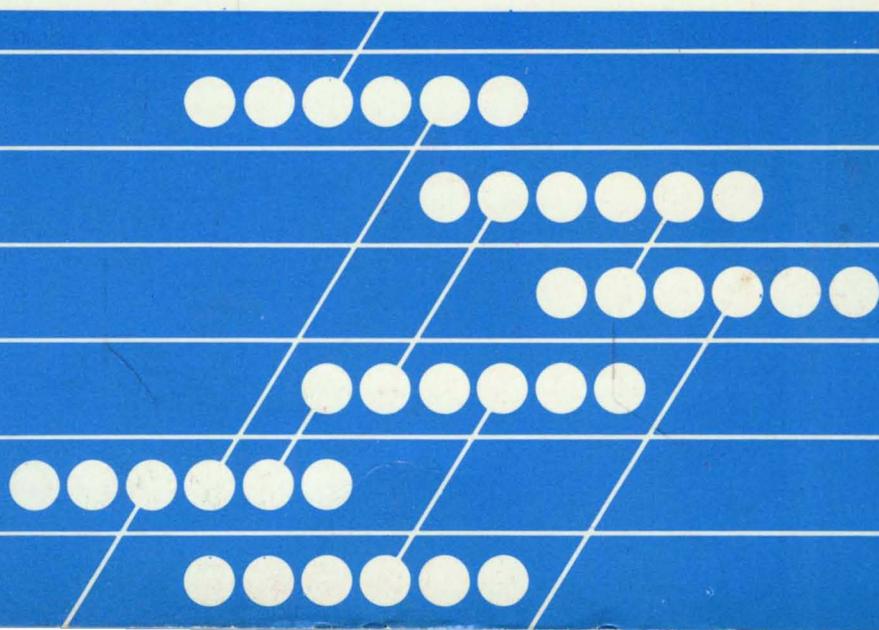


Interactive  
System Productivity Facility

General Information

MVS, VM, and VSE

**IBM**



---

# Interactive System Productivity Facility

---

## General Information

---

## MVS, VM, and VSE

---

Publication Number  
GC34-2078-2

File Number  
S370/4300-39

---

Program Number		
ISPF	5668-960	
ISPF/PDF	5665-268	MVS
	5664-172	VM
	5666-281	VSE

### Third Edition (July 1983)

This edition applies to the Interactive System Productivity Facility (ISPF) and the ISPF/Program Development Facility (ISPF/PDF, or PDF) Program Products, as shown in the table below, and to all subsequent releases until otherwise indicated by technical newsletters.

Program Product	Program Number			
	System Release	OS/VS2	VM/SP	VSE/AF*
	3.8	1.0	3.5	
ISPF	5668.960	5668.960	5668.960	
ISPF/PDF	5665.268	5664.172	5666.281	

\* Refer to Chapter 1 "Introduction" for a more complete explanation of system and release identification for the VSE environment.

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, 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 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, Publications, Dept. Z59, Bldg. 931, P.O. Box 390, Poughkeepsie, New York U.S.A. 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

## PREFACE

This manual provides an overview and initial planning information for two related program products: the Interactive System Productivity Facility (ISPF) and the ISPF/Program Development Facility (ISPF/PDF, or PDF).

These cross-system program products are designed to increase user productivity in the development of applications on display terminals. There is special emphasis on the development and use of interactive applications, called dialogs. The dialog management functions of ISPF and the program development capabilities of PDF are contained in separate products; for MVS and VM/SP they were previously combined in the System Productivity Facility (SPF) Program Product.

The manual covers the following topics:

- **Introduction** describes the purposes of the products and shows the operating environment.
- **Dialog Management** describes the concepts and facilities of the ISPF Program Product.
- **Program Development** describes the concepts and facilities of the PDF Program Product.
- **Planning** describes the system requirements, responsibilities of the customer, and related documentation.
- **Changes from System Productivity Facility** is applicable only to MVS and VM/SP customers and compares the functions of the ISPF and PDF program products to their predecessor program product, SPF.
- **Example of an ISPF Dialog** shows how a fairly simple interactive application could be developed using ISPF, with the functions written as an MVS/TSO CLIST. (page 53)
- **Example of a PDF Session** shows a brief scenario of program development by a user at a terminal.
- **Glossary** defines a number of terms that apply to these program products. Also defined are terms used in this manual to bridge the differences in terminology among the various host environments; for example, library, file, and command procedure.

This publication is intended for installation managers and technical personnel who need basic information about ISPF and PDF. For publications containing additional information, see Chapter 4, "Planning."



# CONTENTS

<b>Chapter 1. Introduction</b> . . . . .	<b>1</b>
Operating Environment . . . . .	1
ISPF Dialog Management Functions . . . . .	2
ISPF/PDF Program Development Functions . . . . .	4
<b>Chapter 2. Dialog Management</b> . . . . .	<b>5</b>
Elements of a Dialog . . . . .	5
Organization of a Dialog . . . . .	6
Use of Variables . . . . .	9
Formatting of Displays . . . . .	10
Dialog Management Services . . . . .	14
Select Service . . . . .	14
Display Services . . . . .	16
Table Services . . . . .	16
General Services . . . . .	17
Row Services . . . . .	18
File Tailoring Services . . . . .	19
Variable Services . . . . .	19
ISPF Invocation . . . . .	21
<b>Chapter 3. Program Development</b> . . . . .	<b>23</b>
PDF Displays . . . . .	23
Types of Panels . . . . .	23
Screen Size . . . . .	24
Panel Format . . . . .	24
Library and Data Set Access . . . . .	25
Library Structure . . . . .	25
MVS and VM/SP Libraries . . . . .	25
VSE System Libraries . . . . .	27
ICCF Libraries . . . . .	28
Conventional Data Set and File Naming Conventions . . . . .	29
MVS Data Set Names . . . . .	29
CMS File Names . . . . .	29
VSE Data Set Names . . . . .	29
PDF Invocation . . . . .	30
PDF Options . . . . .	31
Option 0 - ISPF PARMS . . . . .	31
Option 1 - BROWSE . . . . .	31
Option 2 - EDIT . . . . .	31
Option 3 - UTILITIES . . . . .	34
Option 4 - FOREGROUND (MVS and VM/SP only) . . . . .	34
Option 5 - BATCH . . . . .	35
Option 6 - COMMAND . . . . .	35
Option 7 - DIALOG TEST . . . . .	35
Option C - CHANGES (MVS and VM/SP only) . . . . .	36
Option T - TUTORIAL . . . . .	36
Option X - EXIT . . . . .	36
<b>Chapter 4. Planning</b> . . . . .	<b>37</b>

Programming Requirements . . . . .	37
MVS Requirements . . . . .	37
VM/SP Requirements . . . . .	38
VSE/ICCF Requirements . . . . .	39
Machine Requirements . . . . .	40
Terminal Requirements . . . . .	41
Customer Responsibilities . . . . .	42
Migration . . . . .	43
Converting Dialogs From One Environment To Another . . . . .	44
Supporting Publications . . . . .	44
<b>Appendix A. Changes From System Productivity Facility (SPF)</b> . . . . .	<b>47</b>
Dialog Management Changes . . . . .	47
Program Development Facility Changes . . . . .	50
<b>Appendix B. Example of an ISPF Dialog</b> . . . . .	<b>53</b>
Overview . . . . .	53
Dialog Components . . . . .	53
Invocation . . . . .	54
Description of Processing . . . . .	54
Primary Option Menu Definition EMPL . . . . .	56
Panel Definition . . . . .	56
Panel Display . . . . .	56
Function EMPLFN . . . . .	56
First Data Entry Panel EMPLA . . . . .	60
Panel Definition . . . . .	60
Panel Display . . . . .	60
Second Data Entry Panel EMPLB . . . . .	61
Panel Definition . . . . .	61
Panel Display . . . . .	62
Message Definitions EMPX21 . . . . .	63
<b>Appendix C. Example of a PDF Session</b> . . . . .	<b>65</b>
<b>Glossary</b> . . . . .	<b>73</b>
<b>Index</b> . . . . .	<b>77</b>

## FIGURES

1.	ISPF Environment . . . . .	2
2.	Compilers In Which Functions May be Written . . . . .	3
3.	Typical Dialog Starting with a Selection Panel . . . . .	7
4.	Typical Dialog Starting with a Function . . . . .	8
5.	Sharing Variables Through the VPUT/VGET Service . . . . .	10
6.	Sample Panel Definition . . . . .	12
7.	Sample Panel Display . . . . .	14
8.	Control Flow Between ISPF Services and Libraries or Files . . . . .	15
9.	Sample Table . . . . .	17
10.	ISPF Master Application Menu for MVS . . . . .	21
11.	ISPF Sample Primary Option Menu for MVS . . . . .	22
12.	Hierarchy of ISPF Libraries . . . . .	26
13.	PDF Primary Option Menu for MVS . . . . .	30
14.	Organization of Employee Records Application . . . . .	54



## CHAPTER 1. INTRODUCTION

The Interactive System Productivity Facility (ISPF) and the ISPF/Program Development Facility (ISPF/PDF, or PDF) are related IBM program products. Together, they are designed to increase user productivity in the development of applications by taking advantage of the features of display terminals. They contain special functions for the development and use of interactive applications (**dialogs**). In a dialog, the user communicates his or her needs to the application by responding to inquiries presented on the screen by the application. Specifically:

- ISPF is a dialog manager for interactive applications. It provides control and services to allow processing of the dialogs in different host environments.
- PDF is a facility that aids in the development of various types of applications, including dialogs. It uses display terminals in an interactive environment to assist with many programming tasks.

The dialog management functions of ISPF and the program development functions of PDF were previously combined in the predecessor program product, the System Productivity Facility (SPF). The program was split to reduce the cost for the customers who need only dialog management services or who do not want to develop their own dialogs. The changes from SPF are described in Appendix A.

### OPERATING ENVIRONMENT

As shown in Figure 1, the ISPF dialog manager is, conceptually, an environment-independent extension of the MVS, VM/SP, and VSE/AF host operating environments. ISPF is the base product for PDF and for other interactive applications, or dialogs. PDF, on the other hand, is itself a dialog and runs with the control and services of ISPF.

**VSE Note:** In this document, references to the VSE host system implies VSE/AF Release 3.5. This is the name given to the system outside of the System Installation Productivity Option (SIPO) environment. If you are using VSE in a SIPO environment, the name given to the host system is VSE/SP Release 1.1.

Note in the figure that PDF and other dialogs have access not only to the services provided by ISPF, but also to those provided by the host operating system.

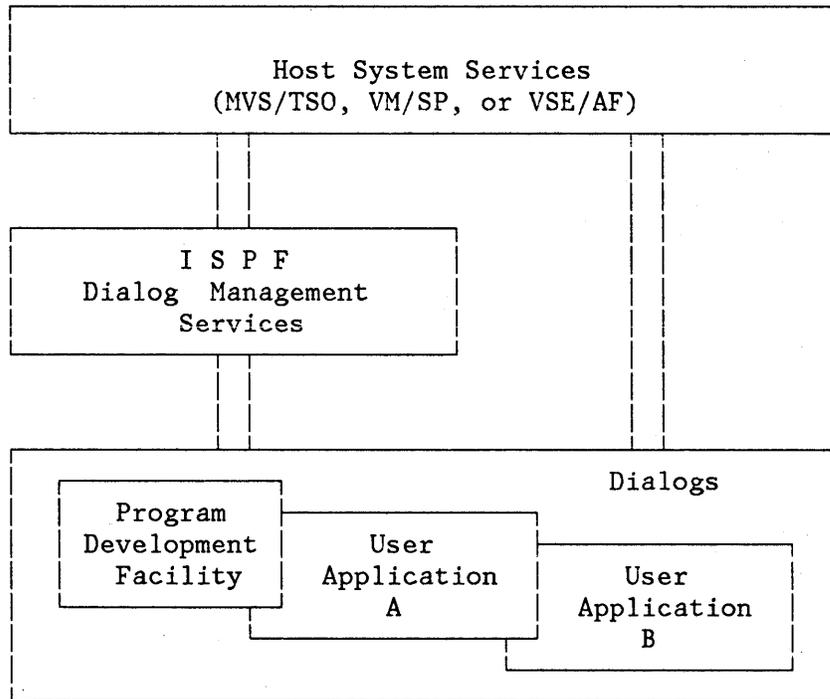


Figure 1. ISPF Environment

---

## ISPF DIALOG MANAGEMENT FUNCTIONS

ISPF allows for a wide variety of dialog organizations (see "Organization of a Dialog" in Chapter 2). A dialog managed by ISPF consists primarily of the following items:

- Selection panels (menus), from which the user selects a particular processing option.
- Functions (commands or programs), that perform the requested processing.
- Data entry panels, on which the user supplies additional information needed for the application.

Functions may take the form of an MVS TSO **CLIST**, a VM/SP CMS **EXEC**, or be written in a **language** such as assembler, or one of the compilers shown in Figure 2.

---

Compiler	MVS	VM/SP	VSE/AF
PL/I Optimizer	5734-PL1		5736-PL1
COBOL	5740-CB1		5746-CB1
FORTRAN IV G1	5734-F02		--
VS FORTRAN Release 2.0	--		5748-F03

Figure 2. Compilers In Which Functions May be Written

---

One way to create and test these functions is through the related PDF Program Product. This product provides a model facility that simplifies the generation of many of the dialog elements for several languages, and a test facility that helps the developer find problems in the use of ISPF.

When a dialog is invoked, ISPF facilities:

- **Display a hierarchy of menus** based on user selections.
- **Invoke functions** from the menus. These functions may be command procedures (CLISTs or EXECs) or programs.
- **Communicate with the user** through data entry displays and messages.
- **Provide online help** and tutorial information.
- **Generate sequential output** to be passed as input to another process; for example, JCL to be submitted as a batch job, or SCRIPT/VS text to be formatted for printing (not on VSE).
- **Maintain user-entered or program-generated data** during this session or from one session to another; for example, ISPF facilities:
  - Define and control user **variables**
  - Build and maintain permanent **tables** of user information
- **Provide split-screen** displays through which the user may partition the display screen into two "logical" areas.
- **Intercept user-entered commands** and take appropriate action based on information contained in command tables.

## ISPF/PDF PROGRAM DEVELOPMENT FUNCTIONS

PDF is an ISPF dialog that can help increase programmer productivity by simplifying frequently performed programming tasks. It provides browse and edit services that also may be invoked from a user application. PDF may be used by individual programmers, each working on a separate project, or by a group of programmers working on a common project. The significant features include:

- **Multilevel programming library** support that allows maintenance and tracking of program segments at different versions or levels.  
  
**VSE Note:** The VSE user must use normal VSE library naming conventions to provide this function.
- **Full-screen, context editing** that allows multiple additions and changes to information on a screen with only one interaction with the host system. The most frequently used editing functions are invoked using simple, one-character commands.
- **Models** that help a user develop dialog panels, messages, function routines, file skeletons, and tables.
- **Scrolling** (in any direction) of source data and listings. Also, location of data by character string or line number.
- **Utilities** to specify and maintain libraries and data sets (MVS and VSE) or files (VM/SP).
- **Interfaces** to standard **language processors** (compilers, assemblers, and linkage editors or loaders). These processors may be invoked as batch jobs in all three host environments and as foreground jobs in the MVS and VM/SP environments.
- **Dialog test** facilities that help a user test dialog applications.
- **Documentation preparation assistance** consisting of text editing facilities, and, on MVS and VM/SP only, an interface to the Document Composition Facility Program Product. Although there is no direct interface to this facility in PDF for VSE, a user can use the Document Composition Facility and its associated Document Library Facility Program Product through the PDF batch facility.
- **Online tutorial** for instruction and reference. This feature is especially valuable for the new or occasional user.
- **Browse and Edit services** that may be invoked from user applications.

## CHAPTER 2. DIALOG MANAGEMENT

ISPF processes interactive applications called **dialogs**. Dialogs are composed of control or program elements that tell ISPF about the flow or paths to be taken by the application, and data elements that supply the information to be presented to the end user of the dialog.

The interface between the end user and the application is a panel or visible display image that conveys information to the user. The user responds to this panel either to supply data or to control the flow of the application.

The user may partition the display screen into two "logical" screens at any time during a dialog when a command can be invoked. The two logical (split) screens are treated as though they were independent terminals.

In split-screen mode, only one of the logical screens is considered active at one time. The location of the cursor is used to identify which of the two screens is active.

Split-screen mode is entered by means of the SPLIT command or a PF key. Split-screen mode is terminated by ending the application on either logical screen. The remaining logical screen is then expanded to full screen size.

An example of an ISPF dialog for MVS TSO is shown in Appendix B.

### ELEMENTS OF A DIALOG

A dialog is made up of the following:

- **Panels**, which are predefined display images. They may be:
  - **Selection panels (menus)**, which prompt the user to select from specified options.
  - **Data entry displays**, which prompt the user to provide information.
  - **Tables** which display selected entries from a two-dimensional array of data.
  - **Information displays**, such as tutorial panels (also known as a text panels).

The functions of selection and data entry may be combined on one panel.

- **Functions** (or dialog functions), which are programs or command procedures that perform processing or control the flow of a dialog,

CLIST / EXEC

or both. These programs may be written in an assembler language or a high-level language such as COBOL, PL/I, or FORTRAN; or they may be written as a command procedure (MVS TSO CLIST, VM/SP EXEC2) for a particular system.

- **Messages**, which are comments that provide specific information to the user. They may provide information about the flow of the dialog, an acknowledgment that a request has been processed, or a warning to alert the user that something is not progressing as expected.
- **Variables**, which are character strings (a word, a value — any kind of data) referred to by a symbolic name. They are the main means of communication between the ISPF services and parts of a dialog such as panels, functions and tables. Dialog variables may also be used for communication between functions.
- **Tables**, which are arrays of data. They may be used to store data temporarily, or they may be retained for use in another session. They may also be shared between different applications.
- **File skeletons**, which are a generalized representation of sequential data that may be customized during dialog processing to produce an output file. The customization consists of the selection of lines to be included in the output, and the substitution of variables obtained from functions, panels, or tables.

These various elements are stored in system or user libraries whose organization is dependent upon the operating system under which ISPF is running. Panels, messages, functions, and file skeletons are created using an appropriate editor (such as the PDF editor) and are stored in libraries. Functions are coded in one of the languages used on the user's system. The particular language may require an assembly or compilation and link editing before the functions can be stored in libraries. Tables are dynamically created and updated during the execution of a dialog and stored in a library appropriate to the operating system.

A dialog does not have to include all types of elements. In particular, tables and file skeletons may be omitted in some applications.

## ORGANIZATION OF A DIALOG

A dialog may be organized in a variety of ways to suit the requirements of the application and the needs of the end user.

A typical dialog organization, as shown in Figure 3, starts with display of the highest level menu, the primary option menu for the application. User options selected from this menu may result in the invocation of a dialog function, or the display of a lower level menu. Each lower level menu may also pass control to other functions or display still other lower level menus.

