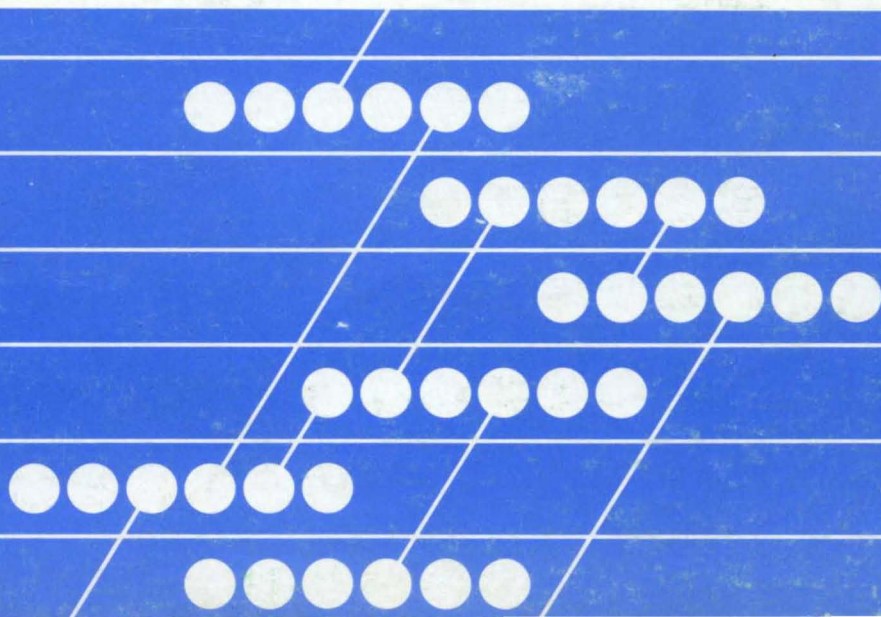


Interactive  
System Productivity Facility  
(ISPF) and ISPF/Program  
Development Facility (PDF)  
Version 2

General Information

MVS/Extended Architecture

**IBM**



---

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

---

**General Information**

---

**MVS/Extended Architecture**

---

Publication Number  
GC34-4041-0

File Number  
S370/4300-39

---

Program Numbers  
5665-317  
5665-319

#### First Edition (December 1984)

| This is a new manual and supports ISPF for MVS/XA and MVS/370.  
| You should use this manual if you have ISPF with an MVS/XA  
| environment or if you are using APL2 with ISPF in either an  
| MVS/370 or MVS/XA environment. This manual is a revision of,  
| but does not obsolete, GC34-2136-0 for ISPF/MVS. Changes or  
| additions to the text and illustrations are indicated by a  
| vertical line to the left of the change or addition.

| This edition applies to version 2, release 1, modification 2 of  
| the Interactive System Productivity Facility (ISPF) Program  
| Product, (Program Number 5665-319) and to the ISPF/Program  
| Development Facility (ISPF/PDF or PDF) Program Product (Program  
| Number 5665-317) for use with OS/VS2 MVS Release 3.8 or MVS/SP  
| Release 1.1.1 and to all subsequent releases until otherwise  
| indicated by new editions or technical newsletters.

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.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Dept. T46, P. O. Box 60000, Cary, North Carolina, 27511. 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.

## 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 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 are contained in ISPF and the program development capabilities are contained in PDF.

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 publications.
- **Summary** describes the testing period, license, program services, warranty, and the availability for ISPF and ISPF/PDF.
- **New ISPF Functions** lists the new capabilities provided in ISPF Version 2.
- **New PDF Functions** lists the new capabilities provided in ISPF/PDF Version 2.
- **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.
- **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.

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" on page 41.



# CONTENTS

<b>Chapter 1. Introduction</b> . . . . .	<b>1</b>
Operating Environment . . . . .	2
ISPF Dialog Management . . . . .	3
ISPF/PDF Program Development Functions . . . . .	5
<b>Chapter 2. Dialog Management</b> . . . . .	<b>7</b>
Concept of a Dialog . . . . .	7
Parts of a Dialog . . . . .	8
Organization of a Dialog . . . . .	9
Use of Variables . . . . .	11
Formatting of Displays . . . . .	14
Dialog Management Services . . . . .	17
Select Service . . . . .	19
Control Service . . . . .	19
Log Service . . . . .	19
Display Services . . . . .	19
Displays Containing a GDDM Generated Picture or Graph . . . . .	20
Displays Containing Dynamically Generated Areas . . . . .	21
Table Services . . . . .	21
General Services . . . . .	22
Row Services . . . . .	22
File Tailoring Services . . . . .	23
Variable Services . . . . .	24
ISPF Invocation . . . . .	26
<b>Chapter 3. Program Development</b> . . . . .	<b>29</b>
PDF Displays . . . . .	29
Types of Panels . . . . .	29
Screen Size . . . . .	30
Panel Format . . . . .	30
Library and Data Set Access . . . . .	31
Library Structure . . . . .	31
Conventional Data Set Naming Conventions . . . . .	33
PDF Invocation . . . . .	33
PDF Options . . . . .	33
Option 0 - ISPF PARMS . . . . .	33
Option 1 - BROWSE . . . . .	34
Option 2 - EDIT . . . . .	35
Option 3 - UTILITIES . . . . .	37
Option 4 - FOREGROUND . . . . .	37
Option 5 - BATCH . . . . .	38
Option 6 - COMMAND . . . . .	38
Option 7 - DIALOG TEST . . . . .	38
Option 8 - LIBRARY MANAGEMENT UTILITIES . . . . .	39
Option C - CHANGES . . . . .	39
Option T - TUTORIAL . . . . .	40
Option X - EXIT . . . . .	40

<b>Chapter 4. Planning</b>	<b>41</b>
Programming Requirements	41
Machine Requirements	43
Terminal Requirements	45
Customer Responsibilities	46
Migration	46
From SPF	46
From ISPF Version 1	47
Converting Dialogs From VM or VSE to MVS	48
Publications	48
<b>Appendix A. New ISPF Functions</b>	<b>51</b>
<b>Appendix B. New ISPF/PDF Functions</b>	<b>59</b>
<b>Appendix C. Example of an ISPF Dialog</b>	<b>63</b>
Overview	63
Invocation	64
Description of Processing	64
Dialog Components	66
Primary Option Menu Definition EMPL	66
Panel Definition	66
Panel Display	66
Function EMPLFN	67
First Data Entry Panel EMPLA	70
Panel Definition	70
Panel Display	70
Second Data Entry Panel EMPLB	71
Panel Definition	71
Panel Display	72
Message Definitions EMPX21	73
<b>Appendix D. Example of a PDF Session</b>	<b>75</b>
<b>Glossary</b>	<b>83</b>
<b>Index</b>	<b>87</b>

## FIGURES

1.	ISPF Operating Environment . . . . .	2
2.	ISPF Application Development Environment . . . . .	3
3.	Dialog Starting with a Selection Panel . . . . .	10
4.	Dialog Starting with a Function . . . . .	12
5.	Sharing Variables Through the VPUT/VGET Service . . . . .	13
6.	Sample Panel Definition . . . . .	16
7.	Sample Panel Display . . . . .	17
8.	Control Flow Between ISPF Services and Libraries or Files . . . . .	18
9.	Sample Table . . . . .	23
10.	Sample ISPF Master Application Menu . . . . .	26
11.	Sample ISPF Primary Option Menu . . . . .	27
12.	Hierarchy of ISPF Libraries . . . . .	32
13.	PDF Primary Option Menu . . . . .	34
14.	Organization of Employee Records Application . . . . .	64





## 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**). Specifically:

- ISPF is a dialog manager that provides control of, and services for, interactive applications.
- 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.

A dialog is a "conversation" between a person (using a full screen terminal) and a computer (executing a program). The user communicates with the application by entering information in response to inquiries (prompts) presented on the screen by the application. Responding to the user's input, the application may invoke a routine, file the user's input in a data base, display information from a data base, or take some other action. After each interaction, the application prompts the user to enter information. The process is repeated until the user ends the dialog.

Capabilities added to ISPF and PDF in Version 2 of these products are listed in Appendix A, "New ISPF Functions" and Appendix B, "New ISPF/PDF Functions." Except as noted in "Migration" on page 46, dialogs that run under ISPF Version 1 will run under Version 2 without change.

## OPERATING ENVIRONMENT

Figure 1 shows the ISPF operating environment. The ISPF dialog manager is, conceptually, an environment-independent extension of the host operating environment. 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. In the figure, User Application A is using PDF facilities, such as EDIT, BROWSE, or library access services.

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

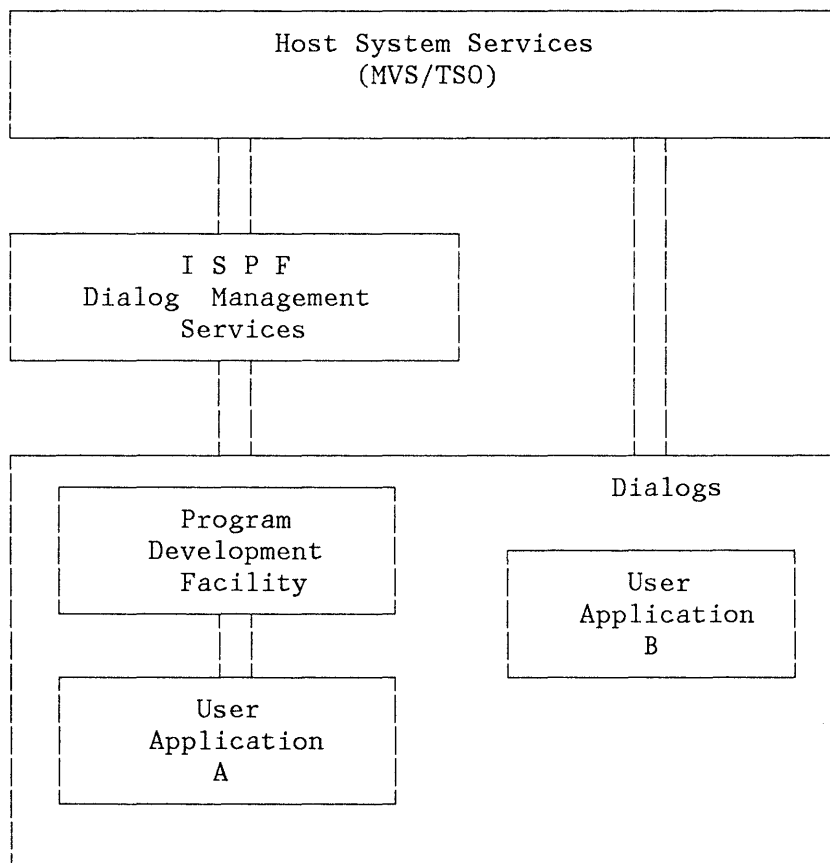


Figure 1. ISPF Operating Environment

---

Figure 2 shows the ISPF application development environment when PDF is used to develop and test a user application dialog.

---

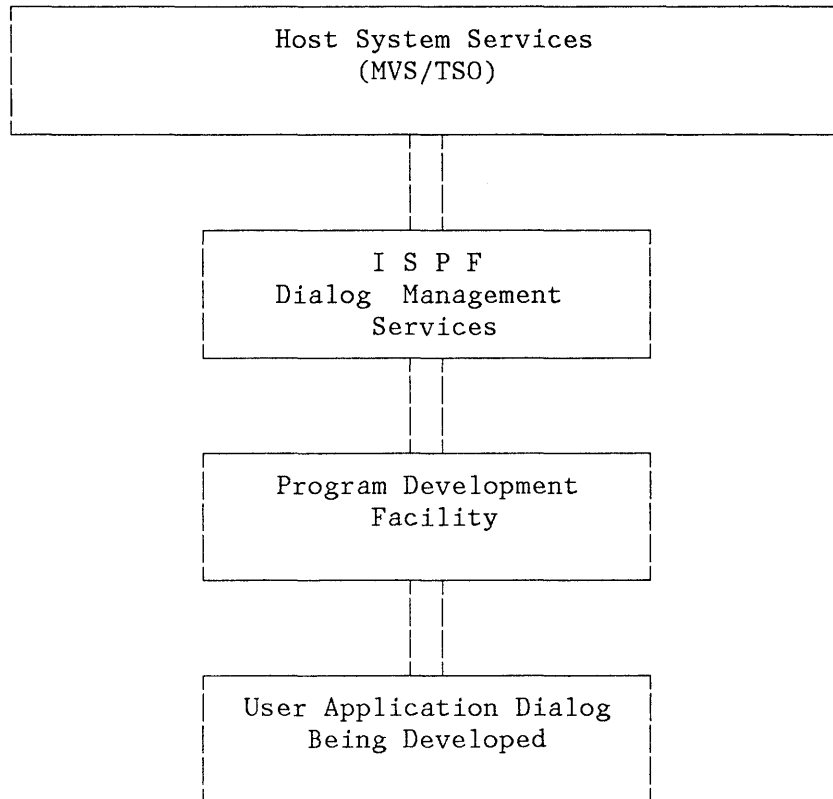


Figure 2. ISPF Application Development Environment

---

## ISPF DIALOG MANAGEMENT

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.

