

Form No. GC34-2039-0
File No. S370/4300-39

**System Productivity
Facility for MVS**

Program Product

General Information

Program Number 5668-009

PLEASE RETURN TO:
PANSOPHIC SYSTEM INC.
DATACENTER LIBRARY

IBM

FIRST EDITION (August 1980)

This edition applies to the Program Product System Productivity Facility for MVS (5668-009), and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services which 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 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, Systems Publications, Dept. Z59, P.O. Box 390, Poughkeepsie, NY U.S.A. 12602. 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

The System Productivity Facility (SPF) is a program development tool designed to take advantage of the characteristics of IBM 3270 display terminals, and to increase programmer productivity in an interactive environment.

The System Productivity Facility replaces the previous Structured Programming Facility program products (SPF/TSO and SPF/CMS). It includes significant new functions that simplify the development of interactive applications.

New services are provided to define and control symbolic variables, display predefined screen images and messages, build and maintain permanent tables of user information, generate output files for job submission or other processing, interface to edit and browse, and log hardcopy output.

This manual provides an overview and functional description of SPF under the MVS Time Sharing Option. It is intended to provide planning information for installation and use of the product.

A separate manual, GC34-2046, provides general information about SPF under the VM/370 Conversational Monitor System.

This manual is divided into the following chapters:

1. Introduction - A general overview of SPF, including its structure and function.
2. Dialog Manager - A description of the control facilities and services that support the operation of interactive applications.
3. Program Development Facility - A description of the interactive facilities that support program development, including browse, edit, utilities, foreground and background processing, and online tutorial.
4. Planning - A description of system requirements, customer responsibilities, and related documentation.
5. Program Summary - A summary of licensing and testing provisions, and planned availability.

The manual also includes an appendix describing the differences in SPF facilities in the MVS and VM environments.

CONTENTS

Chapter 1. Introduction	1
Chapter 2. Dialog Manager	3
Control Facilities	4
Menu Traversal	4
Online Tutorial	6
Screen Management	7
Program Access and Function Keys	7
Dialog Services	8
Variable Services	9
Display Service	11
Table Services	16
File Tailoring Services	18
Other Services	20
Chapter 3. Program Development Facility	21
General Description	22
Display Format	22
SPF Libraries	25
Parameter Specification	27
Program Access and Function Keys	30
Scrolling	32
Overtyping of Scroll and Multiple Choice Parameters	33
Split Screen	33
Help Information	34
Library Activity Statistics	35
List and Log Files	35
Invocation and Termination	36
Member Lists	38
SPF Parms (Option 0)	39
Browse (Option 1)	41
Edit (Option 2)	43
Basic Edit Operations	45
Edit Modes and Profiles	46
Special Features	46
Edit Termination	47
Utilities (Option 3)	48
Library Utility	49
Data Set Utility	50
Move/Copy Utility	50
Catalog Management Utility	51
Reset SPF Statistics Utility	51
Hardcopy Utility	52
List VTOC Utility	52
Outlist Utility	52
SCRIPT/VS Utility	53
Foreground (Option 4)	54
Background (Option 5)	57
Command (Option 6)	59
Support (Option 7)	60
Tutorial (Option T)	61
Chapter 4. Planning	63
Programming Requirements	63
Machine Requirements	64
Terminal Requirements	64
Customer Responsibilities	65
Conversion of Menus and Messages	65
Related Documents	66
Chapter 5. System Productivity Facility: Summary	67
Appendix A. SPF-MVS and SPF-VM Differences	69
Dialog Manager	69
Program Development Facility	70
Index	72

LIST OF ILLUSTRATIONS

Figure 1. SPF Organization	1
Figure 2. Flow of Control	5
Figure 3. Sets of Variables	10
Figure 4. Sample Panel Definition	13
Figure 5. Sample Panel - When Displayed	13
Figure 6. Sample Panel with Assignment, IF, and VER Statements	14
Figure 7. Sample Member in Message Library	15
Figure 8. Sample Table	16
Figure 9. Sample Skeleton File	19
Figure 10. Primary Option Menu	23
Figure 11. Browse - Entry Panel	23
Figure 12. Browse - Member Selection List	24
Figure 13. Browse - Data Display	24
Figure 14. SPF Library Organization	25
Figure 15. Hierarchy of SPF Libraries	26
Figure 16. Default Program Key Arrangement	30
Figure 17. Split Screen Example	34
Figure 18. Primary Option Menu	36
Figure 19. Member List Display	38
Figure 20. SPF Parameter Menu	39
Figure 21. PF Key Assignment Panel (For 24-key Terminals)	40
Figure 22. Browse - Entry Panel	42
Figure 23. Browse - Data Display	42
Figure 24. Edit - Entry Panel	44
Figure 25. Edit - Data Display	44
Figure 26. Utility Selection Menu	48
Figure 27. Library Utility Panel	49
Figure 28. Foreground Selection Menu	54
Figure 29. Foreground - Assembler Example	55
Figure 30. Background Selection Menu	57
Figure 31. Support Selection Menu	60
Figure 32. Tutorial - First Two Pages	62

CHAPTER 1. INTRODUCTION

The System Productivity Facility (SPF) is a new program product which replaces both of the previous Structured Programming Facility products (SPF/TSO and SPF/CMS). The new product supports two environments:

- MVS Time Sharing Option (SPF-MVS)
- VM/370 Conversational Monitor System (SPF-VM)

The new name, System Productivity Facility, reflects the addition of significant new functions beyond the support for structured programming. The new functions simplify the development of interactive applications that run under control of SPF and use new SPF services to:

- Define and control symbolic variables
- Display predefined screen images and messages
- Build and maintain permanent tables of user information
- Generate output files for job submission or other processing
- Interface to edit and browse, and log hardcopy output
- Control operational modes.

Conceptually, SPF consists of two major components: the dialog manager and the program development facility (see Figure 1). The dialog manager provides control and services for running interactive applications. One such application is the program development facility, which includes the previous SPF functions.

The dialog manager allows totally new applications to be developed, each with its own primary option menu. Applications may be independent, entered via separate command procedures, or linked via menu options which transfer from one application to another.

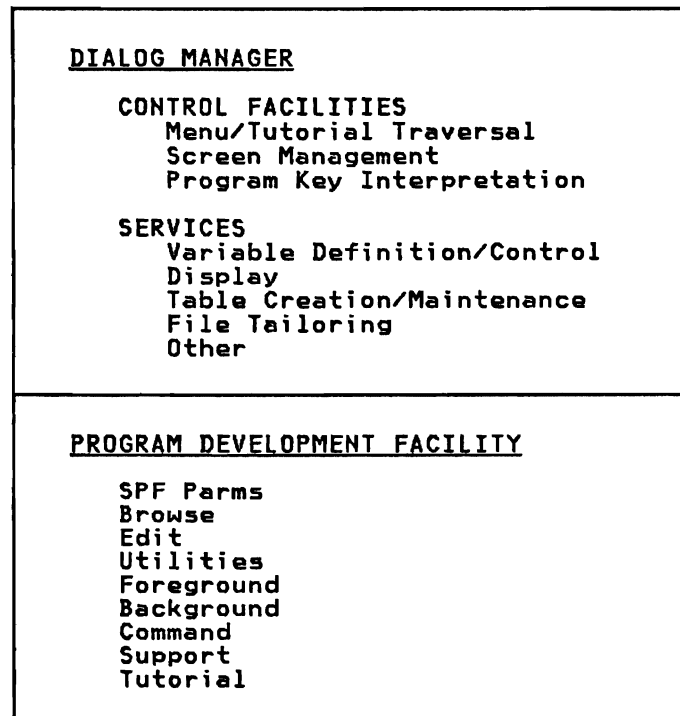


Figure 1. SPF Organization

The dialog manager includes a new panel¹ display service with enhanced function and improved usability over the previous SPF display capability.