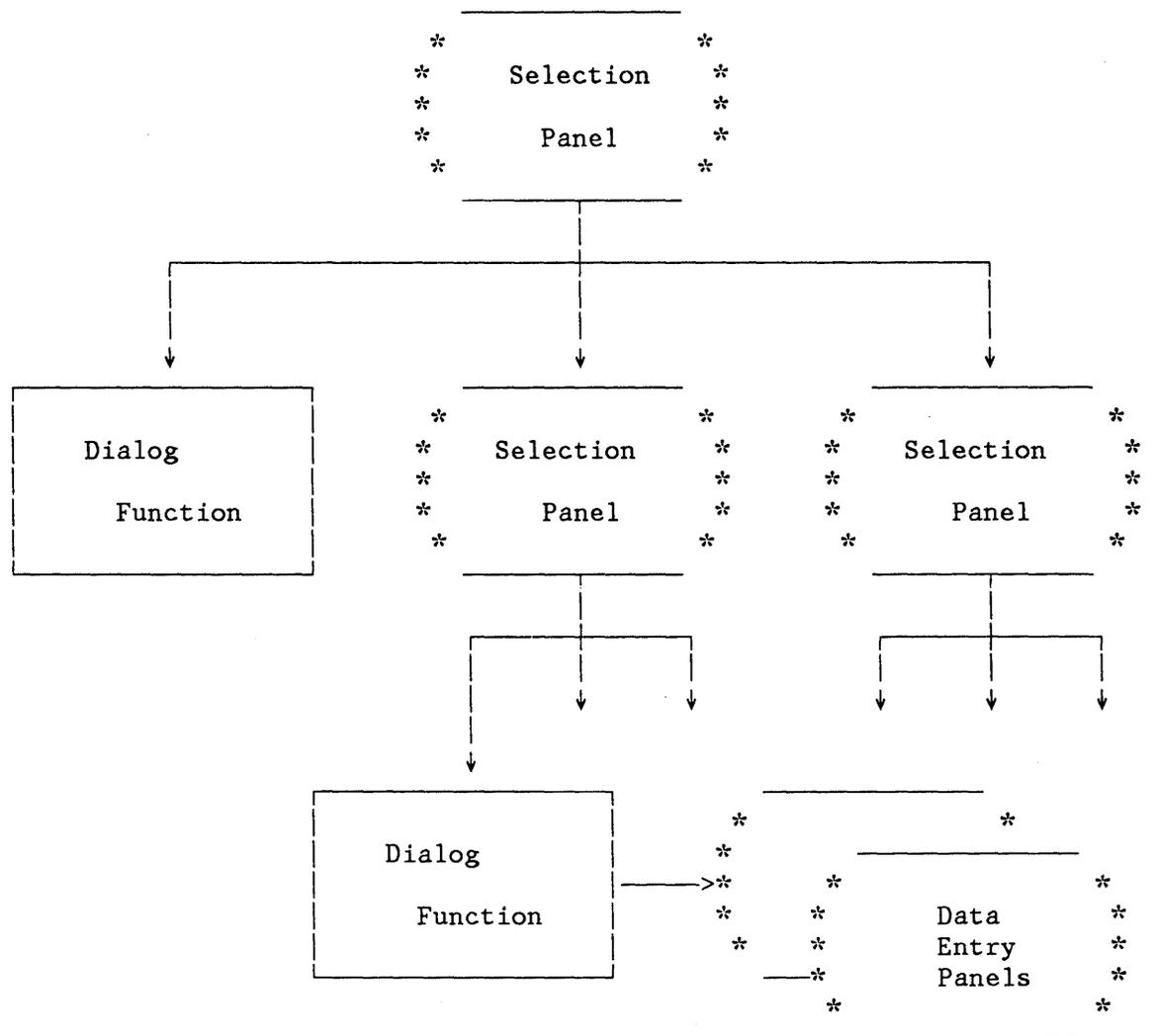


Figure 3. Typical Dialog Starting with a Selection Panel

Eventually, a dialog function receives control. The function may use any of the dialog services provided by ISPF. In particular, the function may continue the interaction with the end user by means of the DISPLAY service. Typically, the function displays data entry panels to prompt the user for information. When the function completes, the panel from which it was invoked is redisplayed.

Figure 4 shows another type of dialog organization, where a dialog function receives control first, prior to the display of a panel. The function may perform application-dependent initialization, and display

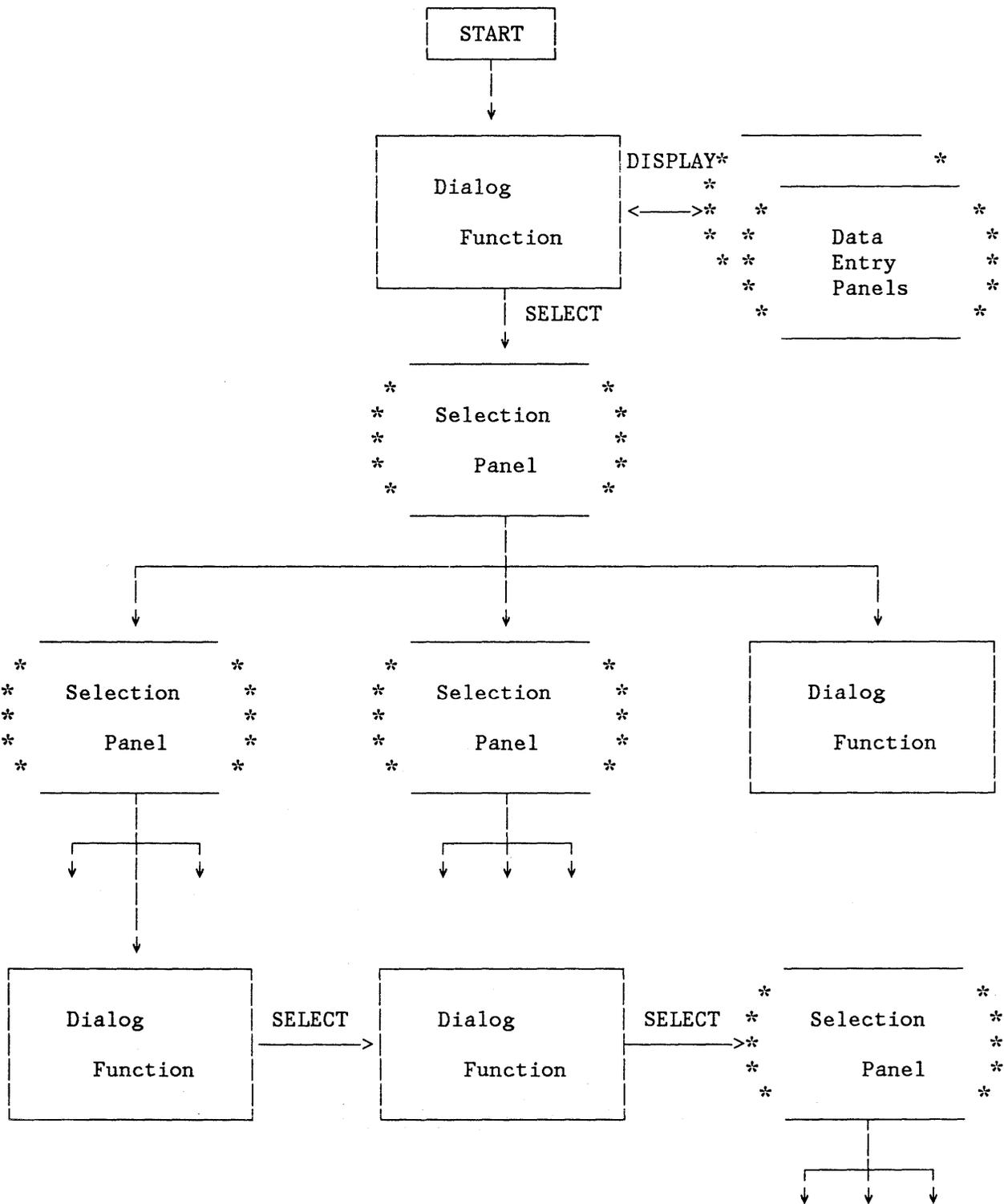


Figure 4. Typical Dialog Starting with a Function

data entry panels to prompt the user for basic information. It may then start the selection process by using the SELECT service to display the primary option menu for the application.

As shown, a function may also invoke another function, via the SELECT service, without displaying a panel. This provides a convenient way to pass control from a program-coded function to a command-coded function, or vice-versa. The invoked function then starts a lower-level selection panel process, again by using the SELECT service.

## USE OF VARIABLES

The ISPF variable services (see "Variable Services") control the definition and use of dialog variables.

Dialog variables are identified to ISPF services either implicitly or explicitly:

- All MVS TSO CLIST and VM/SP CMS EXEC2 variables are automatically treated as dialog variables. No special action is required to define them to ISPF. The variables are created dynamically either by the execution of the command procedure or by the ISPF service used by the command procedure.

**VSE Note:** ISPF under VSE does not support any command procedure language for the purpose of writing functions.

- Within assembled or compiled programs, variables may be defined either explicitly to ISPF through the use of the VDEFINE service, or implicitly through reference to a variable previously defined on a panel or through VGET.

When a variable is created, it is associated with the function that is currently in control and may not be referenced by other functions. When the function finishes processing, all of its variables (explicit and implicit) are automatically deleted.

When a function invokes a lower level function through the SELECT service, the lower level function has its own set of variables (which may have the same names as variables belonging to other functions). Again, the lower level function may not access the variables of the invoking function.

As shown in Figure 5, two mechanisms allow sharing of variables between functions:

- Shared variable pool
- User profile

The shared variable pool allows communication of variables between functions that belong to the same application. When no particular application is specified, the variables are available to all applications invoked during this session. A function may copy one or

more of its variables into the shared pool by means of the VPUT service.. Another function may then obtain the current value of the variable by means of the VGET service. These variables are not retained from session to session.

The user profile contains variables that are automatically retained from session to session for each user. The VPUT and VGET services are also used to store and retrieve variables into and from the user profile.

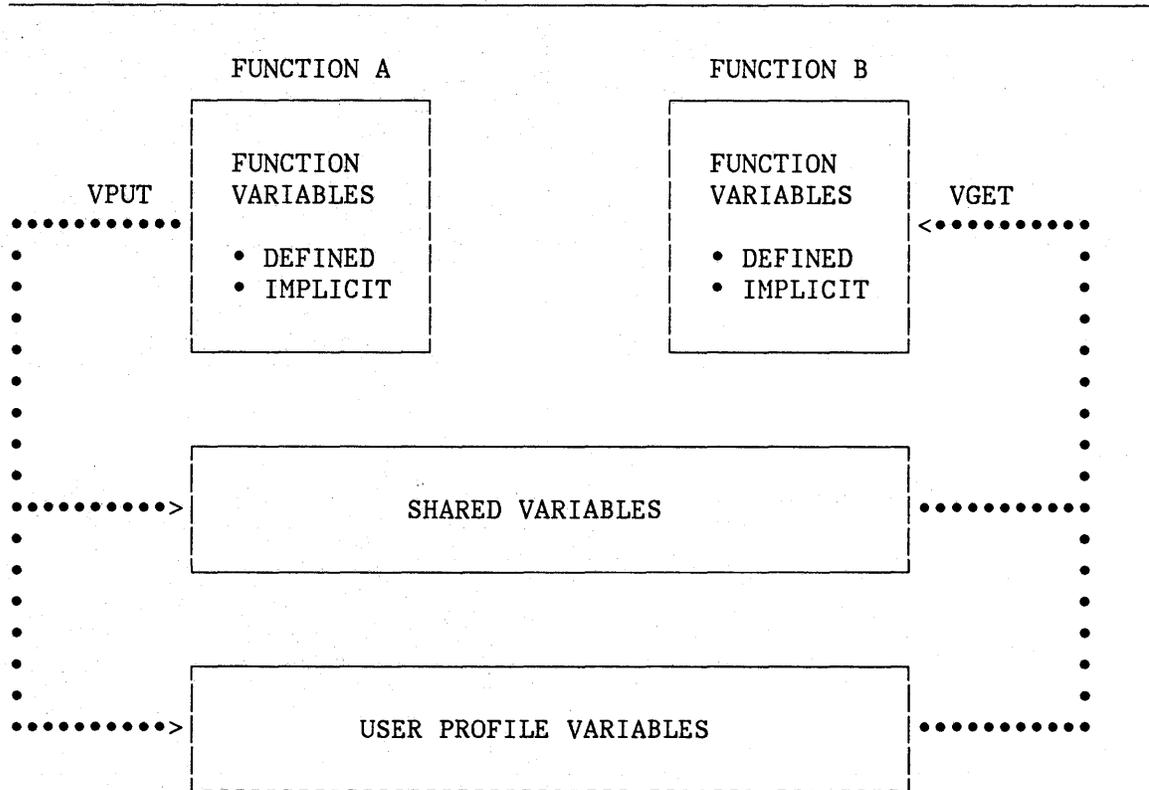


Figure 5. Sharing Variables Through the VPUT/VGET Service

---

## FORMATTING OF DISPLAYS

The ISPF display services (see "Dialog Management Services" and "Display Services") make it easy for programmers to define display images. The following basic concepts are used:

- Display formats are specified by panel definitions that include a "picture" of what the end user will see. This simplifies panel creation and maintenance.
- Messages are treated as supplementary information that may be displayed with a panel, or superimposed on the current panel.

- Panel and message definitions are maintained in libraries separate from the functions. This allows an installation to custom-tailor the display formats easily and simplifies translation to other languages.
- Panel and message definitions are created and maintained by editing directly into the panel and message libraries; no compile or preprocessing steps are required.
- Panel and message definitions contain a mixture of literal text, which is displayed "as is"; variables, for which the current value is dynamically substituted at the time of display; and control information, which is interpreted by the DISPLAY service.

Panel definitions may contain the following sections:

- Attribute section (optional) — defines the special characters that will be used in the body of the panel definition to represent attribute (start of field) bytes. This section, if included, may override the default attribute definitions. Typically, this section is used to define additional attribute characters.
- Body (required) — defines the format of the panel as seen by the user (the picture section of the panel), and defines the names of any variable fields used on the panel.

**VSE Note:** Although you may create a blank line in a panel definition, it will not be written to the library by VSE. Therefore, if you wish to have blank lines on the visible part of a panel, you must provide at least a field delimiter attribute character, such as "+", in the first position.

- Model section (table display panels only) — defines the format of each line of scrollable data. This section is required for table display panels, and is invalid for other types of panels.
- Initialization section (optional) — specifies the processing that is to occur prior to displaying the panel. Typically, this section defines how any variables are to be initialized.
- Processing section (optional) — specifies the processing that is to occur after the panel has been displayed. Typically, this section defines how variables are to be verified and translated.

A sample panel definition is shown in Figure 6 (another example is shown in Appendix B). It has no attribute section, and simply uses the following default attribute characters:

```
% (percent sign) - text (protected) field, high intensity
+ (plus sign)    - text (protected) field, low intensity
_ (underscore)  - input (unprotected) field, high intensity
```

```

%----- EMPLOYEE RECORDS -----
%COMMAND ==> _ZCMD +

%EMPLOYEE SERIAL: &EMPSER

+ TYPE OF CHANGE%=> _TYPECHG + (NEW, UPDATE, OR DELETE)

+ EMPLOYEE NAME:
+ LAST %=> _LNAME +
+ FIRST %=> _FNAME +
+ INITIAL%=> _I+

+ HOME ADDRESS:
+ LINE 1 %=> _ADDR1 +
+ LINE 2 %=> _ADDR2 +
+ LINE 3 %=> _ADDR3 +
+ LINE 4 %=> _ADDR4 +

+ HOME PHONE:
+ AREA CODE %=> _PHA+
+ LOCAL NUMBER%=> _PHNUM +

)INIT
  .HELP = PERS032
  .CURSOR = TYPECHG

)PROC
  VER (&TYPECHG,LIST,NEW,UPDATE,DELETE,MSG=EMPX201)
  VER (&LNAME,ALPHA)
  VER (&FNAME,ALPHA)
  VER (&I,ALPHA)
  VER (&PHA,NUM)
  VER (&PHNUM,PICT,'NNN-NNNN')

)END

```

*hi-intensity* {

*low intensity* {

*single character input* {

*body* }

*initialization* }

*procedure* }

Figure 6. Sample Panel Definition

Each text attribute character (% or +) is followed by the information to be displayed. Substitutable variables, consisting of a dialog variable name preceded by an ampersand (&), may be included in the text. Each input attribute character ( ) is followed immediately by a dialog variable name, with no intervening ampersand. The field ends with a plus sign (+), which indicates the maximum length of the input variable.

The panel body section extends to the ")INIT" header statement, which starts the initialization section. The initialization section in this

example establishes PERS032 as the name of the related help panel (in the event that the user requests help while viewing the panel) and sets the initial cursor position to the TYPECHG field.

After the user enters information, the data in each panel input field is automatically saved in the corresponding dialog variable. Then the processing section of the panel definition, beginning with the ")PROC" header statement, is executed. In this example, the processing section contains VER statements to verify that information entered by the user meets the following criteria:

- Type of change is NEW, UPDATE, or DELETE. The verification keyword LIST indicates that the value entered must match one of the items in the list that follows the keyword.
- Last name, first name, and initial fields contain all alphabetic characters.
- Area code field contains all numeric characters.
- Phone number consists of three numeric characters, followed by a hyphen, followed by four numeric characters. The verification keyword PICT indicates that the value entered must contain characters that match the corresponding type of character in the picture string.

If any verification check fails, a message is automatically displayed and the user is given an opportunity to correct the error. The first VER statement explicitly designates the message id (EMPX201) to be displayed. For the other VER statements, an appropriate default message is displayed. When the user corrects the error, the variables are again stored and the processing section re-executed.

Figure 7 shows how the panel would look when displayed, assuming that the dialog function had first set variables EMPSER to 123456 and TYPCHG to NEW before calling for the display of the panel.

---

```
----- EMPLOYEE RECORDS -----
COMMAND ==> _
EMPLOYEE SERIAL: 123456

TYPE OF CHANGE ==> NEW          (NEW, UPDATE, OR DELETE)

EMPLOYEE NAME:
  LAST   ==>
  FIRST  ==>
  INITIAL ==>

HOME ADDRESS:
  LINE 1 ==>
  LINE 2 ==>
  LINE 3 ==>
  LINE 4 ==>

HOME PHONE:
  AREA CODE ==>
  LOCAL NUMBER ==>
```

Figure 7. Sample Panel Display

---

## DIALOG MANAGEMENT SERVICES

ISPF provides the following dialog management services:

- Select service (p. 14)
- Display services (p. 16)
- Table services (p. 16)
- File tailoring services (p. 19)
- Variable services (p. 19)

Figure 8 shows the flow of control and data between the ISPF services and the libraries and data sets.

### Select Service

SELECT is both a control facility used by ISPF and a service that can be requested by a dialog. It controls the sequence of selection panels based on user inputs, and it invokes dialog functions.