- Data display panels, on which the user receives information from the application.

ISPF applications may take the form of a MVS/TSO CLIST or APL2 procedure, or they may be written in a language such as assembler, PL/I, COBOL, FORTRAN, or PASCAL. See Chapter 4, "Planning" for the program numbers of the compilers that can be used.

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

When a dialog is invoked, ISPF facilities may be used to:

- **Display a hierarchy of menus** based on user selections.
- **Invoke functions** from menus. These functions may be CLISTS or programs.
- **Communicate with the user** through menus, data entry and data display panels, 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.
- **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 or up to four logical areas on a 3290 terminal.
- **Intercept user-entered commands** and take appropriate action based on information contained in command tables.

Version 2 of ISPF includes a number of capabilities not available in Version 1. These capabilities are listed in Appendix A, "New ISPF Functions," and include the following:

- Dialogs may be written as APL2 functions. Dialog services may be executed from the APL2 workspace and APL2 variables may be accessed as the dialog function pool.

- Color and extended highlighting may be specified on panel definitions for use by terminals having those capabilities.
- Displays may be specified for 3290 terminals and may include the capability to fill the entire screen with a single panel or to split the screen vertically as well as horizontally.
- Processing that is to occur in conjunction with the redisplay of a panel may be specified in the )REINIT section, a new section in the panel definition.
- Attributes, specified in the )ATTR section, may be overridden on a field or attribute character basis in the )INIT, )REINIT, or )PROC sections of a panel.
- Table display panels may include statements and attributes available on other types of panels.
- The rows of an existing table may be sorted into a sequence specified by the dialog developer. Later additions to the table may be made so that this sequence is automatically maintained.
- Dialogs may format and display screens using the facilities of the Graphical Data Display Manager (GDDM).
- Dialogs may display panels containing one or more areas that are dynamically generated by the dialog performing the display.

## ISPF/PDF PROGRAM DEVELOPMENT FUNCTIONS

PDF is an ISPF dialog that can help increase programmer productivity by simplifying frequently performed programming tasks. 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** facilities that allow maintenance and tracking of program segments at different versions or levels.
- **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, functions, 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.

- **Interfaces** to standard **language processors** (compilers, assemblers, and linkage editors or loaders). These processors may be invoked as batch jobs and as foreground jobs.
- **Dialog test** facilities that help a user test dialog applications.
- **Documentation preparation assistance** consisting of text editing facilities, and an interface to the Document Composition Facility Program Product.
- **Online tutorial** for instruction and reference. This feature is especially valuable for the new or occasional user.
- **BROWSE, EDIT, and EDIT Recovery services** that may be invoked from user applications.

Version 2 of ISPF/PDF includes a number of capabilities not available in Version 1. These capabilities are listed in Appendix B, "New ISPF/PDF Functions," and include the following:

- A library management facility that controls movement of members between controlled and non-controlled libraries and provides ownership control of members of a controlled ISPF library or a hierarchy of libraries.
- An edit macro language that allows extension of the editing facilities by:
  - Accessing the data being edited
  - Accessing the current cursor position
  - Accessing edit modes and environmental information
  - Accessing external data from a library or data set
  - Invoking any dialog management service directly
- A member parts list that shows, for each specified source program module, the names of the modules it calls or includes and the names of the modules that call or include it. This function may be invoked as either a foreground or background job.
- Library access services that allow access to ISPF libraries, sequential data sets, and partitioned data sets from user applications.

## CHAPTER 2. DIALOG MANAGEMENT

ISPF processes interactive applications called **dialogs**. Dialogs are composed of panel and program elements that tell ISPF about 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 by supplying data or controlling the flow of the application.

The user may partition the display screen into two "logical" screens (up to four logical screens on the 3290 terminal) at any time during a dialog when a command can be invoked. The 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 any time. The location of the cursor is used to identify which of the screens is active.

Split-screen mode is entered by means of a command or a PF key, and is terminated by ending the application on all but one of the logical screens. The remaining logical screen is then expanded to full screen size.

### CONCEPT OF A DIALOG

A dialog is an application that runs under the control of ISPF. A user of a dialog is any person using ISPF and a terminal to process information. (However, an ISPF dialog that does not call for display of information may be run in batch mode.) Typically, a dialog receives requests and data from a user at a terminal and gives an appropriate response. Operations generally provided by interactive applications, and performed while a user is at the terminal, include:

- Identifying processing routines available to the user
- Invoking a requested routine, based on the user's choice
- Prompting the user to enter data
- Waiting for the user to enter the data
- Reading the data into a work area
- Checking the data to verify that it is appropriate for the application



- If the data is not appropriate for the application:
  - Identifying the error to the user
  - Prompting the user to reenter the data and checking the newly entered data for appropriateness
- When the entered data is in the proper form:
  - Obtaining and displaying any information requested by the user and waiting for the user's acknowledgment
  - Processing or storing the user's data for later use and advising the user of the disposition of the data
- Creating sequential output in the form of files or reports
- Providing online documentation, consisting of messages and tutorials, to help the user in processing an application

The developer's task of supplying the above functions is simplified by the use of ISPF.

An example of an ISPF dialog is shown in Appendix C, "Example of an ISPF Dialog."

## PARTS OF A DIALOG

A dialog developer is any person who creates the parts, or elements, of a dialog. The types of elements that make up a dialog are:

- Functions - command procedures or programs that perform processing requested by the user. Functions may invoke ISPF dialog services to display panels and messages, build and maintain tables, generate output files, and control operational modes. Functions are created by the dialog developer.

A developer may use more than one language in a dialog. For example, within a single dialog containing three functions, each function could be written using a different language, such as PL/I, COBOL, and FORTRAN. One (or more) of the functions could be written using a CLIST instead of a programming language.

- Panel definitions - specifications of display images. A panel definition may be a selection panel definition, which controls a display asking the user to choose a processing option; a data entry panel definition, which, controls a display asking the user to enter data; or an information-only panel definition, which controls a display providing information to the user.

Most panels prompt the user for input. The user response may identify which path is to be taken through the dialog, or it may be

interpreted as data. Panel definitions are created by the dialog developer.

- Message definitions - Specifications of data that, when displayed, provide special information to the user. A message may confirm that a user-requested action is in progress or completed, or report an error in the user's input. Messages may be directed to the user's terminal and superimposed on the display to which they apply, to a log file, or both. Message definitions are created by the dialog developer.
- File tailoring skeletons (or simply, skeletons) - generalized representations of sequential data that may be customized during dialog execution to produce output files. After a skeleton is processed, the output file may be used to drive other processes. File skeletons are frequently used to produce job files for batch execution. Skeletons are created by the dialog developer.
- Tables - 2-dimensional arrays that contain data. A table may be created as a temporary data repository, or it may be retained across sessions. A retained table may also be shared among several applications. The type and amount of data stored in a table depends upon the nature of the application. Tables are created by dialog processing.

A dialog need not include all types of elements. In particular, tables and skeletons may not be needed, depending upon the type of application.

Panel definitions, message definitions, and skeletons are stored in libraries prior to execution of the dialog. They are created by a dialog developer by editing directly into ISPF panel, message, or skeleton libraries; no compile or preprocessing step is required for their use.

Tables are generated and updated during dialog execution. The organization of each table is specified to ISPF by the functions that use ISPF.

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

## ORGANIZATION OF A DIALOG

Dialogs generally begin with the display of a selection panel (a menu) or invocation of a dialog function.

A typical dialog organization, shown in Figure 3, starts with display of a selection panel. This menu has been defined to be a primary option menu. A user may choose an option from the primary option menu that causes either the invocation of a dialog function or the display of

another selection panel. (Selection panels invoked after invocation of the primary option menu are referred to as lower-level menus). Each lower-level menu may also pass control to other functions or display still other lower-level menus. A lower-level menu may also be designated to be a primary option menu. In any case, a RETURN command issued on any panel will cause control to be returned to the next higher primary option menu.

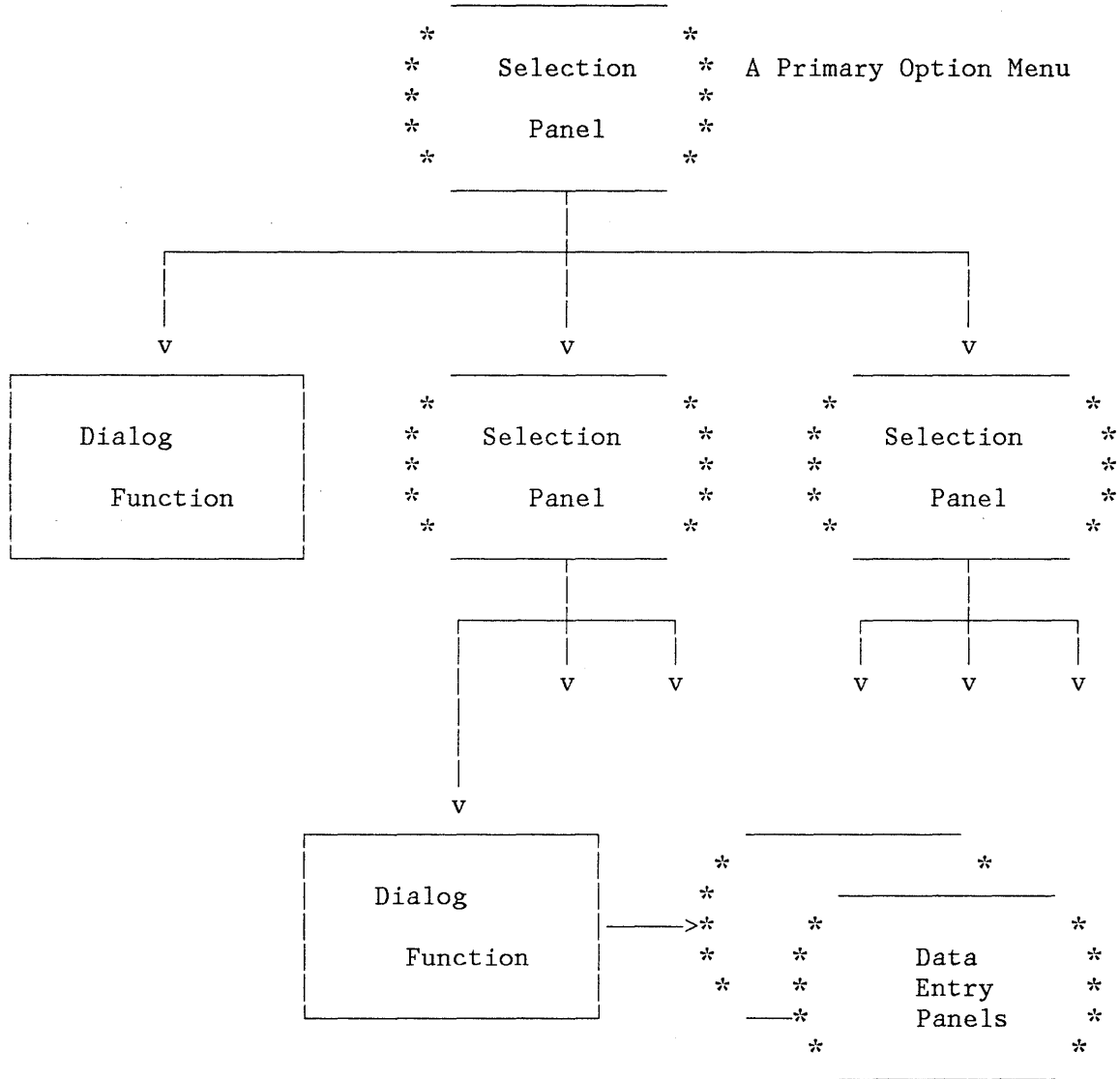


Figure 3. Dialog Starting with a Selection Panel

When a dialog function is in control, it may use any of the dialog services provided by ISPF. When the function completes, and if the

function was invoked from a selection panel, the selection panel is redisplayed. The redisplay of the selection panel provides the opportunity for the user to select another processing option.

A function may be invoked (or selected) from another function. When the selected function completes, control is returned to the function that selected it.

Generally, an EXIT option is provided on a primary option menu. This option, when selected by the user, causes the menu to end. Control is then returned to the function or panel from which the menu was invoked.

Figure 4 shows another dialog. In this dialog, a dialog function is invoked first, before any panels are displayed. The function may perform application-dependent initialization, and display data entry panels to prompt the user for basic information. It may then start the selection process by using the SELECT service to display a selection panel for the application.

As shown, a function may also invoke another function, through the SELECT service, without displaying a panel. This facility 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" on page 24) control the definition and use of dialog variables.

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

- All CLIST and APL2 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.
- 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.

The VCOPY service may be used to access implicit variables directly from an assembled or compiled program.