Installations which have previously extended or custom tailored SPF may need to convert old format selection menus, tutorials, and messages to the new panel and message formats. Conversion utilities have been provided for this purpose in the program development facility (new option 7 on the primary option menu).

Advantages of the new panel and message formats include:

- Communication of user inputs (from panels) and substitution of parameters (in messages) via symbolic variables rather than positional calling sequence parameters.
- Ability to verify user inputs in panel definitions.
- Ability to test and set variables in panel definitions.
- Improved control over panel field attributes, including specification of padding characters and justification.
- Replacement of LIST/RETURN keywords with TRANS keyword, in which the source and translated values are paired.
- Simplification of selection menu and tutorial panel formats.

The program development facility provides interactive facilities for program development and maintenance, including:

- SPFParms - To specify terminal characteristics, including program function (PF) keys, and other SPF parameters.
- Browse - To display source data or output listings.
- Edit - To create or change source data, including program code, test data, or documentation.
- Utilities - To perform a variety of library and data set specification and maintenance functions.
- Foreground - To execute language processors (compilers, assembler, and linkage editor) in the foreground.
- Background - To execute language processors in the background.
- Command - To execute a TSO command or CLIST under SPF.
- Support - To test a panel or dialog function, or convert old format menus/messages to new format.
- Tutorial - To provide online instruction and reference in the operation of the program development facility.

Typically, the end user of the program development facility is an application or systems programmer. The program development facility may also be useful for administrative and clerical personnel who have some knowledge of the operating system environment.

¹ Previously, all SPF screen images were called menus. The terminology has been changed to more closely reflect general usage. The term menu is now used to mean a display from which the user may select options. The term panel is used to mean any predefined display image, of which one type is a menu.

A "dialog" is any application designed to be run under the SPF dialog manager. Each dialog is composed of various program and data elements, which allow an orderly interaction between the computer and the end user of the application. The types of elements that make up a dialog are:

- **Panels.** A panel is a predefined display image. It may be a selection menu, a data entry display, or an information display (e.g., help/tutorial panel). Most panels prompt the user for information. The user response may identify which path is to be taken through the dialog, or it may be interpreted as data.
- **Messages.** A message is 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 completed, or report an error in the user's input. Messages may be directed to the user's terminal, to a hardcopy log, or both.
- **Functions.** A function is a program that controls the sequence in which panels and messages are displayed, oversees the handling of data and files, and performs processing functions requested by the user. A function may be a compiled (and link edited) program or a CLIST.
- **Tables.** A table is a two-dimensional array used to maintain 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 between different applications. The type and amount of data stored in a table depends upon the nature of the application.
- **File Skeletons.** A file skeleton is a generalized representation of sequential data which 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 originating from functions, panels, or tables. The output file may be used to drive other processes. File skeletons are frequently used to produce job files for batch execution.

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

Panels, messages, and file skeletons are constructed by the dialog developer and stored in libraries (partitioned data sets) prior to execution of the dialog. They are created by editing directly into the panel, message, or skeleton libraries; no compile or preprocessing step is required.

Functions are coded (and, for program modules, compiled and link edited) by the dialog developer and stored in load module or command libraries. They are invoked from menus or from other functions during dialog execution.

Tables are generated or updated dynamically during dialog execution. The organization of each table is preplanned by the dialog developer, and specified to SPF by the functions that use SPF table services.

CONTROL FACILITIES

The SPF dialog manager provides control facilities to:

- Traverse a hierarchy of selection menus and invoke the appropriate dialog functions.
- Transfer in and out of the tutorial, and control the sequence of tutorial pages based on user inputs.
- Manage the physical display image in single screen or split screen mode.
- Interpret program function (PF) key usage for system defined functions.

The overall flow of control is shown in Figure 2. The dialog manager is invoked via the ISPF command. (An SPF command may be established as an alias of ISPF). The command may be entered by a user at the terminal or invoked from a CLIST. Command parameters may be used to specify:

- The first selection menu to be displayed, or
- The first dialog function to receive control (prior to the display of a menu).

If a dialog function is specified, it may perform application-dependent initialization, and then invoke the SELECT service to display the first menu. If neither a selection menu nor a dialog function is specified, the SPF program development facility is invoked.

Before the dialog can begin, the SPF environment must be initialized. This is done by the controller. The controller then calls the SELECT service to display the first menu or invoke the first dialog function, as specified via the ISPF command.

MENU TRAVERSAL

Selection menus are normally organized in a hierarchical structure. Each menu contains sufficient information to determine the next action to be taken for any option entered by the user. The next action may be:

- Display a lower level selection menu, or
- Invoke a dialog function.

Selection menus are displayed by the SELECT service. The SELECT service will display the first menu and continue to display successively lower levels of menus, until a dialog function is specified as the next action. The SELECT service will then invoke the function.

A dialog function may display additional panels (via the DISPLAY service), and invoke other SPF services to continue the dialog. When the function completes execution, it returns to the SELECT service. The selection menu from which the function was invoked is then redisplayed.