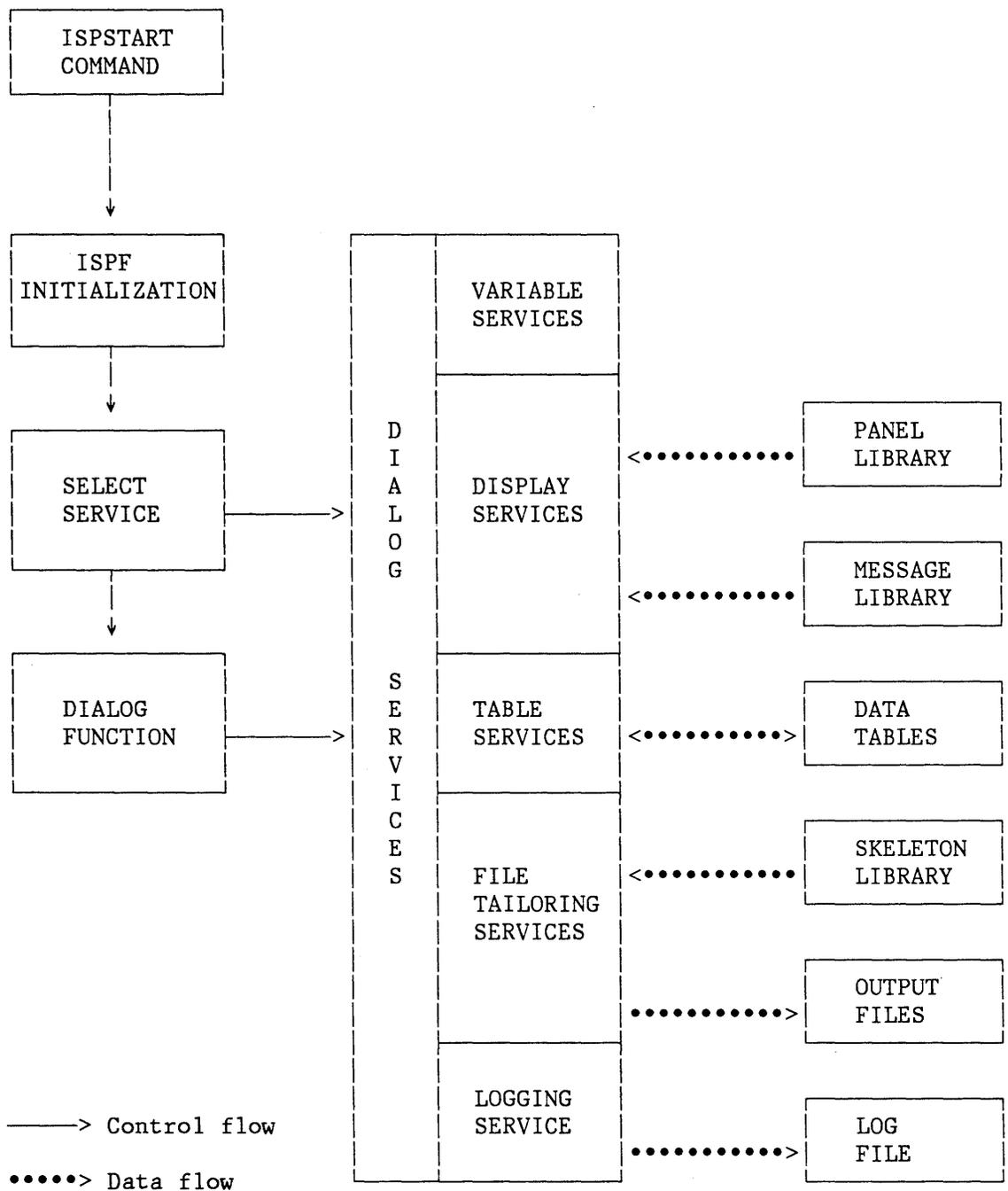


Figure 8. Control Flow Between ISPF Services and Libraries or Files

When an ISPF application is first started (by logon, command procedure, etc.), SELECT displays the first selection panel or invokes the first dialog function, passing it any initial parameters.

When SELECT is invoked from a dialog function, it can start the display of a new selection panel hierarchy or it can invoke other functions without displaying a new selection panel.

## Display Services

Display services allow a dialog to display information and interpret responses from the user. There are three display services:

**DISPLAY** Displays a panel. The DISPLAY service:

- Initializes variable information in the panel
- Displays the panel on the screen (with an optional message)
- Processes the response and saves variables

If the user enters information on the panel, that information is stored in the corresponding dialog variables.

**TBDISPL** Displays a table. The TBDISPL service combines column headings from panel definitions with information stored in ISPF tables. It then displays selected columns from selected rows in the table, and allows the user to scroll through the information. (See "Table Services" below.)

**SETMSG** Saves a message to be displayed on the next panel written by ISPF to the terminal. If any variables are defined in the message, their values are substituted at the time of the SETMSG call. The completed message is saved in a message area for the application. When the next panel is displayed, the message is retrieved from the save area and displayed on that panel.

## Table Services

Table services allow sets of dialog variables to be created, maintained, and accessed in a convenient form. A table is a two-dimensional array of information in which each column corresponds to a dialog variable, and each row contains a set of values for these variables.

An example of a table is shown in Figure 9. In this table, the variables that define the columns are:

EMPSER - Employee Serial Number  
LNAME - Last Name  
FNAME - First Name  
'I - Middle Initial  
PHA - Home Phone: Area Code  
PHNUM - Home Phone: Local Number

One or more columns may be specified as keys for accessing the table. In the example, the EMPSER column, or both EMPSER and LNAME, may be defined as the keys. The information in the table may then be accessed either by the variable values in those keys, or merely row-by-row by a current-row pointer.

---

<u>EMPSER</u>	<u>LNAME</u>	<u>FNAME</u>	<u>I</u>	<u>PHA</u>	<u>PHNUM</u>
598304	Robertson	Richard	R	301	840-1224
172397	Smith	Susan	S	301	547-8465
813058	Russell	Charles	I	202	338-9557
395733	Adams	John	Q	202	477-1776
502774	Caruso	Vincent	E	914	294-1168
512484	Jones	Howard	H	914	279-9911
407397	Ward	Samuel	B	914	436-1000

Figure 9. Sample Table

---

A table may be temporary or permanent. A temporary table is created in virtual storage and deleted upon completion of processing. A permanent table resides on a direct access storage device as a member of a table library.

### General Services

The following services operate on an entire table:

TBCREATE Creates a new table and opens it for processing.  
TBOPEN Opens an existing (permanent) table for processing.  
TBQUERY Obtains information about a table.  
TBSAVE Saves a permanent copy of a table without closing it.  
TBCLOSE Closes a table, and saves a permanent copy if the table was opened in write mode.  
TBEND Closes a table without saving it.

**TBERASE** Deletes a permanent table from the table library.

Temporary tables are created by **TBCREATE** (NOWRITE mode) and deleted by either **TBEND** or **TBCLOSE**.

A new permanent table is created in virtual storage by **TBCREATE** (WRITE mode). The table becomes permanent when it is stored on direct access storage by either **TBSAVE** or **TBCLOSE**.

An existing permanent table is opened and read into virtual storage by **TBOPEN**. If the table is to be updated (WRITE mode), the new copy is saved by either **TBSAVE** or **TBCLOSE**. If it is not to be updated (NOWRITE mode), the virtual storage copy is deleted by either **TBEND** or **TBCLOSE**.

### Row Services

The following services operate on one row of a table at a time:

**TBADD** Adds a new row to the table.

**TBDELETE** Deletes a row from the table.

**TBGET** Retrieves a row from the table.

**TBPUT** Updates an existing row in the table.

**TBMOD** Updates a row in the table if it exists; otherwise, adds a new row to the table.

**TBEXIST** Tests for the existence of a row.

**TBSCAN** Searches a table for a row that matches a list of "argument" variables, and retrieves that row.

**TBSARG** Establishes a new search argument for use with **TBSCAN**.

**TBTOP** Sets the current row pointer to the top (ahead of the first row).

**TBBOTTOM** Sets the current row pointer to the last row and transfers the variables in the last row to the variable pool.

**TBSKIP** Moves the current row pointer forward or backward by a specified number of rows, and then transfers the variables in that row to the variable pool.

**TBVCLEAR** Sets the dialog variables that correspond to variables in the table to null.

## File Tailoring Services

File tailoring services read skeleton files from a library and create current, tailored output that may be used to drive other functions. Typically, these services can take a skeleton file that contains a general job stream for batch execution, fill in variable values for a particular job, and generate a specific set of statements for executing that job.

The file tailoring services are:

<b>FTOPEN</b>	Prepares the file tailoring process, and specifies whether the temporary file is to be used for output.
<b>FTINCL</b>	Specifies the skeleton to be used, and starts the tailoring process.
<b>FTCLOSE</b>	Ends the tailoring process.
<b>FTERASE</b>	Erases (deletes) any output file that was created by file tailoring.

Skeleton files are read record by record. If any dialog variable names are found in a record, the current value of that variable is substituted for the variable name.

Control statements may also be included in a skeleton file. They provide the ability to:

- Set dialog variables
- Imbed other skeleton files
- Conditionally include records
- Iteratively process records using variables from each row of a table

## Variable Services

Variable services allow a function to define and use dialog variables. A dialog variable is a character string (a word, a value — any kind of data) that is referred to by a symbolic name. The variable can be from zero to 32,767 bytes long.

Dialog variables are the main means of communication between dialog functions and ISPF services. They may also be used to communicate between functions.

For example, dialog names may appear in panel, message, and skeleton definitions to permit communication with functions. A variable name in a panel definition corresponds exactly to the name of a dialog variable accessible to a function.

The following sequence, coded as an MVS CLIST, shows the ease with which data is communicated between a function and the DISPLAY service.

```
SET &AAA = 1  
ISPEXEC DISPLAY PANEL(XYZ)  
SET &CCC = &AAA + &BBB
```

Variable AAA is created by the CLIST simply by setting it to a value. The DISPLAY service is then called to display panel XYZ.

Assume that the panel definition for panel XYZ contains two symbolic parameters, named AAA and BBB, and they are defined as input (unprotected) fields. They might appear as follows:

```
Initial value ==> _AAA  
Increment     ==> _BBB
```

where the underscore ( \_ ) indicates the start of an input field, followed by the name of a variable.

When the panel is displayed, the first input field is initialized to 1, since it was set by the dialog function before calling the DISPLAY service. The second field is displayed as blank, since variable BBB was not set.

Now assume the user changes the value of the first field to 100, and enters 10 in the second field. When the user presses the ENTER key, the values are automatically stored. Variable AAA is updated with the new value, and variable BBB is created. The DISPLAY service then returns control to the function. The last statement in the CLIST creates a variable named CCC, which is set to the sum of AAA and BBB (110 in the example).

The variable services are:

VGET	Retrieves variables from a shared pool or profile
VPUT	Updates variables in a shared pool or profile
VDEFINE	Defines a function variable
VDELETE	Removes the definition of function variables
VCOPY	Creates a copy of a variable
VREPLACE	Replaces a variable with a copy
VRESET	Resets function variables

The first two services, VGET and VPUT, may be invoked from any function. The other variable services are for use by program modules only; that is, they are not applicable to functions coded in a command procedure (MVS TSO CLIST or CMS EXEC2).

Certain variable names, reserved for use by the system, provide information about the operating environment. Reserved names all begin with the letter "Z". Dialog developers should avoid names that begin with "Z" when choosing their dialog variable names.

## ISPF INVOCATION

ISPF is invoked by entering the command **ISPSTART** or a defined alias. Keyword parameters are used to specify the first menu to be displayed or the first dialog function to receive control. If no parameter is specified, a master application menu is displayed. A sample master application menu, distributed with ISPF, is shown in Figure 10. This menu may be customized by your installation to display the names and brief descriptions of your available applications.

---

```
                ISPF MASTER APPLICATION MENU
-----
OPTION ==> 1_
1  SAMPLE1   - Sample application 1
2  .         - (Description for option 2)
3  .         - (Description for option 3)
4  .         - (Description for option 4)
5  .         - (Description for option 5)
X  EXIT      - Terminate ISPF using list/log defaults
                USERID   - Z59JEH
                TIME     - 16:19
                TERMINAL - 3277
                PF KEYS  - 12
```

Enter END command to terminate ISPF.

Figure 10. ISPF Master Application Menu for MVS

---

The user selects the desired application by entering the number of that application on the option line and pressing ENTER; for example,

```
OPTION ==> 1
```

as shown, selects the first and only application shown in Figure 10 and results in the display of Figure 11. This sample primary option menu is also distributed with ISPF.

---

SAMPLE PRIMARY OPTION MENU

---

OPTION ==> \_

- 0 ISPF PARMS - Specify terminal and user parameters
- 1 COMMANDS - Create/change command table
- 2 . - (Description for option 2)
- 3 . - (Description for option 3)
- 4 . - (Description for option 4)
- 5 . - (Description for option 5)
- T TUTORIAL - Display information about this application
- X EXIT - Terminate ISPF using list/log defaults

USERID - Z59JEH  
TIME - 16:19  
TERMINAL - 3277  
PF KEYS - 12

Enter END command to terminate application.

Figure 11. ISPF Sample Primary Option Menu for MVS

---

## CHAPTER 3. PROGRAM DEVELOPMENT

The ISPF/Program Development Facility (PDF) Program Product is an ISPF dialog that is designed to increase programmer productivity in an interactive environment. It may be used with both structured and conventional programming techniques. PDF helps to simplify the most common programming tasks by relieving the programmer of many tedious and repetitive operations.

Although ISPF is only one program product, PDF consists of three separate program products, one for each host environment. All contain similar functions to the extent permitted by their operating environment. There are, however, differences in the individual panels. In this document, most of the illustrations are for the MVS PDF product.

### PDF DISPLAYS

A brief scenario of MVS PDF usage is contained in Appendix C. The scenario includes examples of various display panels that handle the interaction between PDF and the user (the programmer).

#### Types of Panels

Five basic types of panels are presented to the PDF user:

1. **Selection panels (menus)**, from which the user chooses from a list of options by typing a one-character code and pressing the ENTER key. The first PDF panel, the primary option menu, is an example.
2. **Data entry panels**, on which the user supplies parameters by filling in labelled fields. One of the features of PDF is that the user's last entry in many fields are pre-entered for him the next time the same panel is displayed. The browse entry panel is an example.
3. **Member selection lists**, which display a list of members in a particular programming library. A member is selected for processing by typing one character in front of the appropriate member name and pressing the ENTER key.
4. **Data display panels**, which display source code or output listings.
5. **Table display panels**, which display table data when in dialog test mode.

## Screen Size

All panels are formatted to fit on a 24-line by 80-character screen. For a 3278 Model 3 or 4, scrollable data is formatted to use the full depth of that screen (32 or 43 lines). On a 3278 Model 5, information is normally displayed in the 24-line by 80-character format. If the data is wider than 80 characters, browse and edit use the full width of 132 characters for their displays. All models of the 3279 are supported in compatibility mode.

## Panel Format

The first three lines of each panel are formatted as follows:

line 1	Title	Short message
line 2	Command/Option	Scroll
line 3	Long message	

The title area identifies the function being performed and, if appropriate, additional information about the library or member being processed. The short message area is used to indicate:

- Current line (browse) and column positions (browse and edit).
- Successful completion of some processing function.
- Error conditions (accompanied by an audible alarm if one is available on the display device).

The command/option area is used to enter a command or a selection option. To the right of this line, the scroll area shows the current scroll amount if scrolling is applicable. This field may be overtyped with a new scroll value if desired.

The long message line is used to display an explanation of error messages when requested. On selection panels and data entry panels, this line is usually blank. On member selection lists, it contains column headings, and on data displays it is used as part of the data.

## LIBRARY AND DATA SET ACCESS

PDF provides access to a particular data set or file by two basic methods. The first is through a library structure. The second is through the normal data set or file naming conventions associated with the particular system on which PDF is running.

### Library Structure

The library structure provided by ISPF in the MVS and VM/SP environments is called the ISPF library structure. In the VSE environment, ISPF provides access to both the VSE libraries and the ICCF libraries.

#### MVS and VM/SP Libraries

An ISPF library is a collection of blocks of source code or units of data, referred to as members. Usually, all members of a library contain the same type of information.

ISPF libraries are normally identified by project name, library name, and type, where:

- **Project name** is the common identifier for all libraries belonging to a particular project.
- **Library name** identifies the particular set of libraries, such as MASTER, TEST, or JOE.
- **Type** identifies the type of information contained in the library. Some type identifiers are dependent on the host operating system.

PDF uses a hierarchy of libraries to permit effective control of different versions during the programming development process, and to reduce contention in library usage. Figure 12 shows a sample 3-level hierarchy for a project. It consists of a set of master libraries, a set of test libraries, and three sets of individual user libraries identified by the user working on that library.

The figure identifies the contents of each library rather than the filetype (VM) or lowest level qualifier (MVS/TSO) because the actual identification required in these fields may depend on the operating system for some types of libraries.

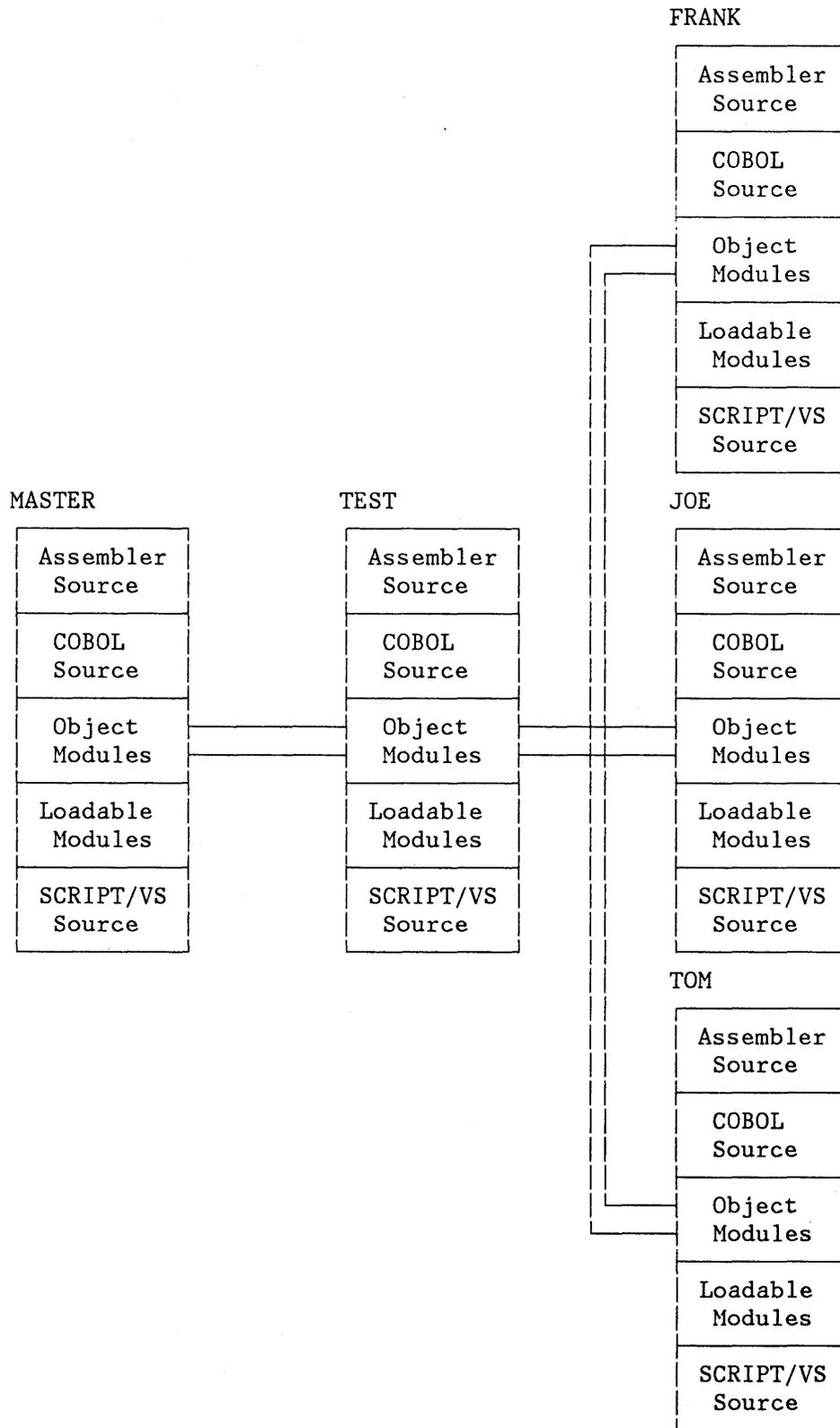


Figure 12. Hierarchy of ISPF Libraries

## VSE System Libraries

Under VSE, ISPF provides access to the VSE system libraries. The rules governing the contents of these libraries are as set forth for the VSE Program Product. For more information on VSE system libraries, refer to VSE/Advanced Functions, System Management Guide, SC33-6094.