When a function variable is created, it is associated with the function that is currently in control and may not be directly 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

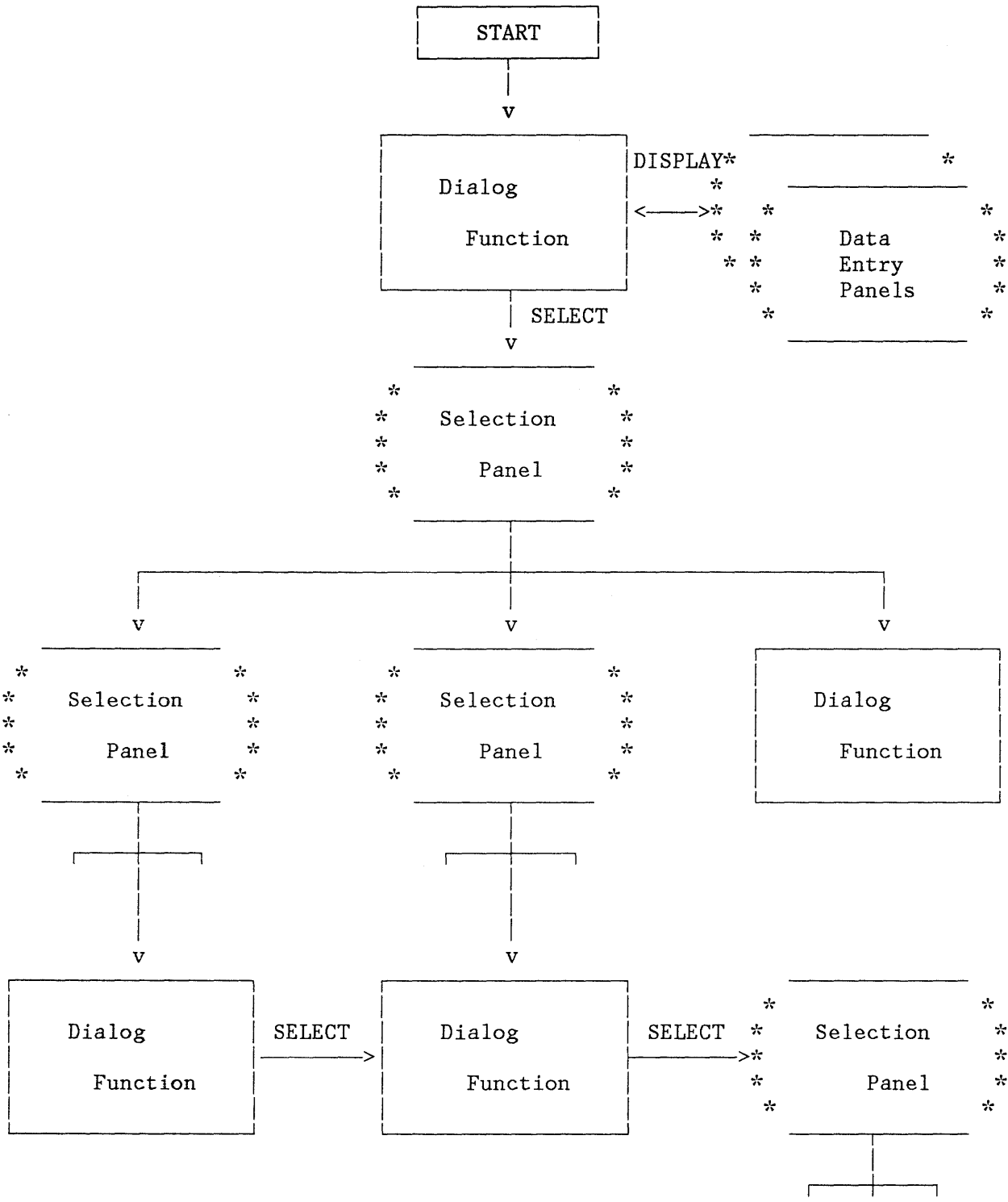


Figure 4. Dialog Starting with a Function

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
- Application profile pool

The shared variable pool allows communication of variables between functions that belong to the same application. 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 application profile pool 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 application profile pool.

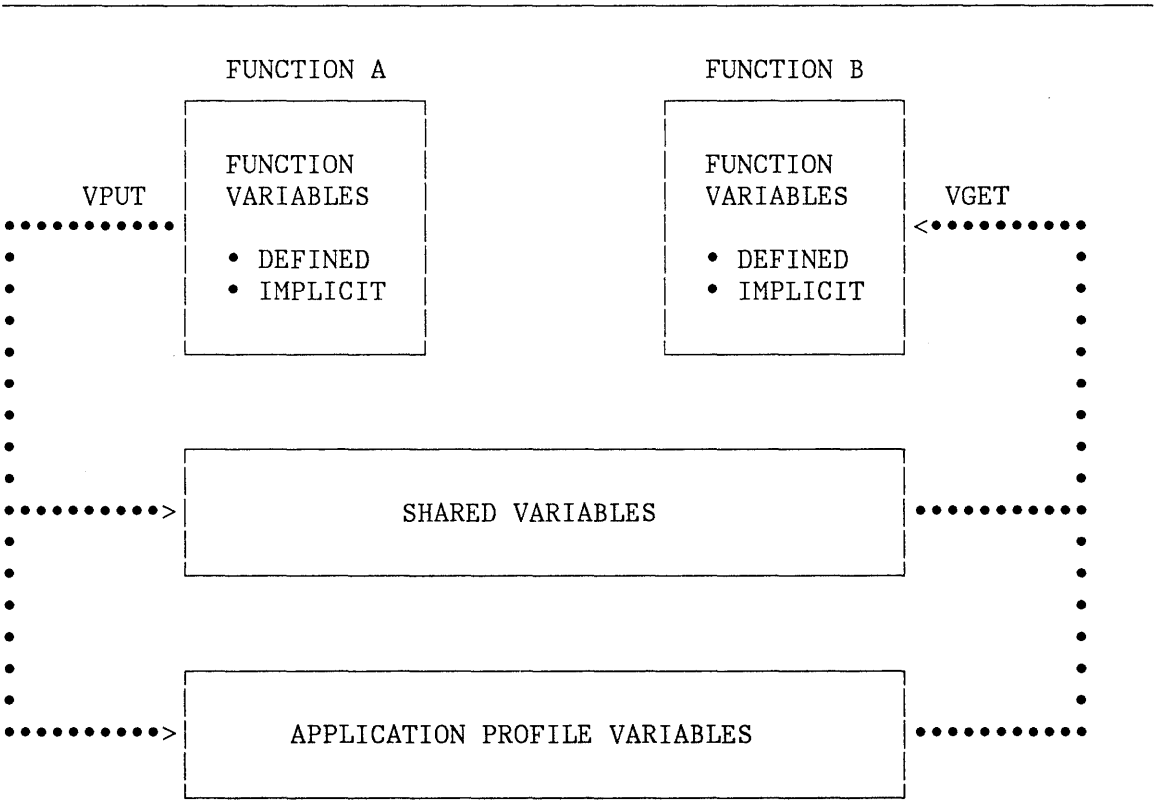


Figure 5. Sharing Variables Through the VPUT/VGET Service

## FORMATTING OF DISPLAYS

The ISPF display services (see "Display Services" on page 19) 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 specified by message definitions created by the dialog developer. A message is generally superimposed on a panel being viewed by a user.
- Panel and message definitions are maintained in libraries separate from dialog functions. This division permits 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.
- 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.
- Reinitialization section (optional) — specifies processing that is to occur prior to the redisplay of a panel.
- 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 C). 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
```

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 either be blank or 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.



---

```

%----- 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
 .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

```

Figure 6. Sample Panel Definition

---

Figure 7 shows how the panel would look when displayed, and after a user had entered a name and address and telephone number.

---

```
----- EMPLOYEE RECORDS -----  
COMMAND ==>_  
  
EMPLOYEE SERIAL: 123456  
  
TYPE OF CHANGE ==> NEW          (NEW, UPDATE, OR DELETE)  
  
EMPLOYEE NAME:  
  LAST   ==> THOMPSON  
  FIRST  ==> FRANK  
  INITIAL ==> T  
  
HOME ADDRESS:  
  LINE 1 ==> 15 ORGAN HILL ROAD  
  LINE 2 ==> APARTMENT C  
  LINE 3 ==> NEW HACKENSACK  
  LINE 4 ==> NEW YORK 12601  
  