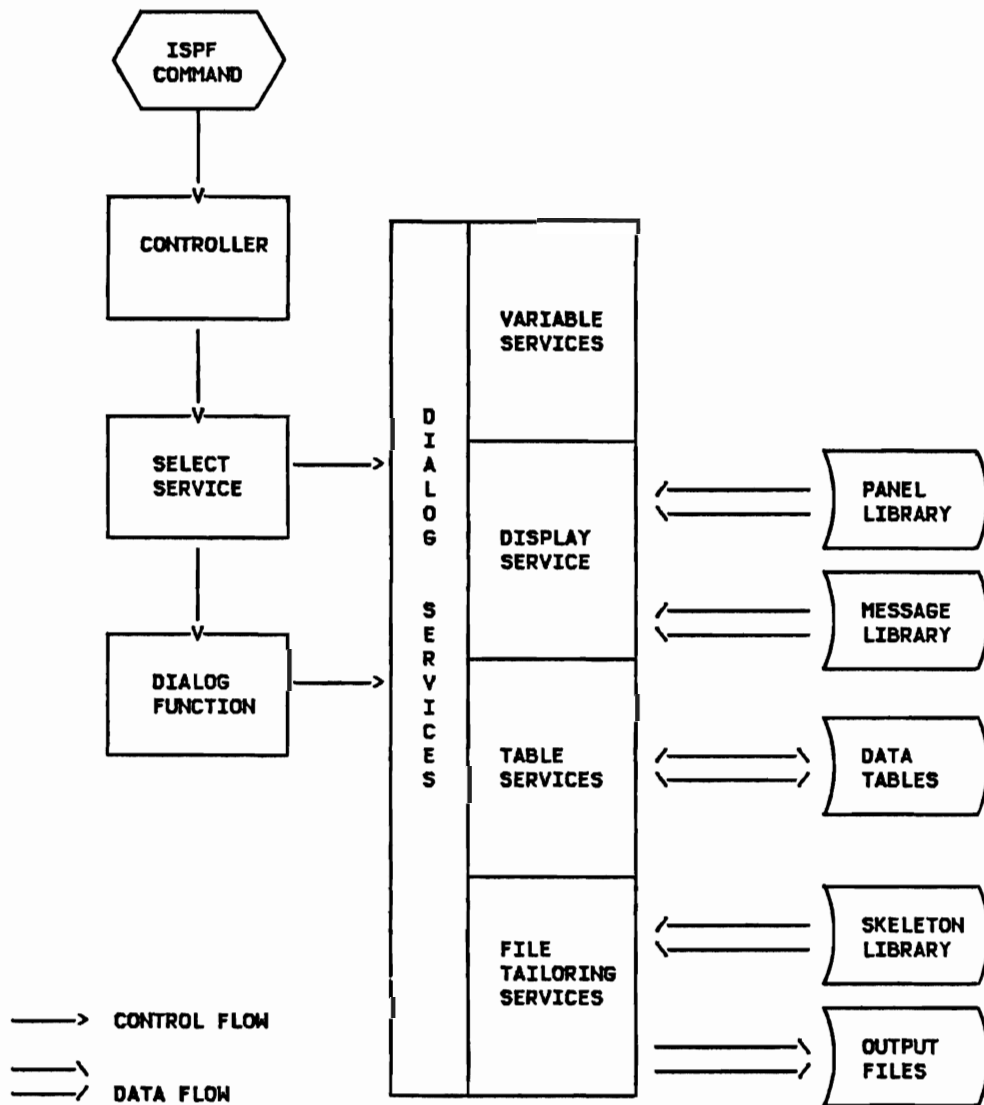


Figure 2. Flow of Control

The foregoing describes a dialog organization in which selection menus are displayed first, and dialog functions are invoked from the menu options. The functions are the "end points" of each leg of the menu hierarchy. Other types of dialog sequences may be constructed by means of the following features:

- Any dialog function may use the SELECT service to display additional selection menus. A function, for example, might display data entry panels to gather user information and then cause a selection menu to be displayed from which lower level functions may be invoked.
- Any dialog function may use the SELECT service to pass control directly to another dialog function without displaying a menu. This provides a simple way to pass control from a program module to a CLIST, or vice-versa.
- Parameters may be passed to any dialog function from the selection menu or function which invokes it. These parameters may be used, for example, to pass the name of a panel to be displayed, a table to be updated, or a file skeleton to be used by the function.
- Whenever a selection menu is specified from the SPF command or via a parameter passed to the SELECT service, an initial option may also be specified. This causes the menu to be processed as if the user had entered the specified option, without actually displaying the menu.

ONLINE TUTORIAL

A tutorial is a set of panels that provide online information to the end user. As with other panels, the tutorial is constructed by the dialog developer and stored in the panel library prior to execution of the dialog.

The program that displays tutorial pages is part of the dialog manager. It may be entered in either of two ways:

- As a selectable option from a menu, or
- Indirectly from any non-tutorial panel when the user presses the Help PF key.

Transfer in and out of the tutorial via the Help key is transparent to the dialog functions.

Tutorial panels are arranged in a hierarchy. When the tutorial is entered from a menu, the first panel to be displayed is normally the top of the hierarchy. When the tutorial is entered via the Help PF key, the first panel to be displayed is some appropriate panel within the hierarchy, depending upon what the user was doing when the Help key was pressed.

When viewing the tutorial, the user may select topics by number (or other appropriate selection code), or simply press the ENTER key to view the next topic. On any panel, the user may also enter the following commands:

BACK or B - to back up to the previously viewed panel
SKIP or S - to skip to the next topic
UP or U - to display a higher level list of topics
TOP or T - to display the table of contents
INDEX or I - to display the tutorial index.

When the user has finished viewing the tutorial, the panel from which the tutorial was entered is redisplayed.

SCREEN MANAGEMENT

At any time during a dialog, the end user may partition the display screen into two "logical" screens. The two logical screens are treated as though they were independent terminals. The dialog manager provides control for mapping the two logical screens onto the physical screen.

In split screen mode, one or the other of the logical screens is considered active at any point in time. The location of the cursor is used to identify which of the two screens is active.

Split screen mode is entered by means of the Split PF key, which may also be used to reposition the split line. Split screen mode is terminated by ending the application on either logical screen. The remaining logical screen is then expanded to its full size.

Use of split screen mode and positioning of the split line is under control of the end user, and totally transparent to the dialog function. Panels that are displayed by the DISPLAY service always pertain to a logical screen.

PROGRAM ACCESS AND FUNCTION KEYS

The dialog manager supports display terminals that have two program access (PA) keys, and 12 or 24 program function (PF) keys. Some keys have system-defined meanings; these are handled by the dialog manager, and are transparent to the dialog function except for the End key.

Other keys may be equated to application-defined commands; these are passed through to the dialog function, as if the user had typed the command on the screen and pressed the ENTER key.

The two PA keys have system-defined meanings. They may not be redefined by the user.

ATTENTION (PA1) This key is logically disabled whenever the keyboard is unlocked, except during execution of a TSO command.

RESHOW (PA2) Redisplays the contents of the screen.

The system-defined PF key operations are described below. The default key assignments are shown in parentheses. For 24-key terminals, PF keys 1-12 have the same defaults as keys 13-24.

HELP (PF1/13) Displays additional information about a message or causes a transition into the tutorial.

SPLIT (PF2/14) Causes split screen mode to be entered, or changes the location of the split line.

END (PF3/15) Terminates the current operation and returns to the previous menu. If the primary option menu is displayed, this key terminates the application.

RETURN (PF4/16) Causes an immediate return to the primary option menu. (Logically equivalent to repeated use of the End key.) May also be used to jump directly from one function to another, without displaying the primary option menu.

UP (PF7/19) Causes a scroll up.

DOWN (PF8/20) Causes a scroll down.

SWAP (PF9/21) Moves the cursor to wherever it was previously positioned on the other logical screen.

LEFT (PF10/22) Causes a scroll left.

RIGHT (PF11/23) Causes a scroll right.

CURSOR (PF12/24) Moves the cursor to the first input field on line 2 (normally, the option selection or command input field). Pressing this PF key again causes the cursor to be moved to the second input field on line 2, if any (normally the scroll field).

PRINT (none) Causes a "snapshot" of the screen image to be recorded in the SPF list file.

PRINT-HI (none) Same as PRINT except that high intensity characters on the screen are printed with overstrikes to simulate the dual intensity display.

NOP (none) Causes the PF key to be functionless.

The scroll keys are used if the dialog function invokes the table display service (TBDISPL) or the interfaces to edit and browse. During execution of the tutorial, the four scroll PF keys are interpreted as follows:

UP - same as UP command
 DOWN - same as SKIP command
 LEFT - same as BACK command
 RIGHT - same as ENTER key (display next page).

The PRINT, PRINT-HI, and NOP functions have no default PF key assignments.

The end user may rearrange the system-defined keys, and may redefine system keys to application-defined commands. The only system key function which is required is the End key.

DIALOG SERVICES

The SPF dialog manager provides a comprehensive set of services that allow an application to:

- Define and control symbolic variables
- Display predefined screen images and messages
- Build and maintain permanent tables of user information
- Generate output files for job submission or other processing
- Interface to edit and browse, and log hardcopy output
- Control operational modes.

SPF services may be invoked from a CLIST by means of the ISPEXEC command. Example:

```
ISPEXEC DISPLAY PANEL(XYZ)
```

This command invokes the DISPLAY service to display a panel named XYZ.

SPF services may be invoked from programs by calling a service interface routine, named ISPLINK. Example:

```
CALL ISPLINK ('DISPLAY', XYZ);
```

ISPLINK is a small object module which is distributed with SPF. It should be link edited with the program functions that call it. It may be called from programs coded in any language that uses standard OS register conventions for call interfaces.

VARIABLE SERVICES

Variable services allow data elements to be identified and used as "dialog variables." A dialog variable is a character string that may vary in length from zero to 32K bytes. It is referenced symbolically, by name. The name may be from one to eight alphanumeric characters in length, and must begin with an alphabetic character.

Dialog variables serve as the main communication vehicle between dialog functions and SPF services. They may also be used to communicate between a function and another function.

Referencing of Variables

Dialog variables are either referenced implicitly or defined to SPF, depending upon the type of dialog function.