PDF panels prompt you for each component of the VSE library identification as follows:

```
VSE LIBRARY:
LIBRARY    ==>
SUBLIBRARY ==>          (For type SL only)
MEMBER     ==>          (Blank for member selection list)
TYPE       ==>          (PL, RL, or SL)
```

where:

**library** is a 7-character library name

**sublibrary** is the 1-character sublibrary name. A sublibrary name is required only if the type is SL.

**member** is the optional 8-character member name

**type** is a 2-character type, as follows:

```
PL    procedure library
RL    relocatable library
SL    source library
```

## ICCF Libraries

Under VSE, ISPF also provides access to ICCF libraries. For more information on ICCF libraries, refer to VSE/ICCF Installation and Operations Reference.

PDF panels prompt you for each component of the ICCF library identification as follows:

```
ICCF LIBRARY:
  PRIMARY  ==>      CONNECT ==>      COMMON ==> (Yes or No)
  MEMBER   ==>                        (Blank for member selection list)
```

where:

**PRIMARY** is the number of your primary library. You can enter 'prim' for your primary library.

**CONNECT** is the number of your connect library. You can enter 'conn' for your connect library.

**COMMON** indicates whether you want to include the common library in the search sequence.

```
yes  include the common library
no   exclude the common library
```

**MEMBER** is an optional 8-character member name.

If you leave the member name field blank, a member list is displayed.

## Conventional Data Set and File Naming Conventions

You may also refer to individual sequential data sets or files by using specific naming conventions appropriate to your environment. Under MVS you may also refer to members of a partitioned data set, and under VM/SP you may refer to members of a MACLIB or TXTLIB.

All of the following examples show representative areas for the entry of the data set or file name. The exact wording and arrangement may differ slightly from one panel to the next. In some examples, extra blanks have been removed to allow the fields to be shown on one line.

### MVS Data Set Names

Under MVS, PDF permits access to data sets using standard MVS data set naming conventions in an entry area similar to the following:

```
OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ==>
  VOLUME SERIAL ==>      (If not cataloged, required for option "C")
```

### CMS File Names

Under VM/SP, PDF permits access to files using standard CMS file naming conventions in an entry area similar to the following:

```
CMS FILE:
  FILE ID: ==>
  MEMBER: ==>                (FOR MACLIB OR TXTLIB)
  IF NOT LINKED, SPECIFY:
  OWNER'S ID ==>            DEVICE ADDR. ==>      LINK ACCESS MODE ==>
                           UPDATE PASSWORD. ==>

  READ PASSWORD ==>
```

### VSE Data Set Names

Under VSE, PDF permits access to sequential data sets using VSE data set naming conventions in an entry area similar to the following:

```
SEQUENTIAL DATA SET:
  FILE NAME ==>                ( 1 to 7 characters)
  RECFORM ==>                RECSIZE ==>          BLKSIZE ==>
  SYS NUMBER ==>                (If not on EXTENT or /FILE stmt)
  VOL SERIAL ==>                (If not on EXTENT or /FILE stmt)
```

## PDF INVOCATION

The ISPF/Program Development Facility is invoked by entering the command "PDF", or a defined alias. The first display is the primary option menu (Figure 13) that lists the available functions for the user. The user may select an option by typing its number or character identifier in the option field and pressing the ENTER key; for example,

```
OPTION ==> 2
```

to select the edit option.

---

```
ISPF/PDF PRIMARY OPTION MENU
```

---

```
OPTION ==> 2_
```

0	ISPF PARMS	- Specify terminal and user parameters	USERID	- Z59JEH
1	BROWSE	- Display source data or output listings	TIME	- 15:21
2	EDIT	- Create or change source data	TERMINAL	- 3277
3	UTILITIES	- Perform utility functions	PF KEYS	- 12
4	FOREGROUND	- Invoke language processors in foreground		
5	BATCH	- Submit job for language processing		
6	COMMAND	- Enter TSO command or CLIST		
7	DIALOG TEST	- Perform dialog testing		
C	CHANGES	- Display summary of changes for this release		
T	TUTORIAL	- Display information about ISPF/PDF		
X	EXIT	- Terminate ISPF using log and list defaults		

Enter END command to terminate ISPF.

Figure 13. PDF Primary Option Menu for MVS

---

(Although the options provided are functionally identical, there are slight differences in the panels from one system to another. The panels shown in this publication are from PDF for MVS.)

For options that have lower level menus (options 0, 3, 4, 5, and 7), the user may bypass the second menu by typing two numbers, separated by a decimal point, on the primary option menu. For example, entering "3.1" on the primary option menu has the same effect as entering "3" on the primary option menu and "1" on the secondary menu.

On initial entry, the user may also bypass the primary (and secondary) menus by entering an initial option as a parameter to the PDF command, for example:

```
PDF 2    - to go directly to edit
PDF 3.1  - to go directly to utilities suboption 1
```

## PDF OPTIONS

The selections on the **primary option menu** (Figure 13) allow the user to choose from the program or dialog development functions described below. These are general descriptions of the rfunctions, with some notations of the differences between operating environments. You should refer to the appropriate Reference manual for your environment for the exact description of each function.

### Option 0 - ISPF PARMS

Selecting **ISPF PARMS** allows the user to display and change a variety of ISPF parameters at any time during the session. Changes remain in effect until the user changes the parameter again, and are retained across sessions.

The ISPF PARMS option allows the user to specify:

- Terminal characteristics
- Console (VM/SP only), log, and list defaults
- Program function (PF) key assignments

### Option 1 - BROWSE

Selecting **BROWSE** allows the user to display source data or output listings. Browse is primarily intended for viewing large files (data sets) such as compiler listings or dumps.

**VSE Note:** Browse is not used to display output listings.

During browse, <sup>(up-down-left-right)</sup>4-way scrolling is available through commands or PF keys. FIND and LOCATE commands may also be used to scroll to a particular character string, line number, or label.

### Option 2 - EDIT

Selecting **EDIT** allows the user to create or change source data, such as program code, test data, or documentation.

**VSE Note:** Edit cannot process ICCF members that contain ICCF commands (CMDS).

Unlike browse, edit reads the selected member (or entire sequential file) into virtual storage, and retains it there during edit operations.

Under edit, 4-way scrolling is available through commands or PF keys. FIND and LOCATE commands may also be used to scroll to a particular character string or line number.

To modify one or more lines of data, the user simply moves the cursor to the desired location and enters the new information by overtyping the existing lines.

Lines may be deleted, inserted, shifted left or right (for indentation changes), duplicated, or rearranged by overtyping the line-number fields with "line commands" consisting of one or more characters. Several line commands as well as data modifications may be typed before pressing the ENTER key.

For general edit operations, "primary commands" may be entered in the command input field. Some of the frequently used primary commands are:

<b>SAVE</b>	Save the data
<b>CANCEL</b>	End the edit function without saving
<b>LOCATE</b>	Scroll to the specified line number
<b>FIND</b>	Scroll to a specified character string
<b>CHANGE</b>	Replace a specified character string with another string
<b>COPY</b>	Duplicate data from another member or file
<b>MOVE</b>	Duplicate data and then delete the member or file

Some of the special features provided by the PDF editor include:

- Column Identification Line — A special line may be displayed anywhere in the data area of the screen to determine column positions.
- Insert Mask — Information may be pre-entered on inserted lines by defining a mask.
- Excluded lines — Designated lines of data may be temporarily excluded from the display to facilitate visual verification of program structure.
- Tabs — Three types of tabs may be defined:
  - Software tabs, which control cursor repositioning
  - Hardware tabs, which cause attribute bytes to be inserted
  - Logical tabs, which cause the PDF to reposition data
- Bounds — The column boundaries, which limit the operation of the shift line commands and the FIND and CHANGE primary commands, may be changed at any time.
- Null Characters — Trailing blanks in each data field may be displayed as null characters to simplify the use of the 3270 INSERT key.

- Hexadecimal Display — Data may be displayed and updated using hexadecimal representation.
- Document Preparation Support — Document preparation is aided by commands that simplify:
  - Bulk entry of text
  - Insertion of new material
  - Rearrangment of text within paragraphs
- Edit Recovery — Under user option, the PDF editor automatically maintains a history of change activity. If there is a system failure, the user can recover the edit session up to the point of failure.
- Models — Predefined sets of statements may be copied into the program being edited. These model statements may then be modified for a particular application.

### Option 3 - UTILITIES

Selecting **UTILITIES** allows the user to choose from a variety of system utilities, shown on a secondary option menu. Some of the functions that can be performed with these utilities are:

- Print, rename, or delete library members (all environments) or entire files (MVS and VM/SP only)
- Allocate files (MVS only)
- Move or copy files
- Reset ISPF library statistics (MVS and VM/SP only)

**VSE Note:** For VSE, the following restrictions apply:

- You cannot move or copy to or from a core image library (CL).
- The utilities cannot write data to ICCF members that contain ICCF commands (CMDS).

Other utility functions vary from one host system to another. Consult the appropriate PDF Reference manual for details.

### Option 4 - FOREGROUND (MVS and VM/SP only)

Selecting **FOREGROUND** allows the user to execute language processing programs in the foreground. Certain output of the language processors, such as error messages, prompts, and return codes, are displayed to the user as an indication of the task's completion. In addition, the output listing (if any) is displayed in browse mode. Some typical processors are:

- Assembler
- COBOL (and COBOL Interactive Debug)
- FORTRAN (and FORTRAN Interactive Debug)
- Link edit
- PL/I (Checkout or Optimizer)
- PASCAL/VS Installed User Program
- SCRIPT/VS Formatter of the Document Composition Facility

Again, as with the utilities, foreground functions vary depending on the host system. The appropriate PDF Reference manual contains the necessary details.

## Option 5 - BATCH

Selecting **BATCH** allows the user to generate and submit appropriate host system job statements and command streams for batch execution of applicable language processors and programs.

**VSE Note:** Any data sets referred to by a batch job must have supporting labels in the system label area, and the source chain for the language processor cannot include ICCF libraries.

## Option 6 - COMMAND

Selecting **COMMAND** allows the user to enter appropriate host system commands (including MVS TSO CLISTS or VM EXECs) during execution of PDF. Typically, the command and any resulting output is displayed to the user as though the command had been issued in the normal host environment.

## Option 7 - DIALOG TEST

Selecting **DIALOG TEST** allows the developer to test applications that are to be run under the ISPF dialog manager before installing them in the production environment. There are three general points at which a developer may wish to use test aids:

- The developer, before beginning to run a dialog or at a breakpoint during testing, may invoke dialog functions, display panels, display and manipulate variable data, display and change table data, and call ISPF services.
- The developer may stop execution of the dialog at any call to an ISPF service by establishing a breakpoint. At a breakpoint, dialog data (variable data, table data, return code from a service, etc.) may be displayed and changed.
- The developer may trace the flow of the dialog's calls to ISPF services and the usage of dialog variables. Trace information is placed in the ISPF log.

In particular, dialog test contains facilities to permit:

- Invoking functions and testing selection panels
- Displaying and testing panels, and, optionally, messages
- Displaying and setting variable values, and copying variable values to other variables and other variable pools
- Displaying and modifying table rows, and displaying table status and table structure
- Browsing the ISPF log
- Invoking any ISPF service (except CONTROL)

- Tracing applications and function calls to ISPF services, and tracing variable usage:
  - For function traces, the developer identifies both the functions and the services for which tracing is requested.
  - For variable traces, the developer identifies the type of trace request by specifying the variable name, variable pool, type of operation (GET, PUT, or CHG), and function for which tracing is requested.
- Managing breakpoints at any calls to ISPF services (at a breakpoint, any of the above facilities may be used)

### Option C - CHANGES (MVS and VM/SP only)

Selecting **CHANGES** allows the user to review a description of the PDF changes from the predecessor program product, the System Productivity Facility (SPF).

### Option T - TUTORIAL

Selecting **TUTORIAL** allows the user to obtain immediate online instruction in the use of PDF. The tutorial may be viewed sequentially from beginning to end, or randomly by selecting topics from the table of contents or index. The tutorial may also be entered from other PDF options by using the **HELP** command or PF key.

### Option X - EXIT

Selecting **EXIT** allows the user to end an ISPF session using the defaults specified under ISPF PARMS.

The user may also leave PDF by using the **END** command or PF key. This causes a termination panel to be shown so that the user may specify the processing to be performed on the files created during this PDF session.

## CHAPTER 4. PLANNING

This chapter identifies system requirements and customer responsibilities. It also includes a brief description of related documents that are planned for availability concurrent with product release.

### PROGRAMMING REQUIREMENTS

This section describes the programming requirements for each of the operating systems supported. Keep in mind that ISPF is a prerequisite for PDF.

#### MVS Requirements

In the MVS environment, ISPF operates as a TSO command processor under the Time Sharing Option of MVS. The BPAM and BSAM access methods are required by ISPF for reading and writing data sets, and TSO/TCAM or TSO/VTAM are required for terminal communication.

One of the following teleprocessing access methods is required:

TCAM 10	(available with MVS)
ACF/TCAM Version 2, Release 2 or later	5735-RC3
VTAM 2	(available with MVS)
ACF/VTAM Release 2 or later	5735-RC2

**Note:** ACF/TCAM or ACF/VTAM is required with Models 3, 4, or 5 display terminals (with screen sizes larger than 24 by 80).

Internal interfaces to the IBM programs listed below are provided. These programs are not required to operate ISPF. However, if an IBM 3284, 3286, 3287, 3288, or 3289 printer is used for ISPF output, the appropriate DSPRINT command processor must be installed on the system. Refer to the chapter on utilities in ISPF/PDF for MVS Reference for further information about these interfaces.

OS/VS2 MVS 3270 Extended Display Support - Session Manager,	Release 2	5740-XE2
TSO/TCAM Command Processor "DSPRINT"		5798-AYF
TSO/VTAM Data Set Print (DSPRINT)		5798-CPF
TSO/VS2 Programming Control Facility (PCF)		5798-BBJ
TSO Programming Control Facility - II (PCF2)		5798-CLW

The ISPF/Program Development Facility provides interfaces to the following IBM processing programs for foreground and batch execution:

VS2 Assembler	(available with MVS)
Linkage Editor	(available with MVS)
TSO Assembler Prompter (foreground only)	5734-CP2
COBOL Compiler and Library	5740-CB1
TSO COBOL Prompter (foreground only)	5734-CP1
COBOL Interactive Debug (foreground only)	5734-CB4
FORTRAN IV G1 Compiler	5734-F02
TSO FORTRAN Prompter (foreground only)	5734-CP3
FORTRAN Interactive Debug (foreground only)	5734-F05
PL/I Checkout Compiler	5734-PL2
PL/I Optimizing Compiler	5734-PL1
PASCAL/VS Compiler	5796-PNQ
Document Composition Facility (SCRIPT/VS) (foreground only) with the Foreground Environment Feature	5748-XX9

The appropriate processing programs and TSO prompters must be installed to use the foreground and batch options. All the program-numbered products listed above can be ordered separately under IBM licensing agreements.

## VM/SP Requirements

In the VM/SP environment, ISPF operates as a CMS command under the VM/System Product (VM/SP), 5664-167, which must be installed.

The operation of ISPF also requires a disconnected virtual machine (ISPVM) that communicates with each user's machine through the VM Communication Facility (VMCF). If ISPF is to be used with SNA 3270 displays, the following IBM program product must be installed:

Virtual Machine/VTAM Communications Network Application (VM/VCNA)	5735-RC5
--	----------

The ISPF/Program Development Facility provides interfaces to the following IBM programs for foreground and batch execution:

VM/370 Assembler	(available with VM)
Linkage Editor	(available with VM)
COBOL Compiler and Library	5740-CB1
COBOL Interactive Debug (foreground only)	5734-CB4
FORTRAN IV G1 Compiler	5734-F02
FORTRAN Interactive Debug (foreground only)	5734-F05
PL/I Checkout Compiler	5734-PL2
PL/I Optimizing Compiler	5734-PL1
PASCAL/VS Compiler	5796-PNQ
Document Composition Facility (SCRIPT/VS) (foreground only) with the Foreground Environment Feature	5748-XX9

Internal interfaces are provided to the following IBM program product.

RSCS Networking

5748-XP1

This program is not required to operate PDF, but it must be installed if spool output is to be sent to another node on the network. Refer to the chapter on utilities in ISPF/PDF for VM Reference for further information about this interface.

The appropriate processing programs must be installed to use the foreground and batch options. All the program-numbered products listed above can be ordered separately under IBM licensing agreements.

## VSE/ICCF Requirements

In the VSE environment, ISPF may be invoked in a VSE batch partition or through an ICCF procedure as an application program in an ICCF interactive partition.

**Note:** ISPF may **not** be invoked as an ICCF command.

The following VSE support programs are required:

Advanced Function Release 3.5	5746-XE8
(or)	
VSE System IPO/E Version 1 Release 4 Level 0	5750-AAP
VSE/POWER Version 2 Release 1.5	5746-XE3
VSE/ICCF Release 3.5 in either of the following telecommunications environments:	
ICCF TTF	(received with ICCF)
CICS/VS (Version 1 Release 5 or later)	5746-XX3

Any telecommunication access method supported by these environments may be utilized.

The ISPF/Program Development Facility provides interfaces to the following IBM programs for batch execution:

System Assembler	(available with VSE)
Linkage Editor	(available with VSE)
Full ANS Cobol Compiler and Libraries	5746-CB1
VS/FORTRAN Compiler Release 2	5748-FO3
PL/I Optimizing Compiler and Libraries	5736-PL3
RPG II Compiler	5746-RG1

The appropriate processing programs must be installed to use the batch option.

## MACHINE REQUIREMENTS

The machine requirements for each environment are the same as those for the operating system under which ISPF is running.

On an MVS/TSO system, most of the ISPF program modules normally reside in the pageable link pack area (PLPA).