HOME PHONE:  
  AREA CODE   ==> 914  
  LOCAL NUMBER ==> 463-1234  
  
-----
```

Figure 7. Sample Panel Display

---

## DIALOG MANAGEMENT SERVICES

ISPF provides the following dialog management services:

- Select service
- Control service
- Log service
- Display services
- Table services
- File tailoring services
- Variable services

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

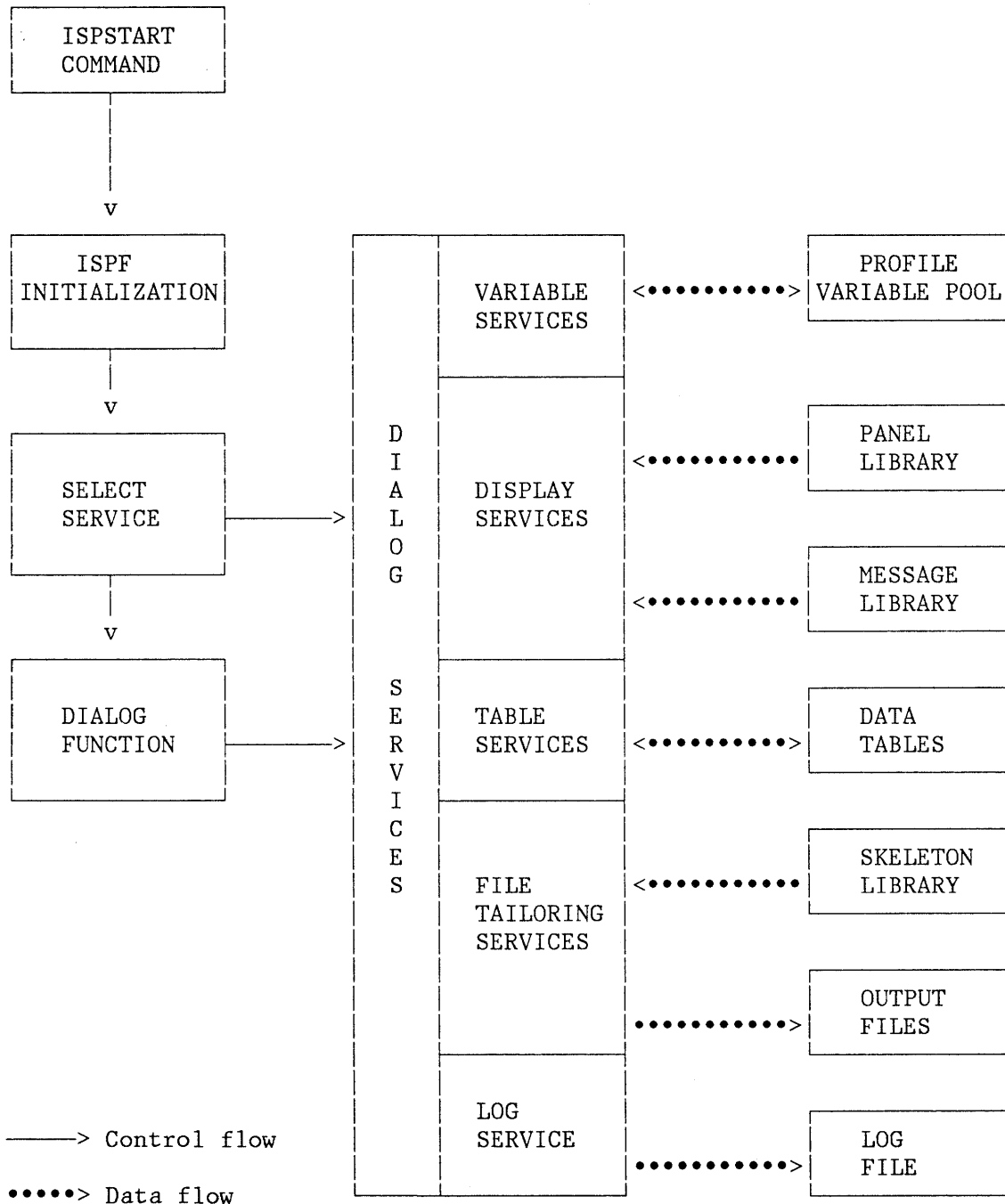


Figure 8. Control Flow Between ISPF Services and Libraries or Files

## 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.

When an ISPF application is first started, 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, SELECT starts the display of a new selection panel hierarchy or it can invoke other functions without displaying a new selection panel.

## Control Service

The control service is used to define certain processing options for the dialog environment. The processing options control the display screen and error processing.

## Log Service

The log service allows a dialog function to write a message to the ISPF log file. The user may specify whether the log is to be printed, kept, or deleted when ISPF is terminated.

## Display Services

Display services allow a dialog to display information and interpret responses from the user. There are four 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)
- Saves, in designated dialog variables, any data entered by the user in input fields on the panel
- Provides a capability to check data entered by a user for validity
- Provides a capability to inform a user of invalid input and prompt for entry of correct input

**TBDISPL** Displays a table. The TBDISPL service:

- Combines column headings from panel definitions with information stored in ISPF tables
- Displays selected columns from all rows or selected rows of a table

- Permits a user to scroll through a displayed table
- Saves, in designated dialog variables, any data entered by a user in input fields on the panel
- Provides a capability to check data entered by the user for validity
- Provides a capability to inform the user of invalid input and prompt for entry of correct input

**SETMSG** Obtains 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.

**GETMSG** Obtains a message and related information and stores them in variables specified in the GETMSG service request.

#### Displays Containing a GDDM Generated Picture or Graph

An ISPF display may contain a picture or graph generated through use of the Graphical Data Display Manager (GDDM) Program Product.

ISPF panel definition syntax allows you to specify a graphic area within a panel. A graphic area may contain either a picture, constructed by use of GDDM services, or a graph, constructed by use of the GDDM Presentation Graphics Feature (PGF).

The dialog uses the ISPF GRINIT and DISPLAY services to initialize and display a panel containing a graphic area. GRINIT establishes an interface to GDDM, which can then be used by an ISPF program function to construct the picture or graph. The DISPLAY service, called by the function, performs the display.

The GRERROR service returns to the caller the address of the GDDM error feedback block and the address of the GDDM parameter descriptor table. The GRTERM service indicates that the caller has completed all GDDM processing and that GDDM may be terminated.

The PQUERY service may be used to obtain type and size information for graphic areas.

## Displays Containing Dynamically Generated Areas

ISPF panel definition syntax allows you to specify one or more dynamic areas within a panel. A display in a dynamic area may consist of alphanumeric or special characters.

The format, as well as the content of a panel's dynamic areas, may be changed during execution of the dialog that displays them. (Generally, when dynamically generated areas are not used and a change to the panel format is to be made, the panel designer must edit the panel definition, make the change, and store the definition in the panel library before executing the dialog that displays the panel.)

Data to be displayed is stored by the dialog function, along with control characters that will control the display, in a dynamic character string. This string may be constructed by the currently executing dialog (dynamically generated) or in an earlier execution of a dialog or program. When completed, the string is stored in a dialog variable called a dynamic string variable. The name of this variable is specified in the panel definition within the dynamic area it is to control.

A dialog uses the DISPLAY, SELECT, or TBDISPL service to display a panel containing a dynamic area. Before the panel is displayed, ISPF uses the value of the dynamic string variable to format the dynamically generated area of the panel. After the display and after entry of any input by the viewer, ISPF stores data from within the dynamic area in the dynamic string variable associated with the area. This data is available for processing by the dialog function.

The PQUERY service may be used to obtain type and size information for dynamic areas.

## Table Services

Table services allow sets of dialog variables to be created, maintained, and accessed in a convenient form. A table is a 2-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.

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. Tables may be sorted and maintained in a sequence specified through use of the TBSORT service.

### General Services

The following services operate on an entire table:

<b>TBCREATE</b>	Creates a new table and opens it for processing.
<b>TBOPEN</b>	Opens an existing (permanent) table for processing.
<b>TBQUERY</b>	Obtains information about a table.
<b>TBSORT</b>	Sorts a table.
<b>TBSTATS</b>	Provides statistics for a table.
<b>TBSAVE</b>	Saves a permanent copy of a table without closing it.
<b>TBCLOSE</b>	Closes a table, and saves a permanent copy if the table was opened in write mode.
<b>TBEND</b>	Closes a table without saving it.
<b>TBERASE</b>	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:

<b>TBADD</b>	Adds a new row to the table.
<b>TBDELETE</b>	Deletes a row from the table.

<b>TBGET</b>	Retrieves a row from the table.
<b>TBPUT</b>	Updates an existing row in the table.
<b>TBMOD</b>	Updates a row in the table if it exists; otherwise, adds a new row to the table.
<b>TBEXIST</b>	Tests for the existence of a row.
<b>TBSCAN</b>	Searches a table for a row that matches a list of "argument" variables, and retrieves that row.
<b>TBSARG</b>	Establishes a new search argument for use with TBSCAN.
<b>TBTOP</b>	Sets the current row pointer to the top (ahead of the first row).
<b>TBBOTTOM</b>	Sets the current row pointer to the last row and then optionally transfers the variables in the last row to the variable pool.
<b>TBSKIP</b>	Moves the current row pointer forward or backward by a specified number of rows, and then optionally transfers the variables in that row to the variable pool.
<b>TBVCLEAR</b>	Sets the dialog variables that correspond to variables in the table to null.

---

<u>EMP SER</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

---

## 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, or 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 variable 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. The plus sign (+) indicates the end of the field.

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:

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

The VGET and VPUT services 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 APL2).

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.

---

```
ISPFB MASTER APPLICATION MENU
```

---

```
OPTION ==> 1_
```

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

Enter END command to terminate ISPF.

Figure 10. Sample ISPF Master Application Menu

---

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	USERID	- Z59JEH
1	COMMANDS	- Create/change command table	TIME	- 16:19
2	.	- (Description for option 2)	TERMINAL	- 3277
3	.	- (Description for option 3)	PF KEYS	- 12
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		

Enter END command to terminate application.

Figure 11. Sample ISPF Primary Option Menu

---



## 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. In addition, PDF can be used to control library members.

### PDF DISPLAYS

A brief scenario of PDF usage is contained in Appendix D, "Example of a PDF Session." 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 the next time the same panel is displayed. The browse entry panel is an example.
3. **Member selection lists**, which display lists 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.

## Screen Size

All ISPF and PDF 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. On a 3290, information is normally displayed in the 24-line by 80-character format. If the 3290 is configured to use more of the screen, the user can choose to display ISPF information using a larger screen size.

## 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. 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. Command and long message lines may be displayed, at the user's option, at the top or bottom of the screen.

The long message line is used to display an explanation of error messages when requested using the HELP command. 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 both user and program access to a particular data set by two basic methods. The first is through a library structure. The second is through the normal data set naming conventions associated with TSO.

### Library Structure

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, group name, and type, where:

- **Project name** is the common identifier for all libraries belonging to a particular project.
- **Group name** identifies the particular set of libraries, such as MASTER, TEST, or JOE.
- **Type** identifies the type of information contained in the library.

PDF uses a hierarchy of libraries to permit effective control of different stages 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 three sets of individual user libraries, a set of test libraries, and a set of master libraries identified by the user working on that library. because the actual identification required in these fields may depend on the operating system for some types of libraries.

If the MASTER and TEST groups have controlled libraries associated with them, the controlled groups must contain the same types (ISPF libraries). See the description of option 8, later in this chapter, and also Library Management for further information.



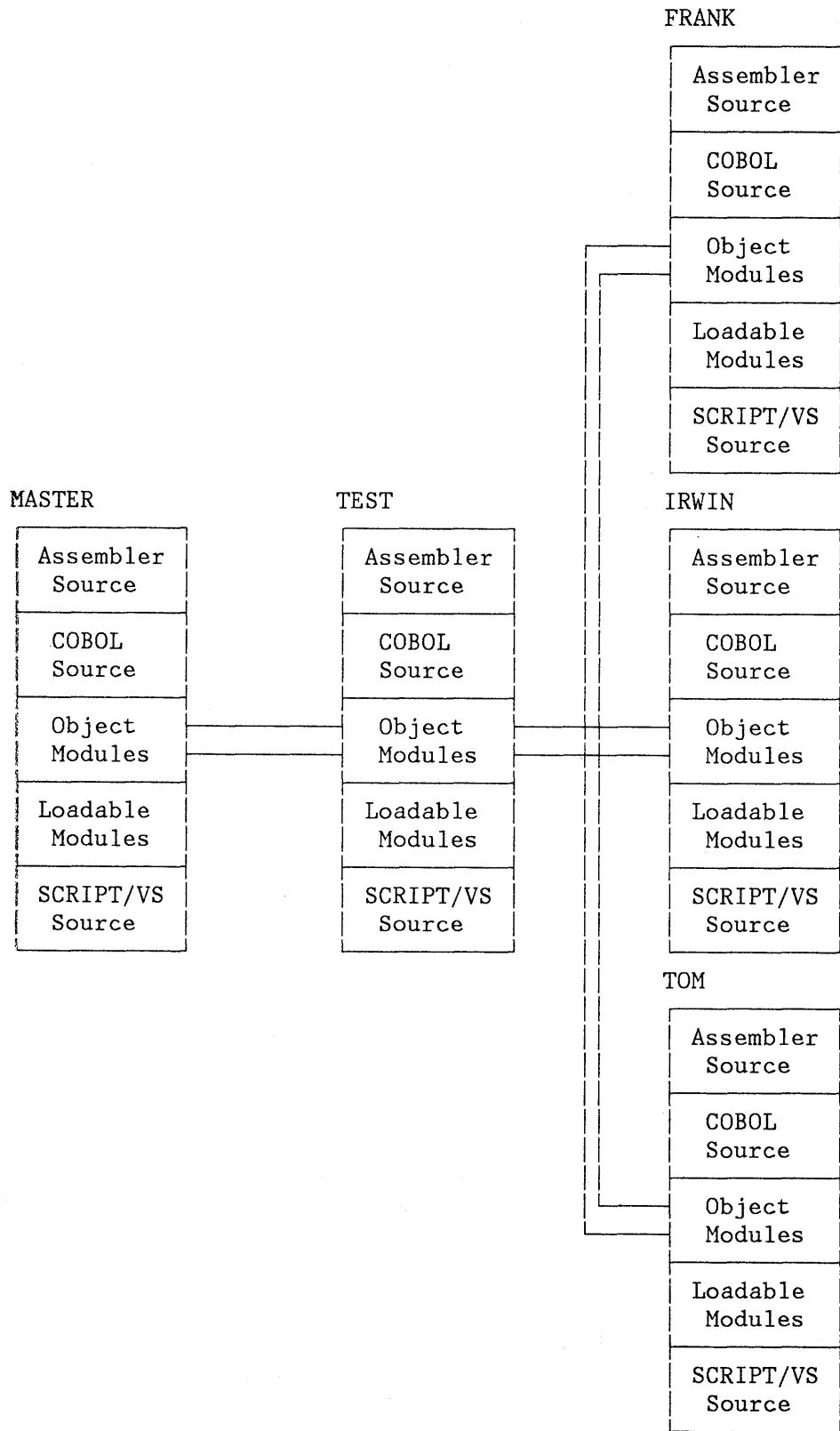


Figure 12. Hierarchy of ISPF Libraries

## Conventional Data Set Naming Conventions

You may refer to individual sequential data sets by using specific naming conventions.

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)
```

## 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.

For options that have lower-level menus (options 0, 3, 4, 5, 7, and 8), 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.

### 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.

---

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	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		
8	LM UTILITIES	- Perform library management utility functions		
C	CHANGES	- Display summary of changes for this release		
T	TUTORIAL	- Display information about ISPF/PDF		
X	EXIT	- Terminate ISPF using list/log defaults		

Enter END command to terminate ISPF.

Figure 13. PDF Primary Option Menu

---

The ISPF PARMS option allows the user to specify:

- Terminal characteristics
- Log and list defaults
- Program function (PF) key assignments
- Display of the command field and long message line at the bottom of the logical screen
- List data set characteristics

### 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.

During browse, 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.

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 or label
<b>FIND</b>	Scroll to a specified character string
<b>CHANGE</b>	Replace a specified character string with another string
<b>COPY</b>	Copy data from another member or file
<b>MOVE</b>	Duplicate data and then delete the member or file
<b>SORT</b>	Sort data within the member or file
<b>DELETE</b>	Delete selected lines within the member or file

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

- **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.
- Column Identification Line — A special line may be displayed anywhere in the data area of the screen to determine column positions.
- 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
  - Rearrangement 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.
- Edit Macros — Users may write macros consisting of edit commands and macro control commands to simplify their normal editing tasks.
- Edit Profiles — The user can define up to 24 different profiles, describing the edit environment, for various types of data.

### 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 or entire files
- Allocate files
- Move, copy, or promote data sets or members
- Display a list of data sets in scrollable format, allowing processing of those data sets directly from the list panel.
- Reset ISPF library statistics

Consult [PDF Reference](#) for details.

### Option 4 - FOREGROUND

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 foreground processing functions 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
- Member parts list

Consult [PDF Reference](#) for 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 and for creation of a member parts list.

## Option 6 - COMMAND

Selecting **COMMAND** allows the user to enter appropriate host system commands (including TSO CLISTS) 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 using 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 (for example, variable data, table data, or a return code from a service) 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 rows in a table, and displaying the status or structure of a table
- 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 8 - LIBRARY MANAGEMENT UTILITIES

Selecting **LIBRARY MANAGEMENT UTILITIES** allows the library administrator to specify library hierarchies and controls for a controlled project. The library administrator can:

- Create, update, or delete promotion hierarchies
- Authorize access to members in library hierarchies
- Authorize access to library controls
- Specify action to be taken when promoting members
- Create, update, or delete lists of users who are authorized to perform library functions
- Create, update, or delete a distribution table that relates different promotion hierarchies to each other
- Activate or deactivate promotion hierarchy controls
- Browse or print activity log and library control information
- Review activity log, control status, and member status listings

## Option C - CHANGES

Selecting **CHANGES** allows the user to review a description of the PDF changes from the previous version.