For CLIST functions, the variables are always referenced implicitly. All CLIST variables are automatically treated as dialog variables; no special action is required to define them to SPF. CLIST variables are created dynamically either by execution of the CLIST or by the SPF services that the CLIST uses. Example:

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

Variable AAA is created by the CLIST, simply by setting it to a value. The CLIST then invokes the DISPLAY service to display panel XYZ. If panel XYZ references variable AAA, its value may be displayed or changed by the user, depending on how the panel is defined.

The same panel may allow the user to enter a value for another variable (BBB). If the CLIST has not already created a variable BBB, the variable will be created automatically. Its value may then be referenced in a subsequent CLIST statement.

For program functions (compiled modules), dialog variables that are to be referenced by the function must be explicitly defined to SPF. The function calls the VDEFINE service to identify the name, address, format, and length of one or more variables within the program to be used as dialog variables. Example:

```
DECLARE AAA CHAR(8);
CALL ISPLINK ('VDEFINE', '(AAA)', AAA, 'CHAR', 8);
CALL ISPLINK ('DISPLAY', XYZ);
```

Variable AAA is declared as an internal program variable (character string, length 8). The program calls the VDEFINE service to define it as a dialog variable. The program then calls the DISPLAY service to display panel XYZ. If panel XYZ references variable AAA, its value may be displayed or changed by the user, depending on how the panel is defined.

If panel XYZ allows the user to enter a value for another variable (BBB) and the program has not defined a dialog variable of that name, storage for the variable will be allocated automatically. BBB is then considered an implicit dialog variable associated with this function. The program has no means of accessing BBB, since it has not been explicitly defined. But if the program invokes another SPF service, the service will be able to access and/or modify the variable (see "Scope of Variables" which follows).

The VDELETE service performs the opposite function of VDEFINE. It may be called to specify the names of one or more program variables which are no longer to be treated as dialog variables.

Scope of Variables

The scope of a dialog variable may be limited to an individual function or shared between functions.

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

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

A shared variable pool is available to allow communication of variables between functions. A function may copy one or more of its variables into the shared variable pool by means of the VPUT service. Another function may then obtain a copy of the variables by means of the VGET service.

Figure 3 shows the logical relationships between different sets of variables. An application with three functions is shown. Function X has both defined and implicit variables. Function Y has only defined variables, and function Z has only implicit variables.

When variables are accessed by SPF services, the search starts with the dialog variables for the current function -- first the defined variables (if any), then the implicit variables (if any). If a variable is not found, the search continues with the shared variable pool. If the variable is still not found, its current value is assumed to be null.

When a variable is created or updated by an SPF service, it is stored in the current function's defined area if a variable of that name has been explicitly defined. Otherwise, it is stored in the current function's implicit area.

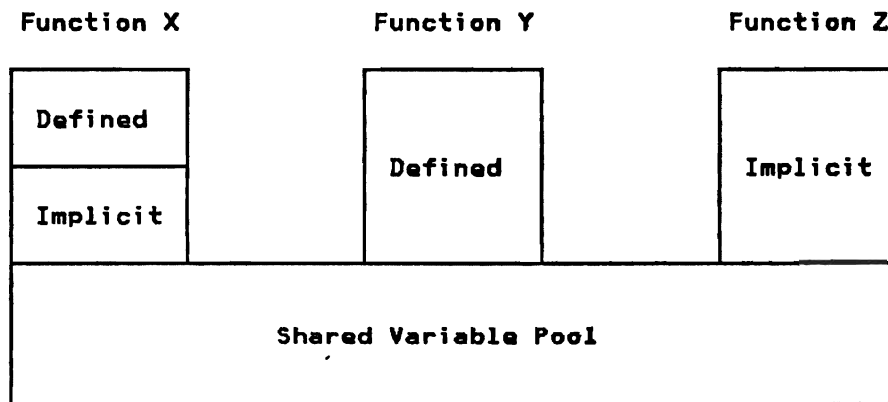


Figure 3. Sets of Variables

System Variables

Certain variable names are reserved for use by the system. They all begin with the letter "Z". Dialog developers should avoid names which begin with "Z" when choosing dialog variable names.

Some system variables cannot be modified. They provide the dialog with information about the environment, such as user id, current date and time, and terminal characteristics. These variables reside in the shared variable pool, and may be obtained via the VGET service.

Other system variables are used for communication between the dialog and certain control facilities within the dialog manager. These variables provide information about error conditions and allow designation of key tutorial pages.

DISPLAY SERVICE

The DISPLAY service reads a panel definition from the panel library, initializes variable panel fields from the corresponding dialog variables, and displays the panel on the screen. A message may optionally be displayed with the panel. After the user presses ENTER or a PF key, user inputs are stored into corresponding dialog variables, and the display service returns to the calling function.

A separate service is provided to display tabular information in scrollable format. See "Table Display Service."

Panel Definitions

A panel definition consists of up to four sections, of which only the body is required:

1. 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. Default attribute characters are provided, which may be overridden.
2. Body (required) - defines the format of the panel as seen by the user, and defines the name of each variable field on the panel.
3. Initialization section (optional) - specifies the initial processing that is to occur prior to displaying the panel. Typically used to define how variables are to be initialized.
4. Processing section (optional) - specifies processing that is to occur after the panel has been displayed. Typically used to define how variables are to be verified and/or translated.

Panel Fields

The special characters defined in the attribute section (or the default attribute characters) are used in the panel body to indicate the start of each field, which is also the end of the preceding field.

The default attribute characters are:

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

For text fields, the information following the attribute character is the text to be displayed. Text fields may contain substitutable variables, consisting of a dialog variable name preceded by an ampersand (&). The name and ampersand are replaced with the value of the variable prior to displaying the panel.

For input fields, a dialog variable name immediately follows the attribute character (with no intervening ampersand). The name is replaced with the value of the variable prior to displaying the panel, and any information entered by the user is stored in the variable after the panel has been displayed.

Inclusion of an attribute section, which precedes the panel body, allows the default attribute characters to be changed and/or additional attribute characters to be defined.

In defining new attribute characters, the following characteristics of the field may be specified:

- Type - text, input, or output
- Intensity - high, low, or non-display
- Caps - automatic translation to uppercase (on or off)
- Justification - left, right, or as-is
- Pad - nulls, or any character (including blank)

Panel Definition Examples

A sample panel definition is shown in Figure 4. It consists of a panel body followed by an ")END" control statement. It has no attribute, initialization, or processing sections. It uses the default attribute characters.

This is a data entry panel with ten input fields (TYPECHG, LNAME, etc.), indicated with underscores. It also has a substitutable variable (EMPSER) within a text field (on line 2). The first two lines of the panel and the arrows preceding the input fields are all highlighted, indicated with percent signs. The other text fields are low intensity, indicated with plus signs.

Before the panel is displayed, all variables in the panel body will be automatically initialized from the corresponding dialog variables (TYPECHG, LNAME, etc., and EMPSER). After the panel has been displayed, the input fields will be automatically stored into the corresponding dialog variables.

Figure 5 shows the panel as it will appear when displayed, assuming that the current value of EMPSER is "123456", and that the other variables are initially null.

```

%----- EMPLOYEE RECORDS -----
%EMPLOYEE SERIAL: &EMPSER

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

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

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

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

)END

```

Figure 4. Sample Panel Definition

```

----- EMPLOYEE RECORDS -----
EMPLOYEE SERIAL: 123456

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

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

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

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

```

Figure 5. Sample Panel - When Displayed

Figure 6 shows a sample panel definition with an initialization section, starting with an ")INIT" statement, and a processing section, starting with a ")PROC" statement.

The initialization section sets the control variable .HELP to the name of the corresponding help (tutorial) panel. This is the name of the panel which will be displayed if the user presses the Help PF key. Then the current value of variable PHA is tested. If it is blank (or null), it is initialized to a default value (301).