On a VM/SP system, if ISPF is run on a Model 4331 with one megabyte of main storage, VM/SP must be generated with the "small CP" option. Refer to IBM Virtual Machine/System Product: Planning and System Generation Guide, SC19-6201, for a description of the restrictions imposed when using this option.

The following table lists:

- The approximate disk storage space required to contain the ISPF and PDF libraries.

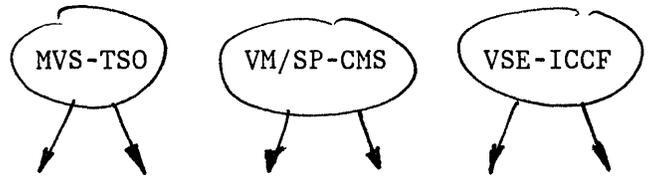
Additional space must be available to contain the panels, messages, function load modules, file skeletons, tables, and other data required by user dialogs that are to be executed under ISPF.

- The minimum region (MVS), virtual machine (VM/SP), or pseudo partition (VSE) size required for ISPF and PDF operation. The second (\*) value shown is the recommended size.
- **For MVS:** the size of the pageable link pack area (PLPA) that may be used for ISPF and PDF program residency on MVS systems. The use of PLPA or SVA is recommended for performance reasons, but not required. Most of the performance benefits may be achieved by using the second value (\*) shown in that entry in the table.

**For VM/SP:** the size of the discontinuous shared segment (DCSS) required for ISPF and PDF program residency on VM/SP systems.

**For VSE:** the size of the shared virtual area (SVA) required for ISPF and PDF residency on VSE systems. The first value shown is the minimum required area. The second value is the size needed for all components of the product. The use of SVA beyond the minimum requirement is recommended for performance reasons but not required. Most of the performance benefits may be achieved by using the third (\*) value shown in this entry in the table.

**Note:** In the table, 'NA' stands for 'not applicable'.



	ISPF	PDF	ISPF	PDF	ISPF	PDF
Cylinders of disk storage (3330) <sup>1 2</sup>	8	19	15	30	10	10
Size in bytes, of minimum: region (MVS) virtual machine (VM/SP) pseudo partition (VSE) (* = recommended)	512K 2 M*	NA <sup>3</sup>	512K 2 M*	NA <sup>3</sup>	128K 256K*	198K
Size in bytes, <sup>2</sup> of: PLPA (MVS) DCSS (VM/SP) SVA (VSE) (* = see description)	325K 280K*	330K 205K*	384K	384K	50 388K 364K*	0 360K 260K*

## TERMINAL REQUIREMENTS

ISPF requires one or more of the following IBM display stations:

- 3275 Model 2
- 3275 Model 12 (using VM/VCNA for VM/SP)
- 3276 Model 2, 3, or 4
- 3276 Model 12, 13, or 14 (using VM/VCNA for VM/SP systems)
- 3277 Model 2 (local or remote)
- 3278 Model 2, 3, 4, or 5 (local or remote)
- 3279 Model 2A, 3A, 2B, or 3B (local or remote)  
(2B and 3B only when operated in 4-color compatibility mode)

The following character sets may be used with 3275 and 3277 display stations:

- EBCDIC
- ASCII (MVS only)
- APL
- Katakana

<sup>1</sup> For other device types, use an equivalent amount of space.  
<sup>2</sup> For these entries, add the ISPF and PDF figures if PDF is installed.  
<sup>3</sup> The minimum region or virtual machine size is the same with or without PDF.

The following character sets may be used with 3276, 3278, and 3279 display stations:

- EBCDIC
- ASCII (MVS only)
- APL
- Text
- Katakana
- Canadian-French

The following features or optional devices may be installed but are not required:

- Audible alarm (Feature 1090)
- IBM 3284, 3286, 3287, 3288, and 3289 printers (MVS/TSO only)
- 3277 dual-case character set (RPQ 8K0366)

Installation of the audible alarm feature enhances usability by sounding an alarm whenever a warning or error message is displayed.

**MVS Note:** In the MVS/TSO environment, use of an IBM 3284, 3286, 3287, 3288, and 3289 printer requires installation of the appropriate DSPRINT command processor.

Regardless of the width of the terminal screen, dialogs may not use lines greater than 80 characters. Therefore, for the 3278-5 terminal, panel definitions are restricted to 80 characters in width but EDIT and BROWSE are capable of using the full 132-character width of the screen.

## CUSTOMER RESPONSIBILITIES

The installation of ISPF requires a properly configured system, with appropriate terminals and other devices, as required for the desired operating system.

ISPF and PDF provide no security or data integrity functions. It is the customer's responsibility to use any existing programs to provide these functions.

Installation procedures are described in ISPF Installation and Customization.

Appropriate processing programs (and TSO prompters for MVS) must be installed to use the ISPF foreground and batch options. The Document Composition Facility must be installed to use the SCRIPT/VS utility. Neither the foreground option nor SCRIPT/VS are available under VSE.

On an MVS system, if ISPF is to be used with TSO/TCAM, the standard TSO/TCAM message handler must be reassembled to incorporate minor modifications. These modifications are designed to eliminate interference between the full-screen I/O operations used by ISPF and the line-oriented I/O operations used by TSO. No other changes are required to TSO or TCAM.

5034-2088

A dialog developer must be familiar with the base operating system and appropriate programming languages, and should review ISPF Dialog Management Services. A user of the ISPF/Program Development Facility must be familiar with the base operating system and should review ISPF/PDF Reference.

## MIGRATION

Dialogs that run under the previous SPF Program Product (under MVS and VM/SP) can also be run under the ISPF Program Product. However, because of differences in the products, certain modifications may have to be made to the dialogs. To migrate existing dialogs to ISPF, you must consider both the differences between the products and the environment in which the dialogs are to be run.

The following differences between SPF and ISPF must be taken into account when migrating SPF dialogs to ISPF. These considerations also apply when converting existing MVS or VM/SP dialogs to run in any of the host environments (MVS, VM/SP, or VSE).

- Changes to the default variable pool setting of the VGET/VPUT services may cause a different variable value to be obtained or set in ISPF than in SPF.
- There is an additional VGET/VPUT return code in ISPF.
- SPF dialogs that use the edit and browse interfaces require that the PDF Program Product be installed in order for those interfaces to execute in the ISPF environment.
- Command tables should be reviewed if the SPF dialog is to be invoked from a new ISPF dialog that uses command tables. The new dialog may have defined commands in the command table that conflict with commands that were processed by the SPF dialog panels or function routines.
- SPF system profile data must be moved from the data set defined by the ISPPARM ddname to the ISPF profile table data set defined by ddname ISPPROF. Facilities are provided within ISPF to perform this relocation.
- Primary option menus are no longer determined by the NEWAPPL keyword on a select service request. To indicate that the selection panel should be displayed when the RETURN key or command is used at a lower point in the application, the variable &ZPRIM must be set to YES in the )INIT section of the selection panel.
- Although not required, it is recommended that panels using the SPF OPT and SEL variables be converted to use the ISPF ZCMD and ZSEL variables, respectively. Also, help/tutorial panels using the UP and CONT variables should be converted to use the ZUP and ZCONT variables, respectively.

- Although not required, it is recommended that the ISPF variable &ZIND be set to YES in the )PROC section of the tutorial index pages. This cause a higher level tutorial text page to be displayed when the user enters the UP command in response to a tutorial page that was displayed as a result of a selection from the index. If &ZIND is not set to YES in the index page, the tutorial function continues to work as it did in SPF and redisplay the index page in response to the UP command.

## CONVERTING DIALOGS FROM ONE ENVIRONMENT TO ANOTHER

If you want to convert existing MVS or VM/SP dialogs to run under ISPF in a different enviroment, you must consider all of the changes and enhancements listed above and also consider the following:

- Dialog functions written in an command procedure such as TSO CLIST or CMS EXEC must be converted to a command procedure or a programming language supported by ISPF in the new environment.

**VSE Note:** Since no command procedure is supported by ISPF under VSE, the function must be rewritten in a supported programming language.

- Any commands or services invoked by the original dialog must be converted to an equivalent command or service in the new environment.
- Any file skeletons that generate JCL (JECL) or commands must be converted to the appropriate JCL (JECL) or commands for the new environment.
- All panels, messages, skeletons, tables, etc., must be moved to the appropriate libraries in the new environment.

## SUPPORTING PUBLICATIONS

The following publications are available with ISPF:

- Dialog Management Services (SC34-2088) - Provides detailed information on the use of dialog management services in the development of interactive applications. One manual covers all host environments.
- Dialog Management Services Examples (SC34-2085) - Supplements the above manual; provides examples of the use of dialog management services in the development of interactive applications. One manual covers all host environments.
- Installation and Customization (SC34-2084 for MVS, SC34-2083 for VM/SP, SC34-2080 for VSE) - Provides detailed information on how to install and custom tailor ISPF and ISPF/PDF. There is a separate manual for each host environment.

The following additional publication is available with ISPF/PDF:

- Reference (SC34-2089 for MVS, SC34-2090 for VM/SP, SC34-2079 for VSE) - Provides detailed information on how to use the Program Development Facility. There is a separate manual for each host environment.



## APPENDIX A. CHANGES FROM SYSTEM PRODUCTIVITY FACILITY (SPF)

In MVS and VM/SP, the dialog management functions of ISPF and the program development functions of PDF were previously combined in the predecessor program product on MVS and VM/SP, System Productivity Facility (SPF). These changes will also be of interest to a VSE user who is considering migrating applications created for the predecessor program product on either MVS or VM/SP. The changes from SPF are summarized in the two sections that follow.

### DIALOG MANAGEMENT CHANGES

Significant changes included in ISPF are listed below:

- Name-List Syntax Enhancement

A parameter in a CALL ISPLINK no longer has to be enclosed in parentheses when there is only one variable in the list.

- NEWAPPL Keyword

The SELECT service and the ISPF command now allow specification of an optional application id on the NEWAPPL keyword. For example:

```
SELECT PANEL(ABCTOP) NEWAPPL(XXXX)
```

where XXXX is the application id.

The application id, which must be unique to the installation, may be up to four characters. It serves as a prefix to identify the user profile and/or command table associated with the application, as follows:

```
XXXXPROF - User Profile  
XXXXCMDS - Command Table
```

- NOCHECK Keyword

A NOCHECK keyword may be specified with either the CMD or PGM keyword. With the NOCHECK keyword, the dialog function (command or program) is invoked even if the user specifies a chain of options.

- Variable Services

- Profile variables are included in standard search sequence.
- Profile variables are now stored by application and by user instead of just by user.

- A name-list interface is provided for the VCOPY and VREPLACE services.
- User exits are provided for variable access and conversion.
- Command Tables
 

A facility is included that allows the dialog manager to intercept user-entered commands and take appropriate action.
- Program Function (PF) Keys
 

Program function keys are no longer required for operation of ISPF. All functions for which PF keys were previously required may now be entered in the command field of any display.
- New commands:
 

**KEYS** provides direct entry to option 0.3 without terminating the current function.

**PANELID** initiates or terminates the display of panel identification at the left of the top line of each panel.

**TSO** (MVS only) allows a TSO command or CLIST to be entered from any panel.

**CP, CMS** (VM/SP only) allows CP or CMS commands or EXECs to be entered from any panel.

**ICCF** (VSE only) allows ICCF commands to be entered from any panel.
- Light Pen and Cursor Select
 

ISPF now detects fields on a panel by means of either a light pen or the cursor select key.
- Set Next Selection Menu
 

A new system variable, ZPARENT, allows the developer to indicate the next panel to be displayed when the user enters the end command or when a function completes operation. This alters the normal reverse sequence established during the original menu selection sequence.
- Display Services
  - A new "set next message" function provides the ability to specify a message to be displayed with the next panel that is written by ISPF to the terminal.
  - The table display services have been enhanced with the following features:

- Multiple line selection or modification
- New MODEL header statement keywords
- Variable names on model lines
- Multiple model lines
- Explicit cursor placement within scrollable data

Enhancements to the panel display service include:

- Extended verification functions
  - "Z" variables as field name placeholders (see "Long Variable Names" below)
  - Truncation remainder function
  - Alternative locations for command and message fields
  - Profile variable support
  - New attribute keywords — SKIP and ATTN (See "Light Pen and Cursor Select" above.)
  - KANA keyword for Japanese keyboards
- Help panels may now contain variables so that dialog information (including information entered by the user) may be displayed on the help panel.

- **Dynamic Selection Panel Formatting**

New selections may be added to existing menus. This facility may be run in either foreground or batch.

- **Dialog Control of Table Service Output**

An application may direct table output to a table output library other than the library specified on the table output ISPTABL DD, FILEDEF, or ISPDEF statement.

- **Dialog Control of File Tailoring Output**

An application may direct file tailoring output to an output library other than the library specified on the file tailoring output ISPFIL DD, FILEDEF, or ISPDEF statement.

In addition, a new keyword, NOREPL, is provided to indicate that an existing member in the file tailoring output library is not to be overlaid by the current FTCLOSE service request.

- Control Services
  - An option has been added to allow display output without unlocking the keyboard.
  - Split-screen mode may be enabled or disabled by a function.
  - The display environment may be saved and restored.

- Batch Execution of Dialog Services

Non-interactive dialogs may be executed in the background.

- IPF Table Migration Utility (VM/SP and VSE only)

A utility is provided to assist with the conversion of tables.

- Long Variable Names

Variable names that are longer than the screen field may now be assigned in a panel definition. The single character "Z" is used in the body section of the panel definition, and the actual name is assigned to that field in the initialization section of the panel definition. The assignment is made through the new control variable (.ZVARS) and a namelist.

## PROGRAM DEVELOPMENT FACILITY CHANGES

Significant changes included in PDF are listed below. Some of these changes result from changes to ISPF.

- Improved Device Support (obtained through ISPF changes)
  - APL and TEXT keyboards may both be used. However, although the terminals are capable of displaying a mixture of characters from both keyboards at one time, only those characters appropriate for the keyboard in use are displayed by ISPF.
  - Program function (PF) keys are no longer required, but are recommended for ease of use. All PF key functions may be entered as commands on any panel.
- Use of Commands (obtained through ISPF changes)
  - All panels now have a command entry field, which is also used for option entry on selection panels.
  - Multiple commands, separated by a delimiter, may now be entered in a command field. The default delimiter is a semicolon, which may be changed using option (0.1).

- PF keys may now be used to simulate command entry. As before, PF keys are defined using option (0.3). A PF key may be equated to a string of commands (separated by delimiters).
- All previous PF key functions (HELP, SPLIT, END, etc.) may now be entered as commands.
- The FIND and CHANGE PF key functions have been renamed RFIND (repeat find) and RCHANGE (repeat change) to avoid confusion with the FIND and CHANGE commands passed through to browse and edit.
- Specify PF Keys - Option 0.3

This facility is provided by ISPF, and may be invoked from PDF option (0.3) or from ISPF through the sample primary option menu.

- The PF key specification panel now includes fields for terminal type and number of PF keys. These must be verified (or changed) before proceeding to define PF keys.
- For 24-key terminals, two panels are now displayed: The first for specification of "primary" keys (13-24), and the second for specification of "alternate" keys (1-12).
- Edit - Option 2
  - Models of all dialog elements are provided with the PDF for use in creating new interactive applications.
  - A new MODEL command is provided to bring models into the data set being edited.
  - The PRINT ON/OFF command has been renamed AUTOLIST ON/OFF.
- Utilities - Option 3
  - Option (3.7) provides for display of indexed VTOCs generated by the Data Facility/Device Support for OS/VS2 MVS Program Product, Program Number 5740-AM7 (MVS and VM/SP only).
  - The SCRIPT/VS interface (previously utility 3.9) is now a foreground option (4.9) (MVS and VM/SP only).
  - A new utility (3.9) allows generation and modification of application command tables (used in dialog development). This facility is provided by ISPF and may also be invoked from the sample primary options menu.
  - The convert menus and messages capabilities (previously options 7.4 and 7.5) are now in utility 3.10. (Not applicable to VSE since there was no previous VSE product.)

- Foreground (not applicable to VSE) and Batch - Options 4 and 5
  - Some suboptions have been renumbered to group the compilers and interactive debug programs.
  - The SCRIPT/VS interface is now a foreground option (4.9) (MVS and VM/SP only).
  - Foreground options that generate listings now result in an automatic entry to browse mode, followed by display of a print option panel (MVS and VM/SP only).
  - The following are now supported from foreground (not applicable to VSE) and batch panels:
    - Member selection list (if member name is left blank)
    - "Other" data set name (MVS only)
    - Additional concatenated SYSLIBs (MVS only)
- Command (Option 6)
  - The ISPEXEC command may no longer be entered from option 6, nor from other panels by using the TSO, CP, or CMS command. Dialog services may be entered through the dialog test facility (option 7.6).
- Dialog Test (Option 7)
 

Option 7 has been significantly enhanced and expanded with new test capabilities. Refer to "Selection 7 - Dialog Test" for details.

The new option 7 menu is a primary option menu, meaning that the RETURN command (or key) stops at that menu. From a dialog test function, two RETURN commands (or RETURN followed by END) are required to go all the way back to the PDF primary option menu.
- Session Manager Interface (MVS only)
  - Entry of Session Manager mode from options 4 and 6 is now controlled through a panel field, rather than by the scroll keys.

## APPENDIX B. EXAMPLE OF AN ISPF DIALOG

This example illustrates how some parts of an "employee records" application might be implemented. This sample demonstrates how this application might be written in PL/I. It has been extracted from ISPF Dialog Management Services Examples. Some editing of the panel definitions and panel displays was performed to fit the format of this document. Refer to Examples for the accurate format and to see how this application might be coded in other languages.

### OVERVIEW

The employee records dialog provides facilities for creating, maintaining, and displaying a table of employee information: serial number, name, address, and phone number. Information for each employee is maintained in a row of an ISPF table. The employee serial number is the KEY field for the table.

### DIALOG COMPONENTS

Dialog components that implement the employee records application are:

- A primary option menu (EMPL) that receives control when the application is invoked.
- A function (EMPLFN) that receives control from the primary option menu (EMPL) and performs the add, update, or delete processing requested for the employee record.
- Two data entry panel definitions (EMPLA and EMPLB) that are displayed by the function EMPLFN. Data entry panel EMPLA is used to obtain the employee serial number for which information is to be entered on the next data entry panel displayed (EMPLB). EMPLB is used to obtain the user's desired action (add, update, delete) and the new or changed employee record information, if required.
- A set of message definitions (EMPX21) that are used by the panels and functions to display information, including error conditions, on the panel message lines.

These dialog components are described after the "Description of Processing". (page 54)

## INVOCATION

The primary option menu, EMPL, may be invoked by:

- Including it as a selectable item on a higher-level menu.
- Specifying 'PANEL(EMPL)' on the ISPSTART command when initiating ISPF.
- Defining a command in the system command table or in an application command table with the action 'SELECT PANEL(EMPL)'. The NEWAPPL or NEWPOOL keywords may optionally be included in the SELECT action.

## DESCRIPTION OF PROCESSING

The organization of the dialog is shown in Figure 14.