## 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 programming requirements for ISPF and PDF. Keep in mind that ISPF is a prerequisite for PDF.

ISPF dialogs may be written as CLIST procedures or as APL2 functions. The APL2 program number is 5668-899. Dialogs also may be written as programs for use with the following compilers:

- PL/I Optimizer 5734-PL1
- COBOL 5740-CB1
- VS FORTRAN (Release 3) 5748-FO3
- PASCAL 5796-PNQ

ISPF operates as a non-authorized 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. VSAM is required if dialogs are to be written using APL2. TSO/TCAM or TSO/VTAM are required for terminal communication unless ISPF is operated only in batch mode.

One of the following teleprocessing access methods is required:

ACF/TCAM Version 2 Release 2 (5735-RC3) or later

**Note:** A TCAM MCP generation may be required by the installation to obtain full-screen support in the TSO/TCAM environment.

VTAM 2 (available with MVS)

ACF/VTAM Version 1 Release 2 (5735-RC2) or later for MVS/370 systems

ACF/VTAM Version 2 (5665-280) for MVS/XA systems

**Note:** ACF/TCAM or ACF/VTAM is required with 3278 Model 3, 4, or 5 terminals (with screen sizes larger than 24 by 80). To use the 3290 or color you must have either ACF/VTAM Version 2 Release 1 (or later) or ACF/TCAM Version 2 Release 2. A PTF for ACF/VTAM allows you to use non-partitioned large screen compatibility mode.

A PTF for ACF/TCAM allows you to use the non-partitioned large screen, color, or the 3290.

Internal interfaces are provided to the IBM programs listed below. 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
Graphical Data Display Manager (GDDM) - GDDM Release 3	5748-XXH
(or GDDM Release 4)	
Presentation Graphics Feature (PGF) - PGF Release 3	5748-XXH
(or PGF Release 4)	
TSO Extensions (TSO/E) for MVS/SP Version 1 and 2	5665-285

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 <sup>1</sup> (foreground only)	5734-CP2
COBOL Compiler and Library	5740-CB1
TSO COBOL Prompter (foreground only)	5734-CP1
COBOL Interactive Debug (foreground only)	5734-CB4
VS FORTRAN	5748-FO3
TSO FORTRAN Prompter <sup>2</sup> (foreground only)	5734-CP3
VS FORTRAN Interactive Debug (foreground only)	5668-903
PL/I Checkout Compiler	5734-PL2
PL/I Optimizing Compiler	5734-PL1
PASCAL/VS Compiler	5796-PNQ
Document Composition Facility (SCRIPT/VS) with the Foreground Environment Feature (foreground only)	5748-XX9

The ISPF Version 2 Program Product is a prerequisite for the operation of PDF. The Library Management Facility (LMF) of PDF requires that the following programs be installed:

- OS/VS2 or MVS/SP Version 1.2 or higher
- OS/VS Sort/Merge or the equivalent.

---

<sup>1</sup> Only with VS2 System Assembler

<sup>2</sup> Supported in an MVS/370 environment only

| No additional programming requirements are necessary beyond those of the  
| prerequisite program product. However, to maintain data integrity for  
| the Library Management facility, RACF or its equivalent is recommended.

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.

## MACHINE REQUIREMENTS

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

| Most ISPF program modules for MVS/370 normally reside in the pageable  
| link pack area. Most ISPF program modules for MVX/XA normally reside in  
| both the PLPA and the extended pageable link pack area (EPLPA).

The table below contains the following information:

- 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 size required for ISPF and PDF operation. The second (\*) value shown is the recommended size.
- The size of the pageable link pack area (PLPA) and extended pageable link pack area (EPLPA) that may be used for ISPF and PDF program residency. The use of PLPA and EPLPA is recommended for performance reasons, but not required. Most of the performance benefits may be achieved by using the recommend value shown for that entry in the following table.

	ISPF	PDF
Cylinders of disk storage (3330) <sup>3 4</sup>	15	31
Size in bytes, note <sup>5</sup> of minimum: region (MVS)	512K	200K to 1000K
(recommended)	2 M	2M
Size in bytes, <sup>4</sup> of:		
MVS/370		
PLPA (maximum)	531K	739K
PLPA (recommended)	451K	428K
MVX/XA		
PLPA (maximum)	107K	64K
EPLPA (maximum)	417K	640K
PLPA (recommended)	78K	64K
EPLPA (recommended)	417K	640K

<sup>3</sup> For other device types, use an equivalent amount of space.

<sup>4</sup> For these entries, add the ISPF and PDF figures if PDF is installed.

<sup>5</sup> The minimum region size varies according to how much PDF code is in the LPA. The minimums include only 200K of dynamic storage.

## TERMINAL REQUIREMENTS

ISPF requires one or more of the following IBM terminals:

- 3178
- 3275 Model 2
- 3275 Model 12
- 3276 Model 2, 3, or 4
- 3276 Model 12, 13, or 14
- 3277 Model 2 (local or remote)
- 3278 Model 2, 3, 4, or 5 (local or remote)
- 3279 Model 2 or 3 (local or remote)
- 3290 Information Panel

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

- EBCDIC
- ASCII
- APL
- Katakana

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

- EBCDIC
- ASCII
- 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
- 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.

**Note:** Use of an IBM 3284, 3286, 3287, 3288, or 3289 printer requires installation of the appropriate DSPRINT command processor.

## 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) must be installed to use the ISPF foreground and batch options. The Document Composition Facility must be installed to use the SCRIPT/VS utility.

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.

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

## MIGRATION

Users may migrate to ISPF Version 2 from the System Productivity Facility (SPF) or from ISPF Version 1.

### From SPF

Dialogs that run under the previous SPF Program Product can also be run under the ISPF Version 2 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. You must also consider making revisions as noted in "From ISPF Version 1" on page 47.

Two CLISTs are provided to assist in the migration from SPF, Program Number 5668-009, to ISPF. One CLIST may be used to create a user profile data set if it does not already exist. The other CLIST assists in the conversion of SPF profile data to ISPF profile table data for each specified user. See Installation and Customization for more information.

The following differences between SPF and ISPF must be taken into account when running SPF dialogs under ISPF.

- 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.

### From ISPF Version 1

Except as noted below, dialogs that run under ISPF Version 1 run under ISPF Version 2 without change.

Since the data set protection philosophy and the interfaces to system services and DASD are unchanged in Version 2, both Version 1 and Version



2 may execute concurrently in two or more TSO address spaces. See Installation and Customization for more information.

With the release of ISPF Version 2, support is being withdrawn for the Structured Programming Facility (old SPF) messages, panels, and procedures (procs). Any user applications that are dependent on the old SPF message, panel, or proc formats will need to be converted to the ISPF format of messages, panels, and skeletons. The Convert Menus/Message facility (option 3.10) of ISPF/PDF is available to assist the user in this conversion effort.

The option 0.2 and termination panels of ISPF are changed to use the same termination options as the PDF foreground options. This change improves usability throughout the product. Any user specified options stored in the profile variable pool that differ from the new options are converted automatically and there will be no migration impact on the user.

There are some minor processing changes in the Table Display dialog service. Previously, an empty table could not be displayed. Version 2 does not treat an empty table as an error situation, but displays the panel with no table data lines. For table display processing, unlike Version 1, Version 2 does not ignore the keywords that control the field justification and capitalization processing.

## CONVERTING DIALOGS FROM VM OR VSE TO MVS

If you want to convert existing VM or VSE dialogs to run under ISPF in MVS, consider the following:

- A dialog functions written as a CMS EXEC must be converted to a command procedure or a programming language supported by ISPF in MVS.
- Any commands or services invoked by the original dialog must be converted to an equivalent command or service in MVS.
- Any file skeletons that generate JCL (JECL) or commands must be converted to the appropriate JCL or commands for MVS.
- All panels, messages, skeletons, tables, etc., must be moved to the appropriate libraries in MVS.

## PUBLICATIONS

The following publications are available with ISPF:

- Dialog Management Services (SC34-4021) - Provides detailed information on the use of dialog management services in the development of interactive applications.

- Dialog Management Services Examples (SC34-4022) - Supplements the above manual; provides examples of the use of dialog management services in the development of interactive applications.
- Installation and Customization (SC34-4019) - Provides detailed information on how to install and custom tailor ISPF and ISPF/PDF.
- Diagnosis (SC34-4020) - Provides technical information to aid in the evaluation of problems related to ISPF and PDF.

The following additional publications are available with ISPF/PDF:

- Reference (SC34-4025) - Provides detailed information on how to use the Program Development Facility.
- Services (SC34-4023) - Provides information on how to use the dialog services available in PDF.
- Edit Macros (SC34-4018) - Provides information on how to use the edit macros available in PDF.
- Library Management (SC34-4025) - Provides information on how to create the library controls available with library management, and how the controlled libraries are used in a development process.



## APPENDIX A. NEW ISPF FUNCTIONS

New functions for ISPF Version 2 are listed below.

- 3290 Terminal Support
  - Provides the capability to address all screen positions on the 3290 terminal (16-bit addressing).
  - Provides multiway split screen capability, including a vertical split as well as horizontal splitting. The hardware partitioning mechanism is used to support split screen on this terminal.
  - Allows use of extended highlighting and graphics, using the same mechanisms as for 3279 terminals (see below).
- Extended 3279 Terminal Support
  - Allows use of color and extended highlighting at the field level in panel definitions.
  - Provides graphic support through an interface to GDDM.
- General Panel Enhancements
  - PQUERY Service

The PQUERY service returns information for a specified area on a specific panel. The type (DYNAMIC or GRAPHIC) and size characteristics associated with the area are returned in variables.
  - Extended Attributes

Provides new attribute keywords to permit use of the 3279 terminal features of color and highlighting at the field level.
  - New )REINIT Section

Provides a means of specifying the processing to occur before a redisplay of a panel.
  - Attribute Override

Provides the capability to dynamically change attributes for a single field or those for an attribute character within the )INIT, )PROC, or )REINIT section through an assignment statement.

- Refreshing of Field Values When Redisplaying
 

Provides the capability for reassigning or refreshing the values of given fields on the screen at the time of a redisplay (when the contents of the screen are not rebuilt and the )INIT section is not reprocessed).
- Floating Command/Message Area
 

Provides the capability of placing the command and long message areas at the top or bottom of the panel as a user-definable option.
- Additional Cursor Placement Capability
 

Provides the capability of additionally specifying the cursor position within the given cursor field.

Provides the capability of returning cursor information to the dialog, field name and position within the field.
- Expanded Panels
 

Extensions to panel syntax provide the capability to expand panels to a width greater than their physical size in the panel library.
- Miscellaneous Panel Enhancements
  - Easing of Syntax Rules and Restrictions
 

Provides the capability of allowing all panel statements, variable names, keywords and keyword values to be entered in either uppercase or lowercase.
  - Extended Verification Function
 

Extends alphabetic verification (such as type of ALPHA or that within a picture specification) to recognize values entered in lowercase as well as uppercase.
  - Extended Pad Character Specification
 

Provides more control over the pad character specification and display, including the capability to specify something other than N (for nulls) and B (blanks) as a general user-supplied padding character. In addition, the capability to suppress the display of a pad character, when a panel field has an initial value.
  - Easing of Field Length Limitation
 

Provides the capability of defining panel input/output fields of a length greater than 255 characters.

- New Alarm Control Variable

Allows the audible alarm control variable .ALARM to be set within an assignment statement in the panel definition. Setting of this variable causes the alarm to sound when the panel is displayed without a message being required.

- Wider Panels

Provides the capability of creating panel definitions of a width greater than 80 characters for larger terminals such as the 3290 terminal.

- Dynamic Screen Generation

Provides a means for a dialog to format a portion (or all) of the panel dynamically. Support is made available for specifying a dynamic area within a panel, for specifying field attributes within the area, and for filling the area with data. On return from the display service, a means is made available of indicating which fields within the area have been modified.

- GDDM Interface

Provides the capability for ISPF dialogs to display graphic images within the framework of the ISPF panel architecture. A single "graphic area" can be specified within a panel. The necessary extensions to the panel syntax have been made within the )ATTR and )BODY section in order to support this function. A graphic area may contain either a picture, constructed via GDDM based services, or a graph, constructed via the non-interactive PGF (Presentation Graphics Feature) services. A new interface also allows ISPF screen snapshots to be written on GDDM supported printers.

- Terminal Interface Information

- Provides the capability of returning a modification indication for input fields within a dynamically formatted area to the dialog.
- Provides the capability of returning PF key assignment information to the dialog.
- Provides the capability of returning the physical and logical screen size (width and depth), scroll amount and terminal type data to the dialog.