The processing section contains VER statements to verify that information entered by the user meets the following criteria:

- The three name variables (LNAME, FNAME, and I) contain all alphabetic characters.
- The area code (PHA) contains all numeric characters.
- The local number (PHNUM) contains three numeric characters, followed by a hyphen, followed by four numeric characters.

```

%----- EMPLOYEE RECORDS -----
%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
IF (&PHA = ' ')
&PHA = 301

)PROC
VER (&LNAME,ALPHA)
VER (&FNAME,ALPHA)
VER (&I,ALPHA)
VER (&PHA,NUM)
VER (&PHNUM,PICT,'NNN-NNNN')

)END

```

Figure 6. Sample Panel with Assignment, IF, and VER Statements

Message Definitions

When the DISPLAY service is invoked, a message may optionally be displayed with the panel, or superimposed on the panel that is currently displayed. Messages are referenced by message id. Several messages may be contained within each member of the message library.

Each message consists of two lines. The first line contains the message id, and optionally:

- Short message text, enclosed in apostrophes ('')
- Corresponding help panel (if the user presses the Help PF key)
- Audible alarm indicator (yes or no).

The second line contains the long message text, enclosed in apostrophes.

If a short message is specified, it will be displayed first in the upper right-hand corner of the screen. If the user presses the Help PF key, the long message will then be displayed on line 3 of the screen. If the user presses the Help key again, tutorial mode will be entered.

If a short message is not specified, the long message will be displayed first (on line 3). If the user then presses the Help key, tutorial mode will be entered.

Substitutable parameters, consisting of a dialog variable name preceded by an ampersand (&), may appear anywhere within the short and long message text. Example:

```
'VOLUME &VOL NOT MOUNTED'
```

Figure 7 shows an example of a member in the message library. This member contains all message ids which begin with "EMPX21".

```
EMPX210 '&TYPECHG INVALID'      .HELP=PERS033  .ALARM=YES
'TYPE OF CHANGE MUST BE NEW, UPDATE, OR DELETE.'
```

```
EMPX213 'ENTER FIRST NAME'      .HELP=PERS034  .ALARM=YES
'EMPLOYEE NAME MUST BE ENTERED FOR TYPE OF CHANGE = NEW.'
```

```
EMPX214 'ENTER LAST NAME'       .HELP=PERS034  .ALARM=YES
'EMPLOYEE NAME MUST BE ENTERED FOR TYPE OF CHANGE = NEW.'
```

```
EMPX215 'ENTER HOME ADDRESS'    .HELP=PERS035  .ALARM=YES
'EMPLOYEE HOME ADDRESS MUST BE ENTERED FOR TYPE OF CHANGE = NEW.'
```

```
EMPX216 'AREA CODE INVALID'     .ALARM=YES
'AREA CODE &PHA IS NOT DEFINED. PLEASE CHECK THE PHONE BOOK.'
```

```
EMPX217 'CHANGE COMPLETED'
'EMPLOYEE &LNAME, &FNAME &I (&EMP SER) ADDED TO FILE.'
```

```
EMPX218 'CHANGE COMPLETED'
'RECORDS FOR &LNAME, &FNAME &I (&EMP SER) UPDATED.'
```

```
EMPX219 'CHANGE COMPLETED'
'RECORDS FOR &LNAME, &FNAME &I (&EMP SER) DELETED.'
```

Figure 7. Sample Member in Message Library

TABLE SERVICES

Table services allow dialog variables to be permanently stored across sessions. A table is a two-dimensional array of information in which each column corresponds to a dialog variable, and each row contains a set of values for those variables. An example is shown in Figure 8.

EMPSER	LNAME	FNAME	I	PHA	PHNUM
598304	Roberston	Richard	P	301	840-1224
172397	Smith	Susan	A	301	547-8465
813058	Russell	Charles	L	202	338-9557
395733	Adams	John	Q	202	477-1776
502774	Caruso	Vincent	J	914	294-1168

Figure 8. Sample Table

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

When a table is created, one or more columns (variable names) may be specified as keys for accessing the table. In the above example, EMPSER might be defined as the key variable for searching the table. Or EMPSER and LNAME might both be defined as the keys, in which case a row would be found only if EMPSER and LNAME both match the current values of those variables.

A table may also be accessed by one or more "argument" variables, which is similar to accessing by key, except that the search arguments may be dynamically defined.

In addition, a table may be accessed by "current row pointer" (CRP). When a table is opened, the CRP is automatically positioned to TOP -- ahead of the first row. The table may be scanned by moving the CRP forward or back and reading rows. The order of the rows is under control of the dialog function when new rows are added.

When a row is retrieved from a table, the contents of the row are stored into the corresponding dialog variables. When a row is stored (updated or added), the current contents of the dialog variables are saved in that row.

For any service that retrieves a row, a list of variable names may be obtained to determine what variables were retrieved. For any service that stores a row, a list of variable names may be provided. This allows the variables in the table to be extended beyond what was specified when the table was created.

A table may be defined as temporary or permanent. A temporary table is created in virtual storage, and deleted upon completion of processing. A permanent table resides on direct access storage. It may be opened for update or for read-only access. An ENQ is automatically issued to prevent multiple access to a table which is being updated.

General Services

The following services operate on an entire table:

TBCREATE Create a new table.
TBOPEN Prepare a table for processing.
TBQUERY Obtain information about a table.
TBSAVE Save a permanent copy of a table without closing.
TBCLOSE Close a table, and save a permanent copy.
TBEND Close a table without saving.
TBERASE Delete a table.

Row Operations

The following services operate on a row of the table:

TBADD Add a new row to the table.
TBDELETE Delete a row from the table.
TBGET Retrieve a row from the table.
TBPUR Update an existing row in the table.
TBMOD Update a row in the table if it exists (if the keys match); otherwise, add a new row to the table.
TBEXIST Test for the existence of a row (by key).
TBSCAN Search a table for a row that matches a list of "argument" variables, and retrieve the row.
TBSARG Establish a new search argument for use with TBSCAN.
TBTOP Set CRP to TOP (ahead of the first row).
TBBOTTOM Set CRP to BOTTOM and retrieve the last row.
TBSKIP Move the CRP forward or back by a specified number of rows, and then retrieve a row.
TBVCLEAR Set dialog variables (that correspond to variables in the table) to null.

Table Display Service

The following service displays information in a table and allows rows to be selected for processing.

TBDISPL Display selected columns from a table.

This service requires specification of a panel name, as well as a table name. The panel definition contains the non-scrollable text, including column headings. It also specifies the columns to be displayed, and whether each column is protected or unprotected (user modifiable). Typically, the left-most column is defined as a selection code. The code is interpreted by the dialog function to determine the particular processing for that row.

When the table is displayed, the user may scroll up and down, and enter information in unprotected fields (one row at a time).

FILE TAILORING SERVICES

File tailoring services read skeleton files from a library and write tailored output that may be used to drive other functions. Frequently, file tailoring is used to generate job files for batch execution.

The file tailoring output may be directed to a data set that has been allocated by the dialog function, or it may be directed to a temporary data set provided by SPF. The name of the temporary data set is available in system variable ZTEMPF.

Each skeleton file is read record by record. Each record is scanned to find any dialog variable names (preceded by an ampersand). When a variable name is found, its current value is substituted.

Skeleton file records may also contain statements that control processing. These allow the user to set dialog variables, imbed other skeleton files, conditionally include records, and iteratively process records in which variables from each row of a table are substituted.

A sample skeleton file is shown in Figure 9. It contains JCL statements for an assembly and optional load-go. The tailored output could be submitted to the background for execution.