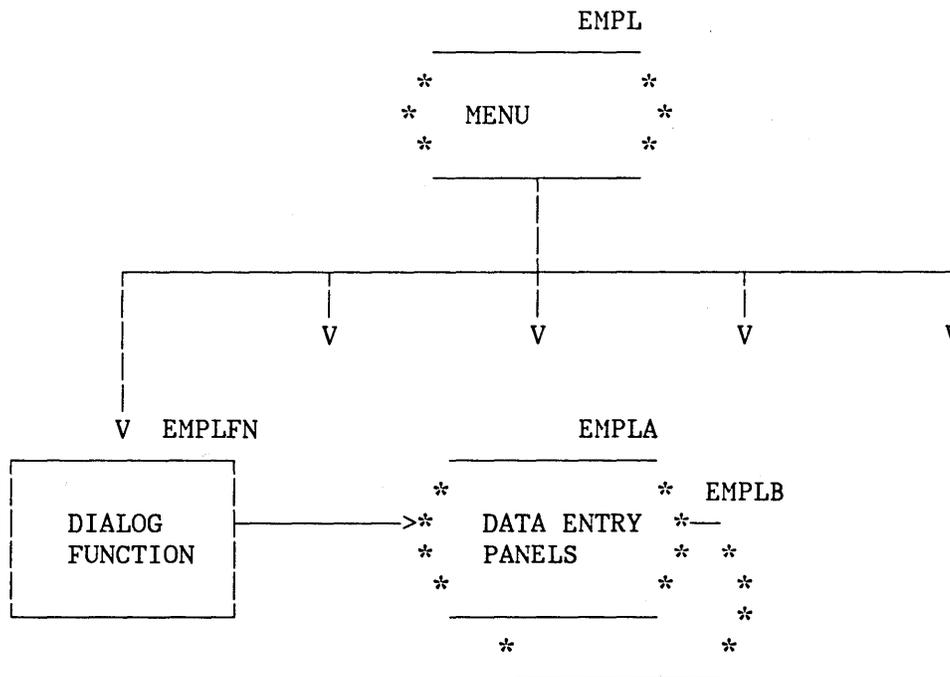


Figure 14. Organization of Employee Records Application

The dialog begins with display of the primary option menu, from which several options are selectable. Only the first option is implemented in this example. When the first option is selected, the menu invokes the dialog function named EMPLFN.

Function EMPLFN processing begins by using the TBOPEN service to open the employee table named EMPLTBL. If the table does not exist (first

execution of the function), the function invokes TBCREATE to create it. Then the function invokes the DISPLAY service to display the first data entry panel (EMPLA).

page 60 { The employee serial number, entered by the user on the first data entry panel, is checked (by VER statements in the panel )PROC section) to verify that it consists of six numeric digits. The employee serial is defined as the key variable for the employee table when it is created by the TBCREATE service, and will be used to locate information in the table.

When control is returned, the function invokes the TBGET service to attempt to find information for the current user in the employee table. Based on this attempt, variables to be displayed on the second data entry panel (EMPLB) are initialized as follows:

- If employee information is not in the table for the current user, the "type of change" field on panel EMPLB is initialized to NEW and the other input fields on the panel are initialized to blanks.
- If the employee information is found in the table for the current user, the "type of change" field on panel EMPLB is initialized to UPDATE (the user may change it to DELETE).

After the user has correctly entered requested information on the panel EMPLB and pressed the ENTER key, the function updates the employee table using TBADD (for NEW), TBPOT (for UPDATE), or TBDELETE (for DELETE). The function then redisplay the first data entry panel (EMPLA), this time with a confirmation message. It also writes this message to the log file using the LOG service.

If the user enters the END command from the second data entry panel (EMPLB), the first panel is redisplayed, no changes are made in the table, and the confirmation message is not displayed and is not written to the log file.

When the first data entry panel is redisplayed, the user may enter the same or another employee serial number and the process is repeated. When the user enters the END command from the first panel, the function closes the employee table using the TBCLOSE service and completes execution. The primary option menu from which the function was invoked is then redisplayed. At this point the user may select another option from the primary option menu.

This same example is also shown in SC34-2088-1

## Primary Option Menu Definition EMPL

The definition and the visible display portion of the primary option menu EMPL are shown below.

### Panel Definition

```
%----- EMPLOYEE RECORDS -----
+
% SELECT OPTION ==> _ZCMD
%
% 1 +MODIFY      - ADD, UPDATE, OR DELETE EMPLOYEE RECORDS
% 2 +(FUTURE)   - FUNCTION NOT YET AVAILABLE
% 3 +(FUTURE)   - FUNCTION NOT YET AVAILABLE
% 4 +(FUTURE)   - FUNCTION NOT YET AVAILABLE
% 5 +(FUTURE)   - FUNCTION NOT YET AVAILABLE
+
+ENTER%END COMMAND+TO TERMINATE.
)PROC
  &ZSEL = TRANS( TRUNC (&ZCMD, '.')
                1, 'PGM(EMPLFN)' )
)END
```

### Panel Display

```
----- EMPLOYEE RECORDS -----

SELECT OPTION ==> 1_

1 MODIFY      - ADD, UPDATE, OR DELETE EMPLOYEE RECORDS
2 (FUTURE)   - FUNCTION NOT YET AVAILABLE
3 (FUTURE)   - FUNCTION NOT YET AVAILABLE
4 (FUTURE)   - FUNCTION NOT YET AVAILABLE
5 (FUTURE)   - FUNCTION NOT YET AVAILABLE

ENTER END COMMAND TO TERMINATE.
```

### Function EMPLFN

```
EMPLFN: PROC OPTIONS(MAIN);                               /*EMPLOYEE UPDATE FUNCTION */
%INCLUDE EMPLDCL;                                         /*DCL & DEFINE VARIABLES*/
MSG = ' ';                                                /*INITIALIZE MESSAGE */
CALL ISPLINK('TBOPEN', 'EMPLTBL ');                       /*OPEN TABLE */
IF PLIRETV() = 0 THEN                                     /*IF TABLE DOESN'T EXIST*/
  CALL ISPLINK('TBCREATE', 'EMPLTBL ', '(EMP SER)', /*CREATE IT */
              '(LNAME FNAME I ADDR1 ADDR2 ADDR3 ADDR4 PHA PHNUM)');
DO WHILE (STATE = '4');                                   /*LOOP UNTIL TERM SET */
```

```

CALL ISPLINK('DISPLAY', 'EMPLA', MSG);           /*SELECT EMPLOYEE */
IF PLIRETV() = 8 THEN                             /*IF END KEY PRESSED */
  STATE = '4';                                   /* TERMINATE */
ELSE DO;                                          /*ENTER KEY PRESSED */
  MSG = ' ' ;                                   /*RESET MESSAGE */
  STATE = '2';                                   /*PROCESS EMPLOYEE PANEL*/
  CALL ISPLINK('TBGET', 'EMPLTBL ');           /*OBTAIN EMPLOYEE DATA */
  IF PLIRETV() = 0 THEN                         /*IF RECORD EXISTS */
    TYPECHG = 'U';                             /* SET UPDATE FLAG */
  ELSE DO;                                       /*RECORD DOES NOT EXIST */
    TYPECHG = 'N' ;                             /* SET TYPE = NEW */
    LNAME = ' ' ;                               /* */
    FNAME = ' ' ;                               /* INITIALIZE */
    I = ' ' ;                                   /* PANEL */
    ADDR1 = ' ' ;                               /* VARIABLES */
    ADDR2 = ' ' ;                               /* TO NULLS */
    ADDR3 = ' ' ;                               /* */
    ADDR4 = ' ' ;                               /* */
    PHA = ' ' ;                                 /* */
    PHNUM = ' ' ;                              /* */
  END;                                           /* */
  CHKTYPE = TYPECHG;                            /*SAVE TYPE OF CHANGE */
  CALL ISPLINK('DISPLAY', 'EMPLB');           /*DISPLAY EMPLOYEE DATA */
  IF PLIRETV() = 8 THEN DO;                     /*END KEY NOT PRESSED */
    IF TYPECHG = 'N' THEN DO;                  /*IF NEW EMPLOYEE */
      CALL ISPLINK('TBADD', 'EMPLTBL ');      /* ADD TO TABLE */
      MSG = 'EMPX217 ' ;                       /*EMPLOYEE ADDED MESSAGE*/
      END;                                     /* */
    ELSE DO;                                    /* */
      IF TYPECHG = 'U' THEN DO;               /*IF UPDATE REQUESTED */
        CALL ISPLINK('TBPUT', 'EMPLTBL ');   /* UPDATE TABLE */
        MSG = 'EMPX218 ' ;                   /* UPDATE MESSAGE */
        END;                                  /* */
      ELSE DO;                                  /*ELSE ASSUME DELETE */
        CALL ISPLINK('TBDELETE', 'EMPLTBL ');
        MSG = 'EMPX219 ' ;                   /*EMPLOYEE DELETED MSG */
        END;                                  /* */
      END;                                     /*END TABLE MODS */
    END;                                       /*END 2ND PANEL PROCESS */
  END;                                         /*END 1ST PANEL PROCESS */
  IF MSG = ' ' THEN CALL ISPLINK('LOG', MSG); /*LOG MSG */
END;                                           /*END DO LOOP */
CALL ISPLINK('TBCLOSE', 'EMPLTBL ');         /*CLOSE TABLE */
%INCLUDE EMPLDEL;                             /*DELETE DEFINED VARS */
RETURN(0);
END EMPLFN;

```

Included Segment EMPLDCL

```

/* */
/* DECLARE STATEMENTS AND VARIABLE DEFINITIONS FOR "EMPLFN" */
/* */
DCL ISPLINK EXTERNAL ENTRY OPTIONS(ASM INTER RETCODE) ;

```

```

/*****
* IN VSE, USE THE FOLLOWING DECLARE STATEMENT FOR ISPLINK:      *
*                                                                *
*   DCL ISPLINK EXTERNAL ENTRY OPTIONS(ASM INTER);            *
*****/

DCL PLIRETV BUILTIN      ;
DCL LENGTH  BUILTIN      ;
DCL RC FIXED BIN(31,0) INIT(0);
DCL EMPSER CHAR(6) INIT((6)' ');
DCL FNAME CHAR(16) INIT((16)' ');
DCL LNAME CHAR(16) INIT((16)' ');
DCL I      CHAR(1)  INIT(' ');
DCL ADDR1 CHAR(40) INIT((40)' ');
DCL ADDR2 CHAR(40) INIT((40)' ');
DCL ADDR3 CHAR(40) INIT((40)' ');
DCL ADDR4 CHAR(40) INIT((40)' ');
DCL PHA   CHAR(3)  INIT((3)' ');
DCL PHNUM CHAR(8)  INIT((8)' ');
DCL MSG   CHAR(8)  INIT((8)' ');
DCL TYPECHG CHAR(1) INIT(' ');
DCL CHKTYPE CHAR(1) INIT(' ');
DCL STATE CHAR(1) INIT('1');                               /*INITIAL ENTRY STATE */
/*                                                                */
/* LENGTH PARAMETER IN 'CALL ISPLINK VDEFINE' MUST BE FULL WORD. */
/*                                                                */
DCL LEMPSEK FIXED BIN(31,0) ;
DCL LFNAME  FIXED BIN(31,0) ;
DCL LLNAME  FIXED BIN(31,0) ;
DCL LI      FIXED BIN(31,0) ;
DCL LADDR1  FIXED BIN(31,0) ;
DCL LADDR2  FIXED BIN(31,0) ;
DCL LADDR3  FIXED BIN(31,0) ;
DCL LADDR4  FIXED BIN(31,0) ;
DCL LPHA    FIXED BIN(31,0) ;
DCL LPHNUM  FIXED BIN(31,0) ;
DCL LTYPECH FIXED BIN(31,0) ;
DCL LCHKTYP FIXED BIN(31,0) ;
LEMPSEK = LENGTH(EMPSEK) ;
LFNAME  = LENGTH(FNAME) ;
LLNAME  = LENGTH(LNAME) ;
LI      = LENGTH(I) ;
LADDR1  = LENGTH(ADDR1) ;
LADDR2  = LENGTH(ADDR2) ;
LADDR3  = LENGTH(ADDR3) ;
LADDR4  = LENGTH(ADDR4) ;
LPHA    = LENGTH(PHA) ;
LPHNUM  = LENGTH(PHNUM) ;
LTYPECH = LENGTH(TYPECHG) ;
LCHKTYP = LENGTH(CHKTYPE) ;

/*DEFINE VARIABLES FOR DIALOG SERVICE USE */
CALL ISPLINK('VDEFINE', '(EMPSEK)', EMPSEK, 'CHAR', LEMPSEK) ;
CALL ISPLINK('VDEFINE', '(FNAME)', FNAME, 'CHAR', LFNAME) ;
CALL ISPLINK('VDEFINE', '(LNAME)', LNAME, 'CHAR', LLNAME) ;

```

```
CALL ISPLINK('VDEFINE', '(I)', I, 'CHAR', LI) ;
CALL ISPLINK('VDEFINE', '(ADDR1)', ADDR1, 'CHAR', LADDR1) ;
CALL ISPLINK('VDEFINE', '(ADDR2)', ADDR2, 'CHAR', LADDR2) ;
CALL ISPLINK('VDEFINE', '(ADDR3)', ADDR3, 'CHAR', LADDR3) ;
CALL ISPLINK('VDEFINE', '(ADDR4)', ADDR4, 'CHAR', LADDR4) ;
CALL ISPLINK('VDEFINE', '(PHA)', PHA, 'CHAR', LPHA) ;
CALL ISPLINK('VDEFINE', '(PHNUM)', PHNUM, 'CHAR', LPHNUM) ;
CALL ISPLINK('VDEFINE', '(TYPECHG)', TYPECHG, 'CHAR', LTYPECH) ;
CALL ISPLINK('VDEFINE', '(CHKTYPE)', CHKTYPE, 'CHAR', LCHKTYP) ;
/*
```

```
*/
```

Included Segment EMPLDEL

```
/*
/*      DELETE VARIABLE DEFINITIONS FOR "EMPLFN"
CALL ISPLINK('VDELETE', '(EMPSER)') ;
CALL ISPLINK('VDELETE', '(FNAME)') ;
CALL ISPLINK('VDELETE', '(LNAME)') ;
CALL ISPLINK('VDELETE', '(I)') ;
CALL ISPLINK('VDELETE', '(ADDR1)') ;
CALL ISPLINK('VDELETE', '(ADDR2)') ;
CALL ISPLINK('VDELETE', '(ADDR3)') ;
CALL ISPLINK('VDELETE', '(ADDR4)') ;
CALL ISPLINK('VDELETE', '(PHA)') ;
CALL ISPLINK('VDELETE', '(PHNUM)') ;
CALL ISPLINK('VDELETE', '(TYPECHG)') ;
CALL ISPLINK('VDELETE', '(CHKTYPE)') ;
```

```
*/
*/
```

referenced  
on page 57

## First Data Entry Panel EMPLA

The definition and the visible display portion of the data entry panel EMPLA are shown below.

### Panel Definition

```
%----- EMPLOYEE SERIAL -----  
%COMMAND ==> _ZCMD  
+  
%ENTER EMPLOYEE SERIAL BELOW:  
+  
+  
+   EMPLOYEE SERIAL%==> _EMPSER+   (MUST BE 6 NUMERIC DIGITS)  
+  
+  
+  
+PRESS%ENTER+TO DISPLAY EMPLOYEE RECORD.  
+ENTER%END COMMAND+TO RETURN TO PREVIOUS MENU.  
)PROC  
  VER (&EMPSER, NONBLANK)  
  VER (&EMPSER, PICT, NNNNNN)  
)END
```

### Panel Display

```
----- EMPLOYEE SERIAL -----  
COMMAND ==>  
  
ENTER EMPLOYEE SERIAL BELOW:  
  
   EMPLOYEE SERIAL ==> 121212_   (MUST BE 6 NUMERIC DIGITS)  
  
PRESS ENTER TO DISPLAY EMPLOYEE RECORD.  
ENTER END COMMAND TO RETURN TO PREVIOUS MENU.
```

Second Data Entry Panel EMPLB

*also duplicated on page 12)*

The definition and the visible display portion of the data entry panel EMPLB are shown below.

Panel Definition

```
%----- EMPLOYEE RECORDS -----
%COMMAND ==> _ZCMD
+
%   EMPLOYEE SERIAL: &EMP SER
+
%   TYPE OF CHANGE%==> _TYPECHG + (NEW, UPDATE, OR DELETE)
+
%   EMPLOYEE NAME:
+   LAST   %==> _LNAME           +
+   FIRST  %==> _FNAME           +
+   INITIAL%==> _I+
+
%   HOME ADDRESS:
+   LINE 1 %==> _ADDR1           +
+   LINE 2 %==> _ADDR2           +
+   LINE 3 %==> _ADDR3           +
+   LINE 4 %==> _ADDR4           +
+
%   HOME PHONE:
+   AREA CODE %==> _PHA+
+   LOCAL NUMBER%==> _PHNUM  +
+
)INIT
  .CURSOR = TYPECHG
  IF (&PHA = ' ')
    &PHA = 914
  &TYPECHG = TRANS(&TYPECHG N,NEW U,UPDATE D,DELETE)
```

```

)PROC
&TYPECHG = TRUNC (&TYPECHG,1)
VER (&TYPECHG,LIST,N,U,D,MSG=EMPX210)
IF (&TYPECHG = N)
  IF (&CHKTYPE = N)
    .MSG = EMPX211
  IF (&TYPECHG = N)
    IF (&CHKTYPE = N)
      .MSG = EMPX212
VER (&LNAME,ALPHA)
VER (&FNAME,ALPHA)
VER (&I,ALPHA)
VER (&PHA,NUM)
VER (&PHNUM,PICT,'NNN-NNNN')
IF (&TYPECHG = N,U)
  VER (&LNAME,NONBLANK,MSG=EMPX214)
  VER (&FNAME,NONBLANK,MSG=EMPX213)
  VER (&ADDR1,NONBLANK,MSG=EMPX215)
  VER (&ADDR2,NONBLANK,MSG=EMPX215)
  VER (&ADDR3,NONBLANK,MSG=EMPX215)
)END

```

Panel Display

```

----- EMPLOYEE RECORDS -----
COMMAND ==>

EMPLOYEE SERIAL: 121212

TYPE OF CHANGE ==> NEW          (NEW, UPDATE, OR DELETE)

EMPLOYEE NAME:
  LAST    ==> JONES
  FIRST   ==> FRANKLIN
  INITIAL ==> A

HOME ADDRESS:
  LINE 1  ==> 123 MAIN STREET
  LINE 2  ==> ANYTOWN
  LINE 3  ==> NEW YORK
  LINE 4  ==> 12601

HOME PHONE:
  AREA CODE   ==> 914
  LOCAL NUMBER ==> 892-1234_

```

## Message Definitions EMPX21

EMPX210 'INVALID TYPE OF CHANGE' .ALARM=YES  
'TYPE OF CHANGE MUST BE NEW, UPDATE, OR DELETE.'

EMPX211 'TYPE 'NEW' INVALID' .ALARM=YES  
'EMPLOYEE SERIAL &EMP SER ALREADY EXISTS. CANNOT BE SPECIFIED AS NEW.'

EMPX212 'UPDATE OR DELETE INVALID' .ALARM=YES  
'EMPLOYEE SERIAL &EMP SER IS NEW. CANNOT SPECIFY UPDATE OR DELETE.'