- Table Display Service

Table Display is enhanced to remove most of the current restrictions on table display panels, so that they may include statements and attributes available on other types of panels. The new capabilities are summarized below.

- Any attributes may be associated with any fields, including fields within the model line(s).
- Any number of input/output fields may be included within the panel body. Alternate command/message fields may be defined within the non-scrollable portion of the panel. The scroll amount field is now optional.
- A )PROC and/or )REINIT section may be included.

If a )PROC section is included, it will be processed each time TBDISPL is called, regardless of whether the panel name is specified. It will also be processed for a redisplay that occurs as a result of verification failure.

If a )REINIT section is included, it will be processed each time TBDISPL is invoked without the panel name. This holds true for a request to redisplay the table panel with a message or a request to process the next selection. The )REINIT section will also be processed for a redisplay that occurs as a result of verification failure.

- Any of the control variables may be assigned within the various sections.

A new table display control variable, .CSRROW, may also be set in the )INIT or )REINIT section to indicate a particular row where the cursor is to be placed. The rules that apply are the same as those for setting the row through the CSRROW parameter.

- An empty table (a table without any rows) may be displayed. Previously, this resulted in a severe error.
- There four new TBDISPL parameters: CSRPOS (cursor position), AUTOSEL (used in conjunction with CSRROW to control "auto-selection"), POSITION (returns the CRP value) and ROWID (returns the row identifier). The CSRPOS and AUTOSEL parameters have corresponding control variables.
- In addition to ZTDTOP there are two new system variables that are created by TBDISPL to give information to the dialog. They are ZTDSELS (number of selected rows) and ZTDROWS (number of rows in the table).
- There are two new system variables that are created by the dialog to give information to the TBDISPL service. They are ZTDMARK (used to define an alternate Bottom-of-Data marker) and ZTDMSG (used to define an alternate top-row-displayed indicator).
- Model lines may be specified dynamically through the use of variable model lines.

- Table Services

The new table service functions are as follows:

- The capability to sort a table is provided. Rows added or revised may be inserted or moved in the table to maintain the table in a predefined sequence.
- The TBSCAN service is extended to allow for the specification of a search "direction" and a "logical-relationship" between the argument and row data that should terminate the scan.
- A new TBSTATS service is provided to retain status information regarding the contents and usage of a table.
- A POSITION parameter has been added to all dialog services (TBBOTTOM, TBDISPL, TBGET, TBSCAN, and TBSKIP) that read table rows. The POSITION parameter returns the current row pointer (CRP) value at the completion of the service.
- A NOREAD parameter has been added to all dialog services (TBBOTTOM, TBGET, TBSCAN, and TBSKIP) that read table rows. The CRP is moved to the selected row but no data is transferred to the variables in the function pool.
- A LIBRARY parameter on the TBCREATE and TBOPEN services allows the dialog to specify an optional input source for the table.
- A ROWID parameter is provided on the TBBOTTOM, TBDISPL, TBGET, TBSCAN, and TBSKIP services to return a row-identifier. This row-identifier can then be specified in a ROW parameter on TBSKIP to cause positioning to the row without regard to the rows relative position in the table.
- A SHARE parameter on the TBOPEN and TBCREATE services allows a table to be shared between logical screens.

- File Tailoring

The file tailoring facility of ISPF is extended to permit output records sizes of other than 80 characters. This support is provided for all sequential data sets for which the logical record size (LRECL) does not exceed 255 bytes. Both fixed and variable length records are supported. In addition, the capability is provided to insert one or more blank lines in an output file. The number of tab stops is increased to 16 (from 8).

- ISPEXEC Program Interface

Provides the ability for a program dialog to invoke dialog services by passing a character string which syntactically matches the ISPEXEC requirements. All services that can be requested through the use of ISPEXEC command procedure statements are valid through this type of call interface.



- APL Support

The main points of the APL support are:

- APL functions can be selected from a panel, command table, or another dialog function. The SELECT implies the startup of APL, the loading of the designated work space, and the passing of control to the APL function in the work space.
- An APL function executing within the ISPF environment can call on ISPF services by executing a monadic APL function which accepts a character string containing the name of the service and all necessary parameters. The format is the same as passed via the ISPEXEC service by CLISTS).
- The variables in the APL work space become function variables for ISPF. All rules for function variables apply to variables in the APL work space.

- Panel Update Utility

The panel update utility is enhanced to provide the following new functions:

- Deletion of an existing option from a selection panel. A deletion is recognized by a keyword field which matches a current keyword on the panel, and a select string which is blank.
- Replacement of an existing option in a selection panel with a new option. A replacement option is recognized by a keyword field which matches a current keyword on the panel, and a select string which is nonblank.
- Generation of a sequential file that is used to record the changes that were made to panels and that may be used later to back out these changes.
- Acceptance of either an ISPF table (as is currently done in ISPF Version 1 Release 1 or the above sequential file as input to control the the update process.

- MVS Environment Extensions

- Authorized TSO Command and Program Support

ISPF permits the invocation of authorized commands and programs in the MVS/TSO environment. Authorized commands and programs can be invoked wherever non-authorized commands can be invoked.

- TSO Subtasking Support

The ISPLINK interface and CALL ISPEXEC interface may be invoked at any task level below the logical screen task. However,

subtasking support applies only to dialogs that are selected through the use of the CMD parameter. There are no changes to the external interface.

- Virtual Storage Constraint Relief (VSCR)

Most of the ISPF load modules will be loaded into the extended address space. This significantly reduces the load on the usage of storage below 16 megabytes.

- Dialog Execution

In an MVS/XA environment, dialog function modules residing below or above the 16 megabyte line can issue ISPF dialog services. The dialog interface module, ISPLINK (and alias entry points ISPLNK, ISPEXEC, and ISPEX), has the attributes RMODE(24) and AMODE(ANY). These attributes allow both 24 and 31 bit addressing mode callers. You must use the new level of ISPLINK to issue dialog services from above the 16 megabyte line.

Data areas placed above the 16 megabyte line are not supported and the results of processing these data areas will be unpredictable.

Existing dialogs that use previous levels of the ISPLINK module will continue to execute below the 16 megabyte line. These dialogs will continue to function without any changes.

• Miscellaneous Enhancements

The capability is provided to retrieve a message from the message library and place its text in dialog variables.



## APPENDIX B. NEW ISPF/PDF FUNCTIONS

New functions for ISPF/PDF Version 2 are listed below:

- Edit Macros

The editor has been extended to process macros. These macros can override built-in commands. An "initial" macro can be invoked to perform tasks prior to or instead of displaying the edited data on the user's screen. Edit macros may take the form of a MVS/TSO CLIST or may be written in a language such as PL/I, COBOL, FORTRAN, or PASCAL.

- Library Management

This facility gives the user precise control over his development cycle. Members of a controlled library can be updated only if the user is authorized by the system. Libraries can be activated and deactivated selectively and can be related by predefining a hierarchical structure.

- Library Access Services

The library access services provide an interface to ISPF libraries and to certain other system data sets. With a few exceptions, the library access services are system-independent and may be used by any ISPF dialog.

The library access services may be used with the following types of data sets:

- An ISPF library known by project, group, and type
- A concatenated set of ISPF libraries
- A single, existing partitioned or sequential data set or a preallocated VIO data set

The data sets must have record types F, FB, V, VB, or U, and reside on a single DASD volume.

The library access services may be invoked by a program function dialog through a CALL to ISPEXEC or ISPLINK, or by a command function dialog (CLIST) through the ISPEXEC interface. The following services are available:

LMCLOSE        Closes a data set.

LMERASE        Deletes an ISPF library or MVS data set with a 3-level qualified data set name.

LMFREE	Releases the data set associated with a given data id.
LMGET	Reads one record of a data set.
LMINIT	Associates one or more ISPF libraries or an existing data set with a data id. Thereafter, this data id is used to identify the data set for processing by other library access services.
LMMADD	Adds a member to an ISPF library or a partitioned data set.
LMMDEL	Deletes a member of an ISPF library or a partitioned data set.
LMMFIND	Finds a member of an ISPF library or a partitioned data set and optionally locks the member.
LMMLIST	Creates a member list of an ISPF library or a partitioned data set.
LMMREN	Renames a member of an ISPF library or a partitioned data set.
LMMREP	Replaces a member of an ISPF library or a partitioned data set.
LMOPEN	Opens a data set.
LMPROM	Promotes a member or a sequential data set to a controlled ISPF library.
LMPUT	Writes one record of a data set.
LMQUERY	Provides requested information regarding the data set associated with a given data id.
LMRENAME	Renames an ISPF library.

- Member Parts List

This facility shows the relationship between the parts of a program, and also distinguishes between a CALL or an INCLUDE to the named part.

- Packed Data

The members of ISPF libraries may coexist in packed or unpacked format. The current state of a member may be changed (for example, from packed to unpacked) when using Edit, Copy, or the Library Management Facility.

- New SORT and VERSION Commands Within Edit

Within PDF edit, the user can change the version number of the member without affecting the modification number. Also, the SORT command permits a reordering of data within the member in ascending or descending order using up to five separate sort fields.

- New Edit/Browse User-defined Panel

A dialog may use a customized panel to display other fields that are meaningful to the user. In this way, each dialog can display data unique to it.

- Recursive Edit/Browse Invocation

A user may invoke an EDIT command (when in edit) or a BROWSE command (when in browse). Effectively, the previous data set or member is "pushed down" in a stack until the current edit or browse is ended.

- Data Set List Utility

This utility produces a scrollable list of data sets from which the user may select items such as browse, edit, print, rename, and delete. These functions and selections are all initiated from the Data Set List panel.

- Concatenation of Libraries for Browse

For added flexibility when searching for a member, Browse allows up to four libraries to be concatenated.

- FORTRAN/VS

The FORTRAN/VS compiler is invoked in place of the FORTRAN G1 compiler for both foreground and batch processing.

- New Format Panels and Messages

All old-style panels and messages are converted to the new more versatile panel and message format. The convert utility to allow customers to convert to the new format is included in PDF.

- Edit Recovery

Edit backup and recovery is controlled by an edit recovery table that determines the number of recursion levels allowed. An installation may modify this table to increase or decrease the number of entries.

- Profile Locking

Specifying PROFILE LOCK allows the changes to the user's edit profile to be treated as read-only. That is, the changes made during a session are not stored in the permanent edit profile. Specifying the NOLOCK operand causes all changes to be stored, as in previous releases.

- Extended Move/Copy Member List

While under Edit, you can copy or move another data set into the current data set. If you do not specify the member name on the extended move/copy panel, a member list is displayed.

- Recovery from Edit Service

When ISPEXEC or ISPLINK invoke the Edit Service, and Edit is running in recovery mode, the normal EDIT recovery panel (option 2) is not displayed. It is the user's responsibility to do the recovery within the dialog. The EDREC service, with QUERY specified, can be used for the recovery.

## APPENDIX C. 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 ISPF Dialog Management Services Examples for the precise 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 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".



## 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.

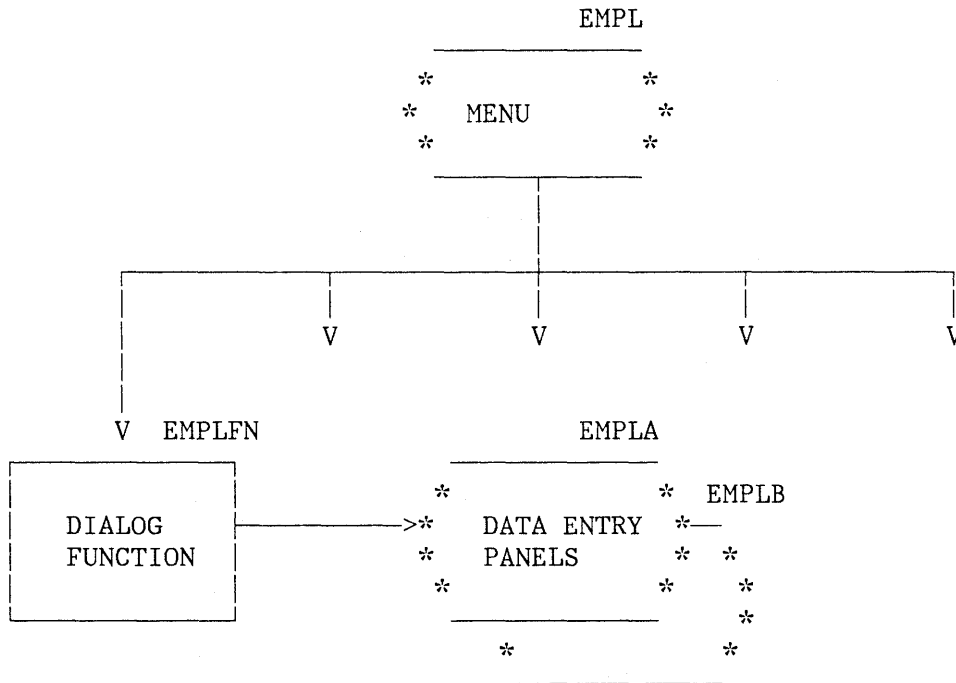


Figure 14. Organization of Employee Records Application

---

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).