The sample skeleton references several dialog variables (ASMPARMS, ASMIN, MEMBER, etc.). It also illustrates use of select statements ")SEL" and ")ENDSEL" to conditionally include records. The first part of the example has nested selects to include concatenated macro libraries if the library names have been specified by the user (i.e., if variables ASMMAC1 and ASMMAC2 are not equal to the null variable Z).

In the second part of the example, select statements are used to conditionally execute a load-go step. An imbed statement, ")IM", is used to bring in a separate skeleton for the load-go step.

The file tailoring services are:

FTOPEN	Prepares the file tailoring process, and specifies whether the temporary data set is to be used for output.
FTINCL	Specifies the skeleton to be used, and starts the tailoring process.
FTCLOSE	Ends the file tailoring process.
FTERASE	Erases (deletes) an output file created by file tailoring.


```

//ASM EXEC PGM=IFOX00,REGION=128K,
// PARM=(&ASMPARMS)
//SYSIN DD DSN=&ASMIN(&MEMBER),DISP=SHR
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR
)SEL &ASMMAC1 = &Z
// DD DSN=&ASMMAC1,DISP=SHR
)SEL &ASMMAC2 = &Z
// DD DSN=&ASMMAC2,DISP=SHR
)ENDSEL
)ENDSEL
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(5,2))
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSPRINT DD SYSOUT=(&ASMPRT)
)CM IF USER SPECIFIED "GO", WRITE OUTPUT IN TEMP DATA SET
)CM THEN IMBED "LINK AND GO" SKELETON
)SEL &GOSTEP = YES
//SYSGO DD DSN=&&&OBJSET,UNIT=SYSDA,SPACE=(CYL,(2,1)),
// DISP=(MOD,PASS)
)IM LINKGO
)ENDSEL
)CM ELSE (NOGO), WRITE OUTPUT TO USER DATA SET
)SEL &GOSTEP = NO
//SYSGO DD DSN=&ASMOUT(&MEMBER),DISP=OLD
)ENDSEL
/**

```

Figure 9. Sample Skeleton File

OTHER SERVICES

The EDIT and BROWSE interface services allow a dialog function to invoke the SPF editor or browse program, which are part of the program development facility. These services require specification of a data set name (and member, if applicable). The entry panel, which is displayed if edit or browse is selected from the primary option menu, is bypassed.

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

The CONTROL service allows a dialog function to condition SPF to expect certain kinds of display output, or to control the disposition of errors encountered by SPF services.

The display conditions are:

- LINE Expect line output, not generated by the dialog (e.g., generated by execution of a TSO command). Optionally, the starting line may be specified.
- REFRESH Refresh the entire screen on the next display. Typically used before or after invoking some other full-screen application which is not using SPF display services.
- SM Transfer to TSO Session Manager mode on the next line output.
- NONDISPL Do not display the next panel (process the panel without actually displaying it, and simulate the ENTER or End key.)

The disposition of errors may be controlled as follows:

- CANCEL Terminate the dialog function on a error (return code 12 or higher from any service). A message is displayed and logged prior to termination.
- RETURN Return control to the dialog function on all errors (with appropriate return code). A message id is stored in system variable ZERRMSG, which may be used by the dialog function to display and/or log a message.

The default disposition is CANCEL. If a dialog function sets the disposition to RETURN, the change affects only the current function. It does not affect lower level functions invoked via the SELECT service, nor a higher level function when the current function completes.

CHAPTER 3. PROGRAM DEVELOPMENT FACILITY

The SPF program development facility is designed to increase programmer productivity in an interactive environment. It can be used either by an individual programmer, or by many programmers working together on a project. Significant features include:

- Support for multilevel programming libraries. Facilitates maintenance and tracking of program segments at varying version and modification levels.
- Full-screen, context editing. Allows additions and changes to multiple lines in a single interaction. Simple one-character edit commands are used for inserting, deleting, duplicating, or rearranging lines of source data.
- Forward, backward, and sideways scrolling of source data or listings, plus the ability to locate information by character string or line number.
- Utilities for specification and maintenance of SPF libraries, data sets, catalogs, and DASD volumes.
- Interface to standard language processors (compilers, assembler, and linkage editor) for execution in the foreground or background.
- Document preparation support. Includes text editing features and an interface to the SCRIPT/VS Document Composition Facility.
- Online tutorial for instruction and reference -- especially valuable for the occasional or novice user.

The program development facility supports both structured and conventional programming techniques.

Structured programming emphasizes the use of segmentation and indentation. A source program normally consists of a large number of relatively small segments. The segments are stored as separate members of a programming library. Within a segment, source statements are block-indented under each "IF-THEN-ELSE" or "DO-WHILE" to show the control structure. Features which are specifically oriented toward structured programming include:

- Ease of segmentation changes. One segment (member) can easily be split into multiple segments, or multiple segments can be merged into one.
- Ease of indentation changes. Single statements or blocks of statements can easily be shifted left or right by a specified number of column positions.
- Insert in context. A "DO-END" pair, for example, may be coded on consecutive lines, and then space can be opened between the two lines to allow insertion of a block of code.
- Visual verification aids. A block of code may be temporarily excluded from display so that the space which it occupies on the screen is closed up. This facilitates visual verification of the control structure, particularly when the length of a segment exceeds the screen size.

GENERAL DESCRIPTION

Following is a description of features that are common across the various processing options of the program development facility.

DISPLAY FORMAT

Four basic types of display presentations are used:

1. **Selection Menus** - The user selects from a list of options by typing a one-character code and pressing the ENTER key. Example: The SPF primary option menu (Figure 10).
2. **Entry Panels** - The user supplies parameters by filling in labeled fields. In many cases, fields are pre-entered based on what the user last entered. Example: The browse entry panel (Figure 11).
3. **Member Selection Lists** - Displays a list of members in a programming library. The user may select a member by entering a one-character code in front of the appropriate member name. Figure 12 shows an example of a member list on which the user has selected member COINS.
4. **Data Display** - Displays source code or output listings. Figure 13 shows an example of the browse display.

All menus and panels are formatted to fit on a 24 line by 80 character screen. On a 3278 Model 3 or 4, scrollable data will occupy the full length of the screen (32 or 43 lines).

On a 3278 Model 5, information is normally displayed in "default" mode (24 by 80) with the same size characters as other models. Browse and edit data that is wider than 80 characters is displayed with the smaller "native" mode characters (up to 132 per line).

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

line 1	Title	Short Message
line 2	Prompt/Input	Scroll
line 3	Long Message	

The title area (line 1) identifies the function being performed and, where appropriate, the library or data set name, member name, version number, and modification level. The short message area (line 1) is used to indicate:

- Current line (browse) and column positions (browse and edit)
- Successful completion of a processing function
- Error conditions (accompanied by audible alarm).

The prompt/input area (line 2) is used to enter an option selection or command. In cases where no option selection or command is applicable, this area contains a prompt. The scroll area (line 2) contains the current scroll amount whenever scrolling is applicable. The scroll amount may be overtyped by the user (see "Scrolling").

