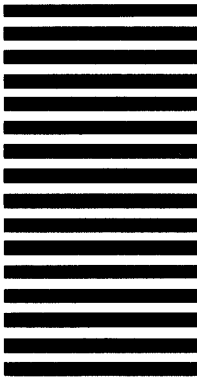
Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department T45, Information Development
P. O. Box 60000
Cary, North Carolina 27511

Fold and tape

Please Do Not Staple

Fold and tape

ISPF and ISPF/PDF for MVS

Diagnosis

Printed in U.S.A.

SC34-4020-1



This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. It will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

What is your occupation? _____

Number of latest Technical Newsletter (if any) concerning this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Reader's Comment Form

Cut or Fold Along Line

Fold and tape

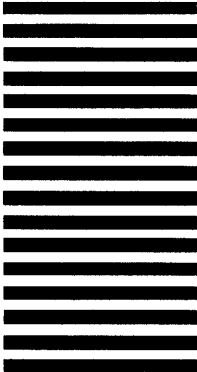
Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department T45, Information Development
P. O. Box 60000
Cary, North Carolina 27511

Fold and tape

Please Do Not Staple

Fold and tape

ISPF and ISPF/PDF for MVS

Diagnosis

Printed in U.S.A.

SC34-4020-1



This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. It will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

What is your occupation? _____

Number of latest Technical Newsletter (if any) concerning this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Reader's Comment Form

Cut or Fold Along Line

Fold and tape

Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS

PERMIT NO. 40

ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department T45, Information Development
P. O. Box 60000
Cary, North Carolina 27511

Fold and tape

Please Do Not Staple

Fold and tape

ISPF and ISPF/PDF for MVS

Diagnosis

Printed in U.S.A.

SC34-4020-1



Publication Number
SC34-4020-1

File Number
S370/4300-39

Program Numbers
5665-317
5665-319

Printed in
U.S.A.

IBM

SC34-4020-01