EMPX213 'ENTER FIRST NAME' .ALARM=YES  
'EMPLOYEE NAME MUST BE ENTERED FOR TYPE OF CHANGE = NEW OR UPDATE.'

EMPX214 'ENTER LAST NAME' .ALARM=YES  
'EMPLOYEE NAME MUST BE ENTERED FOR TYPE OF CHANGE = NEW OR UPDATE.'

EMPX215 'ENTER HOME ADDRESS' .ALARM=YES  
'HOME ADDRESS MUST BE ENTERED FOR TYPE OF CHANGE = NEW OR UPDATE.'

EMPX217 '&EMP SER ADDED'  
'EMPLOYEE &LNAME, &FNAME &I ADDED TO FILE.'

EMPX218 '&EMP SER UPDATED'  
'RECORDS FOR &LNAME, &FNAME &I UPDATED.'

EMPX219 '&EMP SER DELETED'  
'RECORDS FOR &LNAME, &FNAME &I DELETED.'



## APPENDIX C. EXAMPLE OF A PDF SESSION

The following scenario is a brief example of how a program could be developed using PDF. It is intended to show how some typical program development tasks would be performed. The example is for the MVS environment. The panels shown will differ slightly in other host environments.

In the example, the programmer, at the terminal, copies a sample data set from the distributed libraries, assembles it, and views the results of the assembly.

In the example, fields filled in by the program are shown in capital letters, while fields entered by the user are shown in small letters.

**Note:** This example is not intended to show a complete scenario from beginning to end. It shows some of the types of operations that may be performed, and the types of panels that might be displayed as a result of those operations.

After logging on the host system, the user invokes PDF by entering the command PDF (or an available alias). The panel that is first displayed is the ISPF/PDF primary option menu.

At this point the user needs to allocate data sets to receive the assembler file to be copied, and to receive the output of the assembly. Selecting 3 causes the utility selection menu to be displayed.

----- ISPF/PDF PRIMARY OPTION MENU -----

OPTION ==> 3

- |   |             |   |          |          |
|---|-------------|---|----------|----------|
| 0 | ISPF PARMS  | - Specify terminal and user parameters        | USERID   | - Z59JEH |
| 1 | BROWSE      | - Display source data or output listings      | TIME     | - 12:27  |
| 2 | EDIT        | - Create or change source data                | TERMINAL | - 3277   |
| 3 | UTILITIES   | - Perform utility functions                   | PF KEYS  | - 12     |
| 4 | BACKGROUND  | - Invoke language processors in foreground    |          |          |
| 5 | BATCH       | - Submit job for language processing          |          |          |
| 6 | COMMAND     | - Enter TSO command or CLIST                  |          |          |
| 7 | DIALOG TEST | - Perform dialog testing                      |          |          |
| C | CHANGES     | - Display summary of changes for this release |          |          |
| T | TUTORIAL    | - Display information about ISPF              |          |          |
| X | EXIT        | - Terminate ISPF using log and list defaults  |          |          |

Enter END command to terminate ISPF.

To select the data set utility, the user selects 2.

----- UTILITY SELECTION MENU -----

OPTION ==> 2

- |    |           |   |   |  |
|----|-----------|---|---|--|
| 1  | LIBRARY   | - Library utility:                                |   |  |
|    |           |   | Print index listing or entire data set      |  |
|    |           |   | Print, rename, delete, or browse members    |  |
|    |           |   | Compress data set                           |  |
| 2  | DATA SET  | - DATA SET UTILITY:                               |   |  |
|    |           |   | Display data set information                |  |
|    |           |   | Allocate, rename, or delete entire data set |  |
|    |           |   | Catalog or uncatalog data set               |  |
| 3  | MOVE/COPY | - Move or copy members or datasets                |   |  |
| 4  | CATALOG   | - Catalog management:                             |   |  |
|    |           |   | Display or print catalog entries            |  |
|    |           |   | Initialize or delete user catalog alias     |  |
| 5  | RESET     | - Reset statistics for members of ISPF library    |   |  |
| 6  | HARDCOPY  | - Initiate hardcopy output                        |   |  |
| 7  | VTOC      | - Display or print VTOC entries for a DASD volume |   |  |
| 8  | OUTLIST   | - Display, delete, or print held job output       |   |  |
| 9  | COMMANDS  | - Create/change an application command table      |   |  |
| 10 | CONVERT   | - Convert old format menus/messages to new format |   |  |

When this panel is first displayed, the user leaves the option blank and enters the name of the data set containing the member to be copied, SYS1.SAMPLIB in order to display the information about that data set. Subsequently, pressing ENTER causes this panel to be displayed again. Now the user enters option A and the name of the new data set to be allocated. Pressing ENTER displays the next panel, the one to allocate a new data set.

```

----- DATA SET UTILITY -----
OPTION   ===> a

A - Allocate new data set           C - Catalog data set
R - Rename entire data set         U - Uncatalog data set
D - Delete entire data set
blank - Display data set information

ISPF LIBRARY:
PROJECT  ===> z59jeh
LIBRARY  ===> sample
TYPE     ===> asm

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME  ===>
VOLUME SERIAL  ===>           (If not cataloged, required for option "C")

DATA SET PASSWORD ===>           (If password protected)

```

Because the last data set information viewed by the user was for the data set to be copied, its characteristics are again displayed on this panel. The user blanks out the VOLUME SERIAL field to cause allocation to the authorized default volume and presses ENTER.

The user then repeats the above sequence for:

```

PROJECT  ===> z59jeh
LIBRARY  ===> sample
TYPE     ===> obj

```

```

----- ALLOCATE NEW DATA SET -----
COMMAND  ===>

DATA SET NAME: Z59JEH.SAMPLE.ASM

VOLUME SERIAL  ===>           (Blank for authorized default volume)
SPACE UNITS    ===> TRKS      (BLKS, TRKS, or CYLS)
PRIMARY QUAN   ===> 2        (In above units)
SECONDARY QUAN ===> 1        (In above units)
DIRECTORY BLOCKS ===> 1      (Zero for sequential data set)
RECORD FORMAT  ===> FB
RECORD LENGTH  ===> 80
BLOCK SIZE     ===> 3120

```

The user then enters END or presses the END PF key until the utility selection panel is redisplayed. The next operation to be performed is to copy the sample data set to the user's newly allocated data set. Selecting 3 causes the move/copy utility panel to be displayed.

On this panel the user enters C to select the "Copy without print" option.

Now the user enters the name of the data set to be copied, as shown.

Pressing ENTER causes the second panel (COPY) of the move/copy utility to be displayed. On this panel, the user enters the name of the data set to receive the copy, and presses ENTER.

----- MOVE/COPY UTILITY -----

OPTION ==> C

CP - Copy data set or member(s) and print  
MP - Move data set or member(s) and print

C - Copy without print  
M - Move without print

SPECIFY "FROM" DATA SET BELOW, THEN PRESS ENTER KEY TO SPECIFY "TO" DATA SET

FROM ISPF LIBRARY:

PROJECT ==>  
LIBRARY ==>  
TYPE ==>  
MEMBER ==>

(Blank for member list, \* for all members)

FROM OTHER PARTITIONED OR SEQUENTIAL DATA SET:

DATA SET NAME ==> 'sys1.samplib(israsm)'  
VOLUME SERIAL ==> (if not cataloged)

DATA SET PASSWORD ==> (if password protected)

COPY --- FROM SYS1.SAMPLIB(ISPASM) -----

COMMAND ==>

SPECIFY "TO" DATA SET BELOW.

TO ISPF LIBRARY:

PROJECT ==> Z59JEH  
LIBRARY ==> SAMPLE  
TYPE ==> asm  
MEMBER ==>

(BLANK UNLESS MEMBER IS TO BE RENAMED)

TO OTHER PARTITIONED OR SEQUENTIAL DATA SET:

DATA SET NAME ==>  
VOLUME SERIAL ==> (if not cataloged)

DATA SET PASSWORD ==> (if password protected)

"TO" DATA SET OPTIONS:

IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS ==> YES (YES or NO)  
IF SEQUENTIAL, "TO" DATA SET DISPOSITION ==> OLD (OLD or MOD)

Entering RETURN returns the user to the primary option menu, where he may wish to select 1 to browse through the data set, or 2 to edit the data set.

For this example, the user selects 2 to edit the data set. This results in a display of the edit entry panel.

The user enters the name of the data set he has just copied, and presses ENTER. He is then presented with an edit panel displaying the start of the data set.

```
----- EDIT - ENTRY PANEL -----
COMMAND ==> _
```

```
ISPF LIBRARY:
PROJECT ==> z59jeh
LIBRARY ==> sample      ==>          ==>          ==>
TYPE    ==> asm
MEMBER  ==> israsm      (BLANK FOR MEMBER SELECTION LIST)
```

```
OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>      (If not cataloged)
```

```
DATA SET PASSWORD ==>      (If password protected)
```

```
PROFILE NAME      ==>      (Blank defaults to data set type)
```

Here the user may make changes to the data set using a variety of methods. With the full-screen editor, simple changes may be made by merely overtyping old information with the new information.

There are also various commands that assist in locating data to be changed, and in making more complex changes.

When all changes are complete, the user enters an END command to save the data set with its changes. When the edit selection panel is redisplayed, another END command returns him to the primary option menu.

```
EDIT --- Z59JEH.SAMPLE.ASM(ISRASM) ----- COLUMNS 001 072
COMMAND ==> _                               SCROLL ==> HALF
***** ***** TOP OF DATA *****
==MSG> -CAUTION- PROFILE IS SET TO "STATS ON". STATISTICS DID NOT EXIST FOR
==MSG> THIS MEMBER, BUT WILL BE GENERATED IF DATA IS SAVED.
000500 ASMTST CSECT
001000          DC      F'01'
001500          DC      F'02'
002000          DC      F'03'
002500          DC      F'04'
003000          DC      F'05'
003500          DC      F'06'
004000          DC      F'07'
004500          DC      F'08'          COMMENT
005000          DC      F'09'
005500          DC      F'10'
006000          DC      F'11'
006500          DC      F'12'
007000          DC      F'13'
007500          DC      F'14'
008000          DC      F'15'
008500          DC      F'16'
009000          DC      F'17'          COMMENT
009500          DC      F'18'
```

For this example, the user next selects 4 to initiate a foreground assembly of the data set. This results in a display of the foreground selection menu.

Here the user selects 1 to select the system assembler and the foreground assembly panel is displayed.

```

----- FOREGROUND SELECTION MENU -----
OPTION ===> 1

1 - System assembler          9 - SCRIPT/VS
2 - OS/VS COBOL compiler     10 - COBOL interactive debug
3 - FORTRAN IV (G1) compiler 11 - FORTRAN interactive debug
4 - PL/I checkout compiler
5 - PL/I optimizing compiler
6 - PASCAL/VS compiler
7 - Linkage editor

```

The user enters the information shown and presses ENTER and the assembly is initiated.

```

----- FOREGROUND ASSEMBLY -----
COMMAND ===>

ISPF LIBRARY:
PROJECT ===> Z59JEH
LIBRARY ===> SAMPLE    ===>          ===>          ===>
TYPE    ===> ASM
MEMBER  ===> israsm    (Blank for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ===>

LIST ID ===> asmtst1          PASSWORD ===>

ASSEMBLER OPTIONS: (Options LIB, LOAD, and PRINT generated automatically)
===> LIST,TEST,RENT

ADDITIONAL INPUT LIBRARIES:
===>
===>

```

When the assembly has been completed, PDF automatically calls the BROWSE function for the assembly listing.

When the user has finished browsing the assembler output, he enters an END command and is then presented with the foreground print options panel.

```

BROWSE - Z59JEH.ASMTEST1.LIST ----- LINE 000000 COL 001 080
COMMAND ==> _____ SCROLL ==> PAGE
***** TOP OF DATA *****-CAPS ON-**

  LOC  OBJECT CODE  ADDR1 ADDR2  STMT  SOURCE STATEMENT
00000  00000001      1  ASMTEST CSECT
00000  00000001      2      DC      F'01'
00004  00000002      3      DC      F'02'
00008  00000003      4      DC      F'03'
0000C  00000004      5      DC      F'04'
00010  00000005      6      DC      F'05'
00014  00000006      7      DC      F'06'
00018  00000007      8      DC      F'07'
0001C  00000008      9      DC      F'08'
00020  00000009     10      DC      F'09'
00024  0000000A     11      DC      F'10'
00028  0000000B     12      DC      F'11'
0002C  0000000C     13      DC      F'12'
00030  0000000D     14      DC      F'13'
00034  0000000E     15      DC      F'14'
00038  0000000F     16      DC      F'15'
0003C  00000010     17      DC      F'16'
00040  00000011     18      DC      F'17'
00044  00000012     19      DC      F'18'

```

Here the user may indicate what he wants to do with the assembler listing. After entering an option (the default is K if no option is entered) the user presses ENTER to exercise that option and return to the foreground assembly panel.

```

----- FOREGROUND PRINT OPTIONS -----
OPTION ==>

  PK - Print data set and keep           K - Keep data set (without printing)
  PD - PRINT DATA SET AND DELETE       D - DELETE DATA SET (WITHOUT PRINTING)

  If END command is entered, data set is kept without printing

DATA SET NAME: Z59JEH.ISPASM.LIST

SYSOUT CLASS ==> (For system printer)
PRINTER ID   ==> (For 328x printer)

JOB STATEMENT INFORMATION: (Required for system printer)
==> //Z59JEHC JOB (ACCOUNT),'NAME'
==> //*
==> //*
==> //*

```

Entering END from the foreground assembly panel returns the user to the foreground selection panel. Entering END again redisplay the primary option menu.

There are two ways to exit from the primary option menu:

1. Entering X exits PDF using the default disposition for the log and list data sets.
2. Entering END exits PDF but first provides you with a panel for specifying the disposition of the log and list data sets.

----- ISPF/PDF PRIMARY OPTION MENU -----

OPTION ==> X

0	ISPF PARMS	- Specify terminal and user parameters	USERID	- Z59JEH
1	BROWSE	- Display source data or output listings	TIME	- 12:27
2	EDIT	- Create or change source data	TERMINAL	- 3277
3	UTILITIES	- Perform utility functions	PF KEYS	- 12
4	FOREGROUND	- Invoke language processors in foreground		
5	BATCH	- Submit job for language processing		
6	COMMAND	- Enter TSO command or CLIST		
7	DIALOG TEST	- Perform dialog testing		
C	CHANGES	- Display summary of changes for this release		
T	TUTORIAL	- Display information about ISPF		
X	EXIT	- Terminate ISPF using log and list defaults		

Enter END command to terminate ISPF.

## GLOSSARY

This glossary defines the terms and abbreviations used in this manual as they specifically relate to the ISPF and ISPF/PDF Program Products. For general definitions and terms, refer to the IBM Vocabulary for Data Processing, Telecommunications, and Office Systems GC20-1699.

**breakpoint.** A suspension of the execution of a user dialog so that the dialog test facilities may be used. Control is given to the user, and user dialog data such as tables, variables, etc. may be examined and manipulated. In addition, new test conditions, such as traces and additional breakpoints may be set. (see also trace)

**CLIST.** A command procedure for TSO on MVS.

**command procedure.** A previously defined set of commands to be invoked on a particular system.

In the TSO environment, this is a data set or a member of a partitioned data set containing TSO commands to be performed sequentially by the EXEC command.

In the CMS environment, this is a file (usually with a filetype of EXEC), which contains a series of CP or CMS commands (or both) together with conditional branching facilities to eliminate the repetitious re-keying of those command sequences.

**data entry panel.** A type of panel that prompts the user to supply specific information.

**dialog.** Any application designed to be run under the ISPF dialog manager. A dialog is composed of program and data elements, which together provide an orderly interaction between the computer

and the user of that application. This interaction consists of a series of interrelated inquiries and responses analogous to a conversation between two people.

**dialog variable.** A variable that is used to communicate data between the dialog service and functions. (see also function variable, profile variable, shared variable and variable)

**EXEC2.** A command procedure for VM/SP.

**file.** A sequential data set in the MVS environment, or a sequential CMS file in the VM/SP environment.

**file skeleton.** A generalized representation of sequential data that may be customized during dialog execution to produce an output file. The customization consists of the selection of lines to be included in the output, and the substitution of variables as obtained from functions, panels, or tables.

**function.** A program that may (1) control the sequence in which panels and messages are displayed, (2) oversee the handling of data and files, and (3) perform whatever processing is required for the application. A function may be written in an assembler language, a high-level language such as COBOL, PL/I, or FORTRAN, or the command procedures supported by the host operating system.

**function variable.** A variable that is known only within the currently active function. (see also dialog variable, profile variable, shared variable and variable)

**help/tutorial panel.** A type of panel used for reference or instructional purposes. Generally, however, these panels are presented to a user at

specific points in a dialog upon request for more information about what to do at that step. This term also refers to a complete set of panels (a tutorial) specifically used for instructional purposes.

**information display panel.** (see help/tutorial panel)

**ISPF.** The Interactive System Productivity Facility. The "dialog manager," which is the base program product for the ISPF/PDF Program Product as well as other interactive applications. It provides the interface between a display terminal and applications such as the PDF.

**ISPF/PDF.** The Interactive System Productivity Facility/Program Development Facility. An application or dialog program product running under ISPF and designed to increase programmer productivity.

**library.** A collection of related files. For example, one line of an invoice may form an item, a complete invoice may form a file, the collection of inventory control files may form a library, and the libraries used by an organization are known as its data bank.

This may be a partitioned data set in the MVS environment, or a MACLIB in the VM/SP environment.

**menu.** A list of options (see also selection panel).

**messages.** A comment that provides special information to the user. It may alert the user about what to expect next, confirm that a user-requested action is in progress, or provide a report of some error condition.

**model.** A predefined set of statements for a dialog element (for example, an ISPF service) that can be copied into the program being edited, and then modified for a specific application.

**model class.** A grouping of models according to the type of dialog element in which they are used: CLIST, EXEC, MSGS, PANELS, PROGRAM (COBOL, PLI, or FORTRAN), or SKELS.

**panel.** A predefined display image. Panel types are selection, data entry, data display, or help/tutorial.

**PDF.** (see ISPF/PDF)

**profile variable.** A variable that is contained in the user profile and automatically retained from one session to another for the user. (see also dialog variable, function variable, shared variable and variable)

**selection panel.** A type of panel that presents a list of options (a menu) to the user, who must then make a selection to proceed in the dialog.

**shared variable.** A variable that is contained in the shared variable pool, and accessible to different functions belonging to the same application. A variable is placed in the shared variable pool with the VPUT service and retrieved with the VGET service. (see also dialog variable, function variable, profile variable and variable)

**skeleton.** (see file skeleton)

**tables.** A two-dimensional array used to contain data. A table may be created as a temporary repository, or it may be retained from one session to another. A retained table may also be shared between different applications. The type and amount of data stored in a table depend upon the nature of the application.