The long message area (line 3) is used to display an explanation of error conditions upon request (see "Help Information"). Normally this line is blank on menus and panels, contains column headings on member selection lists, and is treated as part of the data area on data displays.

```

----- SPF-MVS PRIMARY OPTION MENU -----
SELECT OPTION ==> _

0  SPF PARMS - SPECIFY TERMINAL AND SPF PARAMETERS          USERID - JOSLIN
1  BROWSE   - DISPLAY SOURCE DATA OR OUTPUT LISTINGS      TIME   - 12:47
2  EDIT    - CREATE OR CHANGE SOURCE DATA                 TERMINAL - 3277
3  UTILITIES - PERFORM SPF UTILITY FUNCTIONS              PF KEYS - 12
4  FOREGROUND - COMPILE, ASSEMBLE, LINK EDIT, OR DEBUG
5  BACKGROUND - COMPILE, ASSEMBLE, OR LINK EDIT
6  COMMAND  - ENTER TSO COMMAND OR CLIST
7  SUPPORT  - TEST DIALOG OR CONVERT MENU/MESSAGE FORMATS
T  TUTORIAL - DISPLAY INFORMATION ABOUT SPF
X  EXIT    - TERMINATE SPF USING LIST/LOG DEFAULTS

PRESS END KEY TO TERMINATE SPF

```

Figure 10. Primary Option Menu

```

----- BROWSE - ENTRY PANEL -----
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
PROJECT ==> SPFDEMO
LIBRARY ==> MYLIB
TYPE   ==> PLI
MEMBER ==> _          (BLANK FOR MEMBER SELECTION LIST)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==>
VOLUME SERIAL ==>          (IF NOT CATALOGED)

DATASET PASSWORD ==>          (IF PASSWORD PROTECTED)

```

Figure 11. Browse - Entry Panel

```

BROWSE - SPFDEMO.MYLIB.PLI -----
COMMAND INPUT ==>

```

NAME	VER.MOD	CREATED	LAST MODIFIED	SIZE	INIT	MOD	SCROLL ==>	PAGE ID
ACCOUNT	01.00	79/01/09	79/01/09 17:07	21	21	0		HOSTETL
ACCT1	01.01	79/01/09	79/04/23 14:52	99	193	0		HOSTETL
ACCT2	01.00	79/01/09	79/01/09 17:07	20	20	0		HOSTETL
S COINS	01.04	79/04/24	79/04/28 16:20	19	19	4		JOSLIN
COMPX	01.00	79/01/09	79/01/09 17:08	44	44	0		JOSLIN
COMPY	01.01	79/01/14	79/01/16 12:30	13	13	1		JOSLIN
DCLS	01.00	79/04/23	79/04/23 15:14	20	20	0		SILVA
LISTNEM	01.02	79/04/23	79/05/06 15:00	17	13	6		EKMAN
MAIN	01.00	79/01/09	79/01/09 17:08	4	4	0		EKMAN
TESTDIR	01.02	79/04/23	79/05/06 17:04	30	43	10		HOSTETL
UPDATE	01.00	79/01/09	79/01/09 17:08	13	13	0		EKMAN

```

**END**

```

Figure 12. Browse - Member Selection List

```

BROWSE - SPFDEMO.MYLIB.PLI(COINS) - 01.04 ----- LINE 00000 COLS 001 080
COMMAND INPUT ==> _ SCROLL ==> PAGE
***** TOP OF DATA *****-CAPS ON-**
COINS:
PROCEDURE OPTIONS (MAIN);
DECLARE
COUNT FIXED BINARY (31) AUTOMATIC INIT (1),
HALVES FIXED BINARY (31),
QUARTERS FIXED BINARY (31),
DIMES FIXED BINARY (31),
NICKELS FIXED BINARY (31),
SYSPRINT FILE STREAM OUTPUT PRINT;
DO HALVES = 100 TO 0 BY -50;
DO QUARTERS = (100 - HALVES) TO 0 BY -25;
DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
NICKELS = 100 - HALVES - QUARTERS - DIMES;
PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);
COUNT = COUNT + 1;
END;
END;
END;
END COINS;
***** BOTTOM OF DATA *****-CAPS ON-**

```

Figure 13. Browse - Data Display

SPF LIBRARIES

An SPF library is a collection of code or data units, called members. Each library generally contains members with the same type of information. Figure 14 shows a sample set of five SPF libraries which contain Assembler source, COBOL source, object modules, load modules, and SCRIPT documentation.

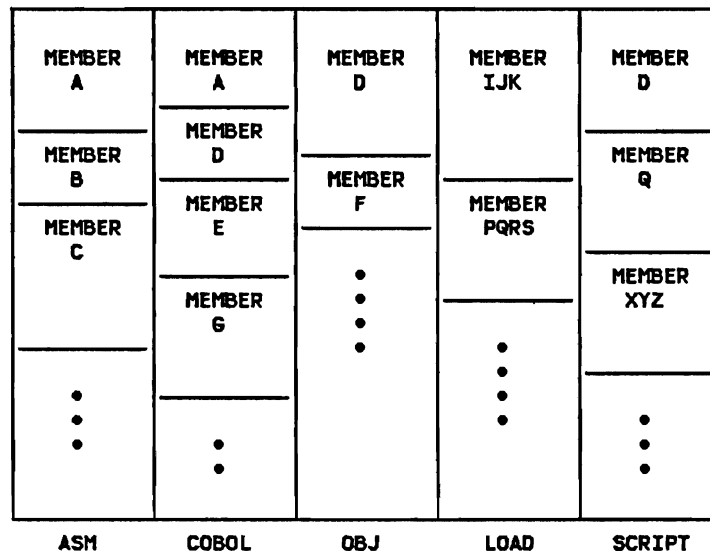


Figure 14. SPF Library Organization

In the TSO environment, each SPF library is a cataloged partitioned data set, with a three-level name of the form:

'project-name.library-name.type'

where --

- "project-name" is the common identifier for all libraries belonging to the same project.
- "library-name" identifies the particular set of libraries, such as MASTER, TEST, or RON.
- "type" identifies the type of information in the library, such as ASM, COBOL, OBJ, LOAD, or SCRIPT. Except for OBJ and LOAD, the SPF type qualifier need not conform to the standard TSO naming conventions.

Most projects will require a hierarchy of related libraries to maintain effective version control over the programming development process and to reduce contention in library usage. Figure 15 shows a sample three-level hierarchy for a project, consisting of a set of master libraries, a set of test libraries, and three sets of development (private) libraries identified by user id.

New members or members undergoing changes will generally reside in the development libraries. A test library may be used to accumulate members which have been unit tested and are ready for integration test. A master library contains fully tested members, which may correspond to a previously released version of the program. Typically, a higher degree of control is maintained over the higher level libraries than the development libraries.