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), TBPUT (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.

## DIALOG COMPONENTS

This section defines the components of the dialog. Both the definition and the visible appearance of the panels are included.

### 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 ', '(EMPSER)', /*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 = ' ' ;                                  /* */
            I = ' ' ;                                      /* INITIALIZE */
            ADDR1 = ' ' ;                                  /* PANEL */
            ADDR2 = ' ' ;                                  /* VARIABLES */
            ADDR3 = ' ' ;                                  /* TO NULLS */
            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) ;

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(EMPSER) ;
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', '(EMP SER)', EMP SER, 'CHAR', LEMP SER) ;
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', '(EMP SER)') ;
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)') ;

```

## 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

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 D. 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                       |          |          |
| 8 | LM UTILITIES | - Perform library management utility functions |          |          |
| C | CHANGES      | - Display summary of changes for this release  |          |          |
| T | TUTORIAL     | - Display information about ISPF/PDF           |          |          |
| X | EXIT         | - Terminate ISPF using list/log 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, copy, or promote members or data sets     |  |
| 4  | DSLIS     | - Data set list:                                  |  |
|    |           |   | Print or display (to process) list of data set names |
|    |           |   | Print or display VTOC information                    |
| 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    S - Data set information (short)
blank - Data set information

ISPF LIBRARY:
PROJECT ==> z59jeh
GROUP   ==> 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
GROUP   ==> sample
TYPE    ==> obj

```

```

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

DATA SET NAME: Z59JEH.SAMPLE.ASM

VOLUME SERIAL   ==> (Blank for authorized default volume)
GENERIC UNIT    ==> (Generic group name or unit address)
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
EXPIRATION DATE ==> (yy/mm/dd or blank)

```

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.

MOVE/COPY UTILITY

OPTION ==> C

C - Copy data set or member(s)	CP - Copy and print
M - Move data set or member(s)	MP - Move and print
P - Promote data set or member(s)	PP - Promote and print

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

FROM ISPF LIBRARY:

PROJECT ==>  
 GROUP ==>  
 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)

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.

COPY --- FROM SYS1.SAMPLIB(ISPASM)

COMMAND ==>

SPECIFY "TO" DATA SET BELOW.

TO ISPF LIBRARY:

PROJECT ==> z59jeh  
 GROUP ==> 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)  
 SPECIFY PACK OPTION FOR "TO" DATA SET ==> NO (YES, NO, or BLANK)

<p>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.</p> <p>For this example, the user selects 2 to edit the data set. This results in a display of the edit entry panel.</p> <p>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.</p>	<pre> ----- EDIT - ENTRY PANEL ----- COMMAND ==&gt; _ ISPF LIBRARY: PROJECT ==&gt; z59jeh GROUP   ==&gt; sample   ==&gt;           ==&gt;           ==&gt; TYPE    ==&gt; asm MEMBER  ==&gt; israsm      (Blank for member selection list)  OTHER PARTITIONED OR SEQUENTIAL DATA SET: DATA SET NAME ==&gt; VOLUME SERIAL ==&gt;      (If not cataloged)  DATA SET PASSWORD ==&gt;      (If password protected)  PROFILE NAME   ==&gt;      (Blank defaults to data set type)  INITIAL MACRO  ==&gt; </pre>
<p>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.</p> <p>There are also various commands that assist in locating data to be changed, and in making more complex changes.</p> <p>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.</p>	<pre> EDIT --- Z59JEH.SAMPLE.ASM(ISRASM) ----- COLUMNS 001 072 COMMAND ==&gt; _                               SCROLL ==&gt; HALF ***** ***** TOP OF DATA ***** ==MSG&gt; -CAUTION- PROFILE IS SET TO "STATS ON". STATISTICS DID NOT EXIST FOR ==MSG&gt; THIS MEMBER, BUT WILL BE GENERATED IF DATA IS SAVED. 000100 ASMTEST CSECT 000200          DC      F'01' 000300          DC      F'02' 000400          DC      F'03' 000500          DC      F'04' 000600          DC      F'05' 000700          DC      F'06' 000800          DC      F'07' 000900          DC      F'08'          COMMENT 001000          DC      F'09' 001100          DC      F'10' 001200          DC      F'11' 001300          DC      F'12' 001400          DC      F'13' 001500          DC      F'14' 001600          DC      F'15' 001700          DC      F'16' 001800          DC      F'17'          COMMENT 001900          DC      F'18' </pre>



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         | 7 - Linkage editor             |
| 2 - OS/VS COBOL compiler     | 9 - SCRIPT/VS                  |
| 3 - VS FORTRAN compiler      | 10 - COBOL interactive debug   |
| 4 - PL/I checkout compiler   | 11 - FORTRAN interactive debug |
| 5 - PL/I optimizing compiler | 12 - Member parts list         |
| 6 - PASCAL/VS compiler       |                                |

SOURCE DATA PACKED ==> (YES or NO)

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

----- FOREGROUND ASSEMBLY -----

COMMAND ==>

ISPF LIBRARY:

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

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

LIST ID ==> asmtest1 PASSWORD ==>

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

ADDITIONAL INPUT LIBRARIES:

==>  
==>

<p>When the assembly has been completed, PDF automatically calls the BROWSE function for the assembly listing.</p>	<pre> BROWSE - Z59JEH.ASMTEST1.LIST ----- LINE 000000 COL 001 080 COMMAND ===&gt; _ ***** TOP OF DATA *****-CAPS ON-** </pre>
<p>When the user has finished browsing the assembler output, he enters an END command and is then presented with the foreground print options panel.</p>	<pre>       LOC  OBJECT CODE      ADDR1 ADDR2  STMT  SOURCE STATEMENT 00000 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' </pre>
<p>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.</p>	<pre> ----- FOREGROUND PRINT OPTIONS ----- OPTION ===&gt;        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.ISRASM.LIST  SYSOUT CLASS ===&gt;                               (For system printer) PRINTER ID   ===&gt;                               (For 328x printer)  JOB STATEMENT INFORMATION:                      (Required for system printer) ==&gt; //Z59JEHC JOB (ACCOUNT),'NAME' ==&gt; //* ==&gt; //* ==&gt; //* </pre>

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		
8	LM UTILITIES	- Perform library management utility functions		
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.

## 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 additional definitions, refer to the IBM Vocabulary for Data Processing, Telecommunications, and Office Systems GC20-1699.

**activity log.** As it pertains to the library management utility: A set of records that contains information describing the retrieval of a member out of the controlled library for the purpose of updating that member and the promotion of a member into and within a controlled library.

**application.** One or more dialogs, each of which have been started using the same application-id.

**application-id.** A 1- to 4-character code, used on ISPSTART and SELECT, that identifies the application that is being invoked.

**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.

**controlled group.** As it pertains to the library management utility: A collection of controlled libraries having a common project and group name.

**controlled library.** As it pertains to the library management utility: An ISPF library that has been identified as a library to which users must have specific authority for access for updates and promotion activity. The identification is usually made by the library administrator.

**controlled project.** As it pertains to the library management utility: A collection of controlled groups having a common project name.

**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)

**distribution table.** As it pertains to the library management utility: A table defined by the library administrator and associated with a library to specify

where a member is to be distributed when promoted in that library.

**file.** A sequential data set.

**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 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.

**library administrator.** As it pertains to the library management utility: A person who creates a controlled project and specifies controls for the groups, libraries, and members associated with that project.

**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, PASCAL, 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)

**promote.** As it pertains to the library management utility: The action by which a member is moved into or within a controlled library.

**promotion hierarchy.** As it pertains to the library management utility: An ordered set of related controlled libraries that a member is promoted through during either a development or maintenance process.

**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

- + (plus sign)
  - attribute character 15
- & (ampersand)
  - substitutable variable 15
- &ZIND variable 47
- &ZPRIM variable 47
- )INIT
  - header statement 15
- )PROC
  - header statement 15
- % (percent sign)
  - attribute character 15
- \_ (underscore)
  - attribute character 15, 25

### A

- access
  - data set 31
- access methods
  - BPAM, BSAM, TCAM, VTAM 41, 42
- access services
  - library 6, 59
- activity log 39
  - definition 83
- additions
  - multiple 5
- allocating
  - files (MVS)
    - under utilities 37
- ampersand
  - substitutable variable 15
- APL
  - display character set 45
- APL2 41
  - variables 11
- application
  - definition 83
- application development environment 2
- application-id
  - definition 83

- ASCII
  - display character set 45
- assembler 42
  - foreground processor 37
    - linkage to 6
- assembler prompter 42
- attribute character
  - default 15
- attribute section
  - panel definition 14
- audible alarm 45

### B

- batch
  - member parts list 6, 60
    - PDF option 38
- batch jobs 6
- body
  - panel definition 14
- BPAM 41, 42
- breakpoint
  - definition 83
  - dialog test 38
- breakpoints
  - managing 39
- BROWSE
  - PDF option 34
  - recursive invocation 61
- BROWSE service 6
- browsing ISPF log 38
- BSAM 41, 42
- building user tables 4
- bulk entry of text 36

### C

- Canadian-French
  - display character set 45
- CANCEL
  - edit command 35
- CHANGE



- edit command 35
- change activity
  - history 36
- changes
  - multiple 5
  - PDF option 39
- checkout
  - PL/I
    - compiler 42
- CLIST
  - definition 83
  - MVS 25
  - TSO 4
  - variables 11
- COBOL 42
  - foreground processor 37
  - Interactive Debug 42
  - prompter 42
- column boundaries 36
- column identification line 36
- COMMAND
  - ISPSTART
    - employee records 64
    - PDF option 38
- command procedure
  - definition 83
- command tables 4
- communication with user 4
- compiler
  - linkage to 6
- conditional include of records 24
- CONT variable 47
- context editing
  - full-screen 5
- control elements 7
- control of user variables 4
- control service 19
- controlled group
  - definition 83
- controlled library
  - definition 83
- controlled project 39
  - definition 83
- converting dialogs from VM or VSE to MVS 48
- COPY
  - edit command 35
- copying data sets
  - under utilities 37
- copying lines
  - under edit 35
- customer responsibility 46
- customized edit/browse panel 61



- data
  - location 5
  - program-generated 4
  - user-entered 4
- data display panel 4, 29
- data elements 7
- data entry 70, 71
- data entry panel 3, 29, 70
  - definition 83
- data entry panel EMPLB, employee records
  - application 71
- data integrity 46
- data set
  - access 31
- data set characteristics
  - specifying 34
- data set list utility 61
- debug
  - interactive
    - COBOL 42
    - FORTRAN 42
    - VS FORTRAN 42
- declare statements
  - PL/I 68
- default
  - attribute character 15
- defaults
  - log and list
    - specifying 34
- definition of user variables 4
- delete variables
  - PL/I 69
- deleting
  - library members or files
    - under utilities 37
- deleting lines
  - under edit 35
- description of processing
  - employee records 64
- development environment 2
- dialog 1
  - definition 83
  - elements 7
  - example 63
  - migration of 46
  - organization 3, 9
  - test 6
  - variable 24

- dialog components
  - creation of 9
  - data entry panels EMPLA, EMPLB 63
  - employee records 63
  - messages 63
  - primary option menu EMPL 63
- dialog management
  - service 17
- dialog manager 1
- DIALOG TEST
  - PDF option 38
- dialog variable
  - definition 83
- display
  - formatting 14
  - PL/I
    - employee records 67
    - service 14, 17, 24
    - services 19
- display image
  - visible 7
- DISPLAY, display service 19
- displaying panels 38
- displaying table rows, table status, and table structure 38
- displaying variable values 38
- distribution table 39
  - definition 83
- Document Composition Facility 6, 42
- document preparation 36
- documentation preparation 6
- DSPRINT command processor 42, 45
- dual-case character set 45
- dynamically generated areas
  - displays containing 21

E

- EBCDIC
  - display character set 45
- edit
  - copying lines 35
  - deleting lines 35
  - inserting lines 35
  - macro language 6, 59
  - modifying data 35
  - recursive invocation 61
  - shifting lines 35
- edit recovery 36
- EDIT recovery service 6
- EDIT service 6

- editor
  - full-screen context 5
- EDREC service 6
- EMPL panel 63, 64, 66
- EMPLA panel 63
  - definition 70
  - display 70
- EMPLB panel 63
  - definition 71
  - display 72
- employee records 63
  - data entry 70, 71
  - dialog components 63
  - first data entry panel EMPLA 70
  - invocation 64
  - ISPSTART command 64
  - message definitions EMPX21 73
  - messages 73
  - organization 64
  - overview 63
  - PL/I
    - declare statements 68
    - delete variables 69
    - included segment 68
    - variable definitions 68
  - processing description 64
  - TBGET
    - PL/I 67
- environment
  - operating 2
- examples
  - dialog 10, 12
  - of a PDF session 75
  - of an ISPF dialog 63
  - sample panel
    - definition 16
    - display 17
  - sharing of variables 13
- excluded lines 36
- EXIT
  - PDF option 40