**trace.** A recording of descriptive data into the ISPF log when certain events occur in the user dialog being tested. These events may include the use of dialog services and/or the use of dialog variables. (see also breakpoint)

**tutorial panel.** (see help/tutorial panel)

**variable.** A character string that may vary in length from zero to 32,767 bytes, referred to by a symbolic name. (see also dialog variable)



## INDEX

### Special Characters

& (ampersand), substitutable variable 12  
+ (plus sign), attribute character 11, 12  
&ZIND variable 43  
&ZPRIM variable 43  
)INIT, header statement 12  
)PROC, header statement 13  
% (percent sign), attribute character 11  
\_ (underscore), attribute character 11, 12, 20

### A

access  
  data set or file 25  
  ICCF libraries 28  
  VSE system libraries 27  
access methods, BPAM, BSAM, TCAM, VTAM 37  
acknowledgement, message, dialog element 6  
additions, multiple 4  
Advanced Function 39  
allocating, files (MVS), under utilities 34  
ampersand, substitutable variable 12  
APL  
  display character set 41, 42  
  keyboard 50  
array, data, See table  
ASCII, display character set 41, 42  
assembler 38, 39  
  foreground processor 34  
  language 6  
  linkage to 4  
assembler prompter 38  
ATTN attribute keyword, changes 49  
attribute character, default 11  
attribute section, panel definition 11  
audible alarm 42

### B

background, See batch  
BATCH, PDF option 35  
  changes 52  
batch execution of non-interactive dialogs, changes 50  
batch jobs 4  
blank lines, VSE restriction 11  
body, panel definition 11  
BPAM 37  
breakpoint  
  definition 73  
  dialog test 35  
breakpoints, managing 36  
browse 4  
  PDF option 31  
browse service 4  
browsing, ISPF log 35  
BSAM 37  
building user tables 3  
bulk entry of text 33

### C

Canadian-French, display character set 42  
CANCEL, edit command 32  
CHANGE  
  edit command 32  
  PF key function 51  
change activity, history 33  
changes  
  multiple 4  
  PDF option 36  
changes from predecessor program product (SPF)  
  dialog management 47  
  program development facility 50  
checkout, PL/I, compiler 38  
CICS/VS 39  
CLIST  
  definition 73  
  language 5  
  MVS 20  
  TSO 3

- variables 9
- CMS
  - EXEC 3
    - new command 48
- COBOL 3, 38, 39
  - Compiler and Library 38
  - foreground processor 34
  - Interactive Debug 38
    - language 6
    - prompter 38
- column boundaries 32
- column identification line 32
- COMMAND, ISPSTART, employee records 54
  - PDF option 35
- command delimiter 50
- command field 50
  - alternative location, changes 49
- command option, changes 52
- command procedure
  - definition 73
  - dialog element 6
- command tables 3
  - changes 48
- communication with user 3
- compiler, linkage to 4
- conditional include of records 19
- CONT variable 43
- context editing, full-screen 4
- control elements 5
- control of user variables 3
- control services, changes 50
- COPY, edit command 32
- copying, files, under utilities 34
- copying lines, under edit 32
- CP, new command 48
- cursor select, changes 48
- customer responsibility 42
- customization, of data 6

## D

### data

- customization 6
  - location 4
  - program-generated 3
  - user-entered 3
- data array, See table
- data display panel 23
- data elements 5
- data entry 60, 61
- data entry panel 2, 23, 60
  - definition 73

- dialog element 5
- data entry panel EMPLB, employee records application 61
- data integrity 42
- data set, access 25
- debug, interactive
  - COBOL 38
  - FORTTRAN 38
- declare statements, pl1 57
- default, attribute character 11
- defaults, console, log, and list, specifying 31
- definition of user variables 3
- delete variables, pl1 59
- deleting, library members or files, under utilities 34
- deleting lines, under edit 32
- delimiter, command 50
- description of processing, employee records 54
- dialog 1
  - definition 73
  - elements 5
  - example 53
  - migration of 43
  - organization 2, 6
  - test 4
  - variable 19
- dialog components
  - data entry panels EMPLA, EMPLB 53
  - employee records 53
  - messages 53
  - primary option menu EMPL 53
- dialog management, service 14
- dialog manager 1
- DIALOG TEST, PDF option 35
  - changes 52
- dialog variable, definition 73
- disconnected virtual machine 38
- DISPLAY
  - formatting 10
  - pl1, employee records 57
  - service 7, 10, 14, 20
  - services 16
- display image, visible 5
- display services, changes 48
- DISPLAY, display service 16
- displaying
  - panels 35
  - table rows, table status, and table structure 35
  - variable values 35
- Document Composition Facility 4, 38

Document Library Facility 4  
document preparation 33  
documentation preparation 4  
DSPRINT command processor 37, 42  
dual-case character set 42  
dynamic selection panel formatting,  
changes 49

## E

EBCDIC, display character set 41,  
42  
edit 4  
    copying lines 32  
    deleting lines 32  
    inserting lines 32  
    modifying data 32  
    shifting lines 32  
edit option, changes 51  
edit recovery 33  
edit service 4  
editor, full-screen context 4  
EMPL panel 53, 54, 56  
EMPLA panel 53  
    definition, 60  
    display 60  
EMPLB panel 53  
    definition 61  
    display 62  
employee records 53  
    data entry 60, 61  
    dialog components 53  
    first data entry panel EMPLA 60  
    invocation 54  
    ISPSTART command 54  
    message definitions EMPX21 63  
    messages 63  
    organization 54  
    overview 53  
    pll  
        declare statements 57  
        delete variables 59  
        included segment 57  
        variable definitions 57  
    processing description 54  
    TBGET, pll 57  
environment, operating 1  
examples  
    dialog 7, 8  
    of a PDF session 65  
    of an ISPF dialog 53  
    sample panel

    definition 12  
    display 14  
    sharing of variables 10  
excluded lines 32  
EXEC, CMS 3  
EXEC2  
    CMS 20  
    definition 73  
    language 5  
    variables 9  
EXIT, PDF option 36  
extended verification, changes 49

## F

file  
    access 25  
    definition 73  
file skeleton  
    definition 73  
    dialog element 6  
file tailoring, service 14  
file tailoring output under dialog  
    control, changes 49  
file tailoring service 19  
FIND  
    edit command 32  
    PF key function 51  
FOREGROUND, PDF option 34  
Foreground Environment Feature 38  
foreground jobs 4  
foreground option, changes 52  
format, of panel 24  
formatting, of displays 10  
FORTRAN 38  
    foreground processor 34  
    Interactive Debug 38  
    language 6  
    Prompter 38  
FORTRAN IV 3  
FTCLOSE, file tailoring service 19  
FTERASE, file tailoring service 19  
FTINCL, file tailoring service 19  
FTOPEN, file tailoring service 19  
full-screen, editing 4  
function 2  
    definition 73  
    dialog element 5  
function variable, definition 73  
functions  
    invoking 35  
    invoking from menus 3

## G

general services 17  
generation of sequential output 3  
glossary 73

## H

hardware tabs 32  
help, online 3  
help panel  
    changes 49  
    definition 73  
hexadecimal display 33

## I

ICCF 39  
    PDF option 35  
ICCF libraries, access 28  
include records conditionally 19  
included segment, pl1 57  
information display, dialog  
    element 5  
initialization section, panel  
    definition 11, 13  
insert mask 32  
inserting lines, under edit 32  
insertion of new material 33  
interactive debug  
    COBOL 38  
    FORTRAN 38  
interface, to language  
    processors 4  
invocation  
    employee records 54  
    of ISPF 21  
    of PDF 30  
invoking  
    functions 35  
    ISPF service 35  
ISPF  
    definition 74  
    invocation 21  
    library 25  
    library structure 25  
    log, browsing 35  
    PDF, definition 74  
ISPF PARMS, PDF option 31  
ISPF service, invoking 35  
ISPPARM data set 43  
ISPPROF data set 43  
ISPSTART, command 21

ISPSTART command, employee  
    records 54  
ISPVM, See disconnected virtual  
    machine  
iteratively process records 19

## K

KANA keyword, changes 49  
Katakana, display character  
    set 41, 42  
KEYS, new command 48

## L

language processor interface 4  
library  
    definition 74  
    hierarchy 25  
    identifier 25  
    ISPF 25  
    maintenance 4  
    multilevel support 4  
    organization 6  
library identifier  
    library name 25  
    project name 25  
    type 25  
library name, library  
    identifier 25  
library structure, ISPF 25  
light pen select, changes 48  
link edit 34  
linkage editor 38, 39  
    linkage to 4  
LIST  
    selection 23  
    verification keyword 13  
loader, linkage to 4  
LOCATE, edit command 32  
location, of data 4  
LOG, pl1, employee records 57  
logical tabs 32

## M

machine requirements 40  
maintenance  
    of libraries 4  
    of user tables 3  
    panel and message  
        definitions 11

- managing, breakpoints 36
- menu 2, 23
  - definition 74
  - dialog element 5
  - display 3
- message
  - definition 74
  - dialog element 6
  - display formatting 10
- message definitions, maintenance 11
- message field, alternative location, changes 49
- message handler, TCAM 42
- message saving, SETMSG display service 16
- messages, employee records 63
- migration of dialogs 43
- MODEL
  - definition 74
  - new edit command 51
- model class, definition 74
- model facility 3, 33
- model section, panel definition 11
- models 4
- modifying, table rows 35
- modifying data, under edit 32
- MOVE, edit command 32
- moving, files, under utilities 34
- multilevel programming library 4
- multiple, additions and changes 4
- multiple screen display 3
- MVS requirements 37

## N

- name-list syntax, changes 47
- names
  - CMS file 29
  - MVS data set 29
  - reserved 21
  - VSE data set 29
- naming conventions, conventional data set and file 29
- NEWAPPL keyword 43
  - changes 47
- NOCHECK keyword, changes 47
- null characters 32

## O

- online
  - help 3
  - tutorial 3, 4
- operating environment 1
- optimizing, PL/I, compiler 38, 39
- organization
  - employee records 54
  - of a dialog 2, 6
  - of a library 6
- overview, employee records 53

## P

- pageable link pack area 40
- panel 5
  - data display 23
  - data entry 2, 23, 60
    - definition 73
  - definition 74
  - dialog element 5
  - EMPL 53, 54
    - definition 56
    - display 56
  - EMPLA 53
    - definition 60
    - display 60
  - EMPLB 53
    - definition 61
    - display 62
  - format 24
  - selection 2, 23
  - table display 23
  - types of 23
- panel definition
  - attribute section 11
  - body 11
  - initialization section 11, 13
  - model section 11
  - processing section 11, 13
- panel definitions, maintenance 11
- panel EMPL
  - definition 56
  - display 56
- PANELID, new command 48
- panels
  - displaying 35
  - testing 35
- partition, display screen 3
- partition (pseudo) size (VSE) 40
- PASCAL 38
  - foreground processor 34

PCF, See Programming Control Facility

PDF  
 definition 74  
 invocation 30  
 options 31

percent sign, attribute character 11

PF key assignments, specifying 31

PICT, verification keyword 13

picture, display format 10

PL/I  
 Checkout Compiler 38  
 foreground processor 34  
 language 6  
 Optimizing Compiler 38, 39

PL/I Optimizing Compiler 3

planning 37

PLPA, See pageable link pack area

plus sign, attribute character 11, 12

pll  
 declare statements 57  
 delete variables 59  
 DISPLAY, employee records 57  
 included segment,  
 define variable 57  
 delete variables 59  
 LOG, employee records 57  
 TBADD, employee records 57  
 TBCLOSE, employee records 57  
 TBCREATE, employee records 56  
 TBDELETE, employee records 57  
 TBGET, employee records 57  
 TBOPEN, employee records 56  
 TBPUT, employee records 57  
 variable definitions 57  
 VDEFINE, employee records 58  
 VDELETE, employee records 59

primary option menu 30, 53

primary option menu panel 56

primary option menu panel EMPL 56

PRINT ON/OFF, command renamed 51

printing, library members or files, under utilities 34

processing section, panel  
 definition 11, 13

processor, language, interface to 4

profile data 43

profile variable 10  
 changes 49  
 definition 74

program 2  
 function, dialog element 6

program development 4

program elements 5

program function keys, changes 48

program-generated data 3

Programming Control Facility 37

programming requirements 37

project name, library identifier 25

prompter  
 assembler 38  
 COBOL 38  
 FORTRAN 38

publications, supporting 44

## Q

qualifier, See library identifier

## R

RCHANGE, PF key function 51

rearrangement of text 33

region size, minimum (MVS) 40

renaming, library members or files, under utilities 34

requirements  
 machine 40  
 MVS 37  
 programming 37  
 terminal 41  
 VM/SP 38  
 VSE/ICCF 39

reserved names 21

responsibility, customer 42

restriction  
 blank lines in VSE 11  
 command procedures not supported for VSE 9

VSE  
 cannot edit ICCF members containing ICCF commands 31  
 cannot invoke ISPF as ICCF command 39  
 cannot write data to ICCF members containing ICCF commands 34  
 core image libraries 34  
 data set labels 35  
 foreground option 42  
 no command procedure supported 44  
 SCRIPT/VS 42

RFIND, PF key function 51

RPG 39  
RSCS Networking 39

## S

SAVE, edit command 32  
screen, split 5  
screen size 24, 42  
SCRIPT/VS  
    See also Document Composition  
        Facility, foreground  
        processor 34  
scrolling 4  
    4-way, browse 31  
security 42  
SELECT, service 9  
select service 14  
SELECT, service 15  
selection list 23  
selection panel 2, 23  
    definition 74  
    dialog element 5  
sequential data, See skeleton,  
file, dialog element  
sequential output generation 3  
service  
    display 16  
    file tailoring 19  
    general 17  
    variable 19  
services  
    dialog management 14  
    display 14  
    file tailoring 14  
    select 14  
    table 14, 16  
    variable 14  
session manager 37  
set next selection menu,  
changes 48  
SETMSG, display service 16  
setting, variable values 35  
shared variable 10  
    definition 74  
shifting lines, under edit 32  
SIPO 1  
size  
    DCSS (VM/SP) 40  
    mimumum region (MVS) 40  
    PLPA (MVS) 40  
    pseudo partition (VSE) 40  
    recommended 40  
    SVA (VSE) 40  
    virtual machine (VM/SP) 40

skeleton, file, dialog element 6  
    file tailoring services 19  
    imbedding other 19  
SKIP attribute keyword, changes 49  
small CP option 40  
SNA 38  
software tabs 32  
space requirements 40  
specification, of PF keys,  
changes 51  
specifying  
    console, log, and list  
    defaults 31  
    PF key assignments 31  
    terminal characteristics 31  
SPF 1  
    changes from 47  
SPLIT, command 5  
split-screen 3, 5, 50  
substitution  
    of variable data 6  
    of variables, file tailoring  
    service 19  
support option, See dialog test  
System Installation Productivity  
Option, See SIPO  
System Productivity Facility, See  
SPF

## T

table  
    definition 74  
    dialog element 6  
    permanent 17  
    rows, displaying and  
    modifying 35  
    services 16  
    status and structure,  
    displaying 35  
    temporary 17  
table display, dialog element 5  
table display panel 23  
table service output under dialog  
control, changes 49  
table services 14  
tabs  
    hardware 32  
    logical 32  
    software 32  
TBADD  
    employee records 55  
    pll, employee records 57  
table service 18

TBBOTTOM, table service 18  
 TBCLOSE, pll, employee records 57  
   table service 17  
 TBCREATE  
   employee records 55  
   pll, employee records 56  
   table service 17  
 TBDELETE  
   employee records 55  
   pll, employee records 57  
   table service 18  
 TBDISPL, display service 16  
 TBEND, table service 17  
 TBERASE, table service 18  
 TBEXIST, table service 18  
 TBGET, pll, employee records 57  
   table service 18  
 TBMOD, table service 18  
 TBOPEN, pll, employee records 56  
   table service 17  
 TBPUT  
   employee records 55  
   pll, employee records 57  
   table service 18  
 TBQUERY, table service 17  
 TBSARG, table service 18  
 TBSAVE, table service 17  
 TBSCAN, table service 18  
 TBSKIP, table service 18  
 TBTOP, table service 18  
 TBVCLEAR, table service 18  
 TCAM 37  
   message handler 42  
 terminal characteristics,  
   specifying 31  
 terminal requirements 41  
 test, dialog 4  
 test facility 3  
   See also dialog test  
 testing, panels 35  
 text  
   display character set 42  
   keyboard 50  
 text panel, dialog element 5  
 trace, definition 74  
 tracing  
   applications and function calls  
     to ISPF services 36  
   dialog test 35  
   variable usage 36  
 tracking, program segments 4  
 truncation remainder function,  
   changes 49  
 TSO  
   CLIST 3

  new command 48  
 tutorial  
   online 3, 4  
   PDF option 36  
 tutorial panel  
   definition 73  
   dialog element 5  
 two-dimensional array, table 16  
 two-screen 3, 5  
 type, library identifier 25

## U

underscore, attribute  
   character 11, 12, 20  
 UP variable 43  
 user, communication with 3  
 user tables, building and  
   maintaining 3  
 user-entered data 3  
 utilities 4  
   PDF option 34  
 utilities option, changes 51

## V

variable  
   association with functions 9  
   communication between  
     functions 9  
   definition 19, 75  
   dialog 19  
   dialog element 6  
   explicitly defined 9  
   function  
     defined 10  
     implicit 10  
   implicitly defined 9, 10  
   profile 10  
   services 9, 14, 19  
     changes 47  
   shared 9, 10  
   substitutable 6, 12  
   use of 9, 19  
   user, definition and control 3  
   user profile 9  
 variable definitions, pll 57  
 variable usage, tracing 36  
 variable values, displaying and  
   setting 35  
 VCOPY, variable service 20  
 VDEFINE, pll, employee records 58  
   service 9

- variable service 20
- VDELETE, pl1, employee records 59
  - variable service 20
- VER statement 13
- verification, of data 13
- VGET
  - service 9, 10
  - variable service 20
- VGET/VPUT 43
- virtual machine size (VM/SP) 40
- VM Communication Facility (VMCF) 38
- VM/System Product (VM/SP) 38
- VM/VTAM Communications Network Application (VM/VCNA) 38
- VPUT
  - service 9, 10
  - variable service 20
- VREPLACE, variable service 20
- VRESET, variable service 20
- VS FORTRAN 3
- VS/FORTRAN 39
- VSE restriction
  - blank lines 11
  - cannot edit ICCF members containing ICCF commands 31
  - cannot invoke ISPF as ICCF command 39
  - cannot write data to ICCF members containing ICCF commands 34

- command procedures 9
- core image libraries 34
- data set labels 35
- foreground option 42
- no command procedure supported 44
- VSE restriction/SCRIPT/VS 42
- VSE System IPO/E 39
- VSE system libraries, access 27
- VSE/ICCF 39
- VSE/ICCF requirements 39
- VSE/POWER 39
- VTAM 37

## W

- warning, message, dialog element 6

## Z

- Z, reserved names 21
- Z variables, changes 49
- ZCMD variable 43
- ZCONT variable 43
- ZSEL variable 43
- ZUP variable 43



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 Z59, Building 931  
P. O. Box 390  
Poughkeepsie, New York 12602



ISPF: General Information

Printed in U.S.A.

GC34-2078-2

Fold and tape

Please Do Not Staple

Fold and tape



Publication Number  
GC34-2078-2

File Number  
S370/4300-39

Program Number  
ISPF 5668-960  
ISPF/PDF 5665-268 MVS  
5664-172 VM  
5666-281 VSE

Printed in  
USA