F

- file
  - definition 84
- file skeleton
  - definition 84
- file tailoring
  - service 17
- file tailoring service 23

FIND  
 edit command 35  
 foreground  
 member parts list 6, 60  
 PDF option 37  
 Foreground Environment Feature 42  
 foreground jobs 6  
 format  
 of panel 30  
 formatting  
 of displays 14  
 FORTRAN 42  
 foreground processor 37  
 Interactive Debug 42  
 Prompter 42  
 FORTRAN/VS 61  
 four-screens 4  
 FTCLOSE  
 file tailoring service 24  
 FTERASE  
 file tailoring service 24  
 FTINCL  
 file tailoring service 24  
 FTOPEN  
 file tailoring service 24  
 full-screen  
 editing 5  
 function 3  
 definition 84  
 function variable  
 definition 84  
 function, dialog  
 description of 8  
 functions  
 invoking 38  
 invoking from menus 4  
 Version 2 PDF 6, 59

G

GDDM 42  
 GDDM generated pictures or graphs  
 displays containing 20  
 general services 22  
 generation of sequential output 4  
 GETMSG service 20  
 glossary 83  
 Graphical Data Display Manager  
 (GDDM) 42  
 See also GDDM  
 group name

library identifier 31

H

hardware tabs 36  
 help  
 online 4  
 help panel  
 definition 84  
 hexadecimal display 36

I

include records conditionally 24  
 included segment  
 PL/I 68  
 initialization section  
 panel definition 14, 15  
 insert mask 35  
 inserting lines  
 under edit 35  
 insertion of new material 36  
 interactive debug  
 COBOL 42  
 FORTRAN 42  
 VS FORTRAN 42  
 interface  
 to language processors 6  
 invocation  
 employee records 64  
 of ISPF 26  
 of PDF 33  
 invoking  
 ISPF service 38  
 invoking functions 38  
 ISPF  
 definition 84  
 invocation 26  
 library 31  
 library structure 31  
 PDF  
 definition 84  
 ISPF library  
 access services 6, 59  
 management facility 6, 59  
 ISPF log  
 browsing 38  
 ISPF PARMS

- PDF option 33
- ISPF service
  - invoking 38
- ISPPARM data set 47
- ISPPROF data set 47
- ISPSTART
  - command 26
- ISPSTART command
  - employee records 64
- iteratively process records 24

K

- Katakana
  - display character set 45

L

- language processor interface 6
- library
  - definition 84
  - hierarchy 31
  - identifier 31
  - ISPF 31
  - maintenance 5
  - multilevel support 5
- library access services 6, 59
  - list 59
- library administrator 39
  - definition 84
- library identifier
  - group name 31
  - project name 31
  - type 31
- library management facility 6, 59
- library management utilities
  - PDF option 39
- library structure
  - ISPF 31
- link edit 37
- linkage editor 42
  - linkage to 6
- LIST
  - selection 29
  - verification keyword 15
- listing
  - data sets
    - under utilities 37

- loader
  - linkage to 6
- LOCATE
  - edit command 35
- location
  - of data 5
- LOG
  - PL/I
    - employee records 67
  - log service 19
  - logical tabs 36

M

- machine requirements 43
- macro language
  - edit 6, 59
- maintenance
  - of libraries 5
  - of user tables 4
  - panel and message definitions 14
- management facility
  - library 6, 59
- managing breakpoints 39
- member parts list 6, 60
  - foreground processor 37
- menu 3
  - See also selection panel
    - definition 84
    - display 4
- message
  - definition 84
  - description of 9
  - display formatting 14
- message definitions
  - maintenance 14
- message handler
  - TCAM 46
- message saving
  - GETMSG service 20
  - SETMSG display service 20
- messages
  - employee records 73
- migration of dialogs 46
- model
  - definition 84
- model class
  - definition 84
- model facility 4, 9, 36
- model section
  - panel definition 14

- models 5
- modifying data
  - under edit 35
- modifying table rows 38
- MOVE
  - edit command 35
- moving data sets
  - under utilities 37
- multilevel programming library 5
- multiple
  - additions and changes 5
- multiple screen display 4

N

- names
  - reserved 25
- naming conventions
  - conventional data set 33
- new edit commands 61
- NEWAPPL keyword 47
- null characters 36

O

- online
  - help 4
  - tutorial 4, 6
- operating environment 2
- optimizing
  - PL/I
    - compiler 42
- organization
  - employee records 64
  - of a dialog 3, 9
- overview
  - employee records 63

P

- packed data 60
- pageable link pack area 43
- panel 7
  - data display 4, 29
  - data entry 3, 29, 70

- definition 83
- definition 84
- EMPL 63, 64
  - definition 66
  - display 66
- EMPLA 63
  - definition 70
  - display 70
- EMPLB 63
  - definition 71
  - display 72
- format 30
- selection 3, 29
- table display 29
- types of 29
- panel definition
  - attribute section 14
  - body 14
  - initialization section 14, 15
  - model section 14
  - processing section 14, 15
  - reinitialization section 14
- panel definitions
  - maintenance 14
- panel display
  - description of 8
- panel EMPL
  - definition 66
  - display 66
- panels
  - displaying 38
  - testing 38
- partition
  - display screen 4
- partitioned data set
  - access services 6, 59
- parts of a dialog
  - dialog components 8
- PASCAL 42
  - foreground processor 37
- PCF
  - See Programming Control Facility
- PDF
  - definition 84
  - functions
    - Version 2 6, 59
  - invocation 33
  - options 33
- percent sign
  - attribute character 15
- PF key assignments
  - specifying 34
- PGF 42
- PICT

- verification keyword 15
- picture
  - display format 14
- PL/I
  - Checkout Compiler 42
  - declare statements 68
  - delete variables 69
  - DISPLAY
    - employee records 67
  - foreground processor 37
  - included segment
    - define variable 68
  - included segment/delete variables 69
  - LOG
    - employee records 67
  - Optimizing Compiler 42
  - TBADD
    - employee records 67
  - TBCLOSE
    - employee records 67
  - TBCREATE
    - employee records 67
  - TBDELETE
    - employee records 67
  - TBGET
    - employee records 67
  - TBOPEN
    - employee records 67
  - TBPUT
    - employee records 67
  - variable definitions 68
  - VDEFINE
    - employee records 69
  - VDELETE
    - employee records 69
- planning 41
- PLPA
  - See pageable link pack area
- plus sign
  - attribute character 15
- position of command line
  - specifying 34
- position of long message line
  - specifying 34
- Presentation Graphics Feature (PGF) 42
- primary option menu 33, 63
- primary option menu panel 66
- primary option menu panel EMPL 66
- printing
  - library members or files
    - under utilities 37
- processing section
  - panel definition 14, 15
- processor

- language
  - interface to 6
- profile data 47
- profile variable 13
  - definition 85
- program 3
- program development 5
- program elements 7
- program-generated data 4
- Programming Control Facility 42
- programming requirements 41
- project name
  - library identifier 31
- promote 39
  - definition 85
- promoting data sets
  - under utilities 37
- promotion hierarchy
  - definition 85
- prompter
  - assembler 42
  - COBOL 42
  - FORTRAN 42
- publications
  - supporting 48

Q

- qualifier
  - See library identifier

R

- rearrangement of text 36
- recursive edit/browse invocation 61
- region size
  - minimum 44
- reinitialization section
  - panel definition 14
- renaming
  - library members or files
    - under utilities 37
- requirements
  - machine 43
  - programming 41
  - terminal 45
- reserved names 25
- responsibility

customer 46

S

SAVE  
  edit command 35  
screen  
  split 7  
screen size 30  
SCRIPT/VS  
  See also Document Composition  
  Facility  
  foreground processor 37  
scrolling 5  
  4-way  
    browse 35  
security 46  
SELECT  
  service 11  
select service 17, 19  
SELECT, service 19  
selection list 29  
selection panel 3, 29  
  definition 85  
sequential data set  
  access services 6, 59  
sequential output generation 4  
service  
  display 19  
  file tailoring 23  
  general 22  
  variable 24  
services  
  control 19  
  dialog management 17  
  display 17  
  file tailoring 17  
  log 19  
  select 17, 19  
  table 17, 21  
  variable 17  
SETMSG  
  display service 20  
setting variable values 38  
shared variable 13  
  definition 85  
shifting lines  
  under edit 35  
size  
  EPLPA (MVS) 44  
  mimumum region 44

PLPA (MVS) 44  
  recommended 44  
skeleton  
  description of 9  
  file tailoring services 23  
  imbedding other 24  
software tabs 36  
SORT  
  edit command 35  
space requirements 44  
specifying  
  data set characteristics 34  
  log and list defaults 34  
  PF key assignments 34  
  position of command line 34  
  position of long message line 34  
  terminal characteristics 34  
SPLIT  
  command 7  
split-screen 4, 7  
substitution  
  of variables  
    file tailoring service 24

T

table  
  definition 85  
  description 9  
  permanent 22  
  rows  
    displaying and modifying 38  
  services 21  
  temporary 22  
  when created or updated 9  
table display panel 29  
table services 17  
table status and structure  
  displaying 38  
tabs  
  hardware 36  
  logical 36  
  software 36  
TBADD  
  employee records 65  
  PL/I  
    employee records 67  
  table service 22  
TBBOTTOM  
  table service 23  
TBCLOSE

PL/I  
     employee records 67  
     table service 22  
 TBCREATE  
     employee records 65  
     PL/I  
         employee records 67  
         table service 22  
 TBDELETE  
     employee records 65  
     PL/I  
         employee records 67  
         table service 22  
 TBDISPL  
     display service 19  
 TBEND  
     table service 22  
 TBERASE  
     table service 22  
 TBEXIST  
     table service 23  
 TBGET  
     PL/I  
         employee records 67  
         table service 23  
 TBMOD  
     table service 23  
 TBOPEN  
     PL/I  
         employee records 67  
         table service 22  
 TBPUT  
     employee records 65  
     PL/I  
         employee records 67  
         table service 23  
 TBQUERY  
     table service 22  
 TBSARG  
     table service 23  
 TBSAVE  
     table service 22  
 TBSCAN  
     table service 23  
 TBSKIP  
     table service 23  
 TBSORT  
     table service 22  
 TBSTATS  
     table service 22  
 TBTOP  
     table service 23  
 TBVCLEAR

    table service 23  
 TCAM 41, 42  
     message handler 46  
 terminal characteristics  
     specifying 34  
 terminal requirements 45  
 test  
     dialog 6  
 test facility 4, 9  
     See also dialog test  
 testing panels 38  
 text  
     display character set 45  
 trace  
     definition 85  
 tracing  
     applications and function calls to  
         ISPF services 39  
         dialog test 38  
         variable usage 39  
 tracking  
     program segments 5  
 TSO  
     CLIST 4  
 tutorial  
     online 4, 6  
     PDF option 40  
 tutorial panel  
     definition 84  
 two-dimensional array  
     table 21  
 two-screen 7  
 two-screens 4  
 type  
     library identifier 31

U

underscore  
     attribute character 15, 25  
 UP variable 47  
 user  
     communication with 4  
 user tables  
     building and maintaining 4  
 user-entered data 4  
 utilities 5  
     library management  
         PDF option 39  
         PDF option 37



V

variable  
  application profile pool 13  
  association with functions 11  
  communication between functions 13  
  definition 24, 85  
  dialog 24  
  explicitly defined 11  
  function  
    defined 13  
    implicit 13  
  implicitly defined 11, 13  
  profile 13  
  services 11, 17, 24  
  shared 13  
  substitutable 15  
  use of 11, 24  
  user  
    definition and control 4  
variable definitions  
  PL/I 68  
variable usage  
  tracing 39  
variable values  
  displaying and setting 38  
VCOPY  
  service 11  
  variable service 25  
VDEFINE  
  PL/I  
    employee records 69  
  service 11  
  variable service 25

VDELETE  
  PL/I  
    employee records 69  
    variable service 25  
VER statement 15  
verification  
  of data 15  
Version 2  
  ISPF functions 4  
  PDF functions 6, 59  
VGET  
  service 11, 13  
  variable service 25  
VGET/VPUT 47  
VPUT  
  service 13  
  variable service 25  
VREPLACE  
  variable service 25  
VRESET  
  variable service 25  
VS FORTRAN  
  Interactive Debug 42  
VTAM 41, 42

Z

Z  
  reserved names 25  
ZCMD variable 47  
ZCONT variable 47  
ZSEL variable 47  
ZUP variable 47

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 T46  
P. O. Box 60000  
Cary, North Carolina 27511

Fold and tape

Please Do Not Staple

Fold and tape

ISPF and ISPF/PDF for MVS/Extended Architecture General Information Printed in U.S.A. GC34-4041-0



Publication Number  
GC34-4041-0

File Number  
S370/4300-39

Program Numbers  
5665-317  
5665-319

Printed in  
U.S.A.

**IBM**

GC34-4041-0