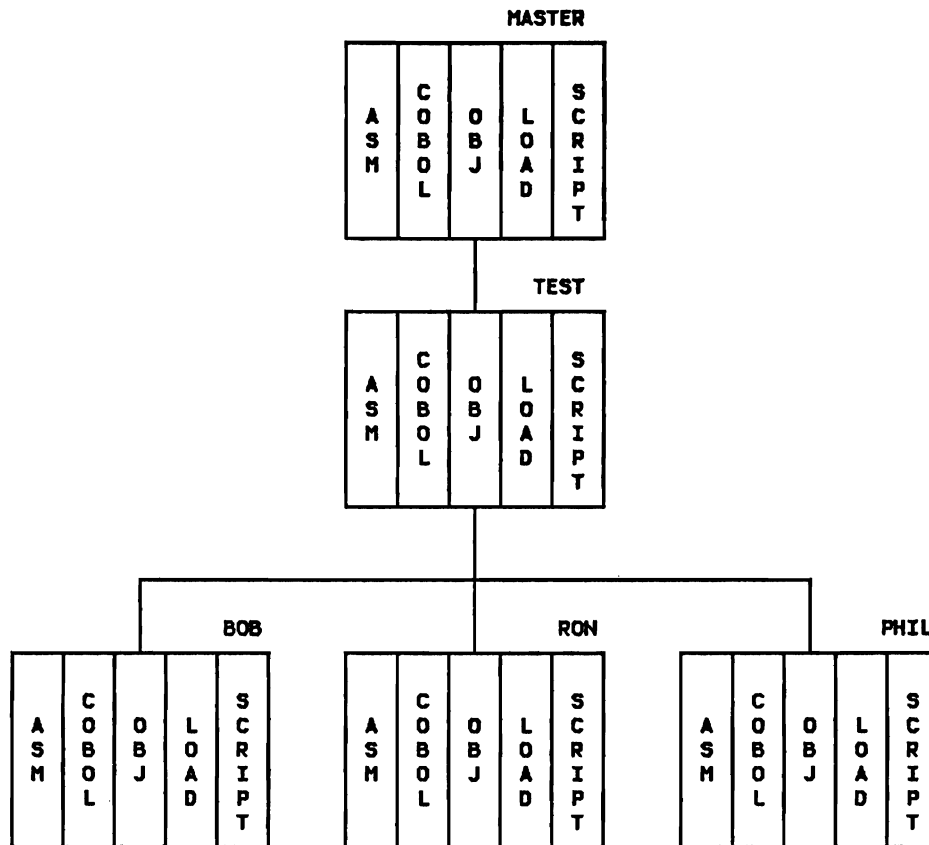


Figure 15. Hierarchy of SPF Libraries

SPF allows concatenation of up to four SPF libraries during source editing, compilation, assembly, link edit, or SCRIPT/VS processing. Generally, the lowest level library is concatenated ahead of the next higher level library, and so on, in bottom-to-top order. For the hierarchy shown in Figure 15, a typical concatenation sequence might be library BOB, followed by TEST, followed by MASTER. The concatenation applies to libraries of the same type, and is restricted to libraries which belong to the same project.

The purpose of concatenation during editing is to provide downward copying of a member from a test or master library to a development library. The edited member is saved in the development library (the first library in the concatenation sequence), while the unchanged version remains in the test or master library. When the new version is fully tested, it may be promoted back to a higher level library by means of the move/copy utility.

The purpose of concatenation during language processing is to facilitate inclusion of source segments via INCLUDE or COPY statements (or SCRIPT "imbed" controls), and to allow debugging of new or modified programs without altering the contents of the test or master libraries. The output from a compilation or assembly (object module) or link edit (load module) is stored in the lowest level OBJ or LOAD library (the first library in the concatenation sequence).

PARAMETER SPECIFICATION

Several entry panels require specification of the following:

- Library and data set names
- Data set password (if applicable)
- Job statement information (for submitting background jobs).

These are described in the following paragraphs.

Library and Data Set Names

To specify a member of an SPF library, the user must enter a project name, library name, type qualifier, and member name. Each of these may contain up to eight alphanumeric characters, of which the first character must be alphabetic.

SPF panels prompt the user for each component of the library identification as follows:

```
SPF LIBRARY:
PROJECT ===>
LIBRARY ===>
TYPE    ===>
MEMBER  ===>          (BLANK FOR MEMBER SELECTION LIST)
```

For convenience, any cataloged data set (sequential or partitioned) with a three-level name may be entered in this manner, even if the three components of the data set name do not actually identify project, library, and type. For partitioned data sets, if the member name is not specified, a member selection list will be displayed.

Exception: In foreground and background processing, options 4 and 5, only a partitioned data set may be specified, and the member name must be specified.

Some panels allow a concatenated sequence of up to four SPF libraries. For example:

```
SPF LIBRARY:
PROJECT ===> SAMPLE
LIBRARY ===> MYOWN   ===> TEST   ===> MASTER   ===>
TYPE    ===> COBOL
MEMBER  ===>          (BLANK FOR MEMBER SELECTION LIST)
```

In this example, three libraries would be concatenated in the following order:

```
SAMPLE.MYOWN.COBOL
SAMPLE.TEST.COBOL
SAMPLE.MASTER.COBOL
```

It is a user responsibility to insure that the concatenated libraries have consistent record formats, logical record lengths, and block sizes.

Partitioned and sequential data sets may also be specified using standard TSO syntax, as follows:

<p>OTHER PARTITIONED OR SEQUENTIAL DATA SET: DATA SET NAME ==> VOLUME SERIAL ==> (IF NOT CATALOGED)</p>

Any fully qualified data set name may be entered, enclosed within apostrophes. If the apostrophes are omitted, the TSO user prefix is automatically left-appended to the data set name. For partitioned data sets, a member name enclosed in parentheses may follow the data set name. For example:

'SYS1.PROCLIB(ASMHC)'

If the member name (and parentheses) are omitted, a member selection list will be displayed.

If both an SPF library and "other" data set name are specified on the same panel, the "other" data set name is used. In other words, to specify an SPF library, the other data set name must be blank.

The volume serial parameter may specify a real DASD volume or a virtual volume residing on an IBM 3850 Mass Storage System. To access 3850 virtual volumes, the user must also have MOUNT authority (see TSO ACCOUNT command).

Note: SPF does not support multivolume data sets.

Data Set Password

SPF supports the use of OS password protection. Any data set may be protected for read-only or read/write access. More than one password may be assigned to the same data set. A data set that is read/write protected, for example, might allow several authorized users to read it, but only one user to write.

A data set password may be specified in the following manner:

<p>DATA SET PASSWORD ==> (IF PASSWORD PROTECTED)</p>

A non-display input field is used so that the password will not appear on the screen. For entry panels that allow specification of a concatenated sequence of libraries, the password applies to all data sets in the concatenation sequence.

Note: SPF may be used with either the TSO/VS2 Programming Control Facility (PCF) or the Resource Access Control Facility (RACF). PCF and RACF provide extensive facilities for protecting data set security. With PCF or RACF the user should not attempt to enter a password on the SPF panels, since both of these facilities rely on the TSO user id and (logon) password to identify the user and check for proper authorization.

Job Statement Information

SPF supports submission of background jobs for printing and language processing. Before submitting a background job, the user must supply a job statement. For this purpose, four lines are provided on each job submission panel. The first time a job submission panel is displayed, the job statement information appears as follows:

```
JOB STATEMENT INFORMATION:
===> //user-ida JOB (ACCOUNT),'NAME'
===> /**
===> /**
===> /**
```

The pre-entered job name consists of the user id right-appended with the letter "A". The last character of the job name is automatically incremented to "B", "C", etc., each time the job statement information is used. The user may change the last character of the job name to a numeric digit, rather than a letter, in which case the last character will be cycled from 0 to 9, rather than A to Z.

The user may overwrite the entire job name, if desired, but automatic incrementing of the last character is suspended unless the job name starts with the user id.

The remaining job statement information must be entered by the user the first time a background job is submitted. The lines containing "/*" may be used as continuation lines (by changing the "/*" to "/*"), or they may be used to enter other JCL statements, such as a JOBLIB DD statement. If these lines are not needed, they may be blanked out. Blank lines are not submitted to the job stream.

Saving User Parameters

SPF pre-enters appropriate information on panels according to what was last entered by the user. In many cases, default values are provided if the user has not specified otherwise. This includes the following types of information:

- Project name, library name(s), and type
- Job statement information
- SYSOUT class for printed output
- Defaults for list/log allocation and processing
- Terminal characteristics and PF key definitions
- Edit profiles, including mask, tabs, and bounds
- Processing options for each of the language processors
- Data set allocation/information parameters

This information is automatically maintained from one session to another. A new user will have to enter certain information the first time it appears. But from that point on, the user simply verifies the information and makes whatever changes are desired before proceeding.

PROGRAM ACCESS AND FUNCTION KEYS

The program access (PA), and program function (PF) keys are used to request commonly used operations. Keys that are not needed for SPF-defined key operations may be equated to edit and browse primary commands, or edit line commands, using the SPF parms option (option 0.3).

The default key arrangement for the 3-by-4 key pad (right-hand side of the keyboard) is shown in Figure 16. These are PF keys 1-12 on a 12-key terminal, or keys 13-24 on a 24-key terminal.

For 24-key terminals, PF keys 1-12 have the same defaults as keys 13-24. It is strongly recommended that users of 24-key terminals continue to use the key pad (13-24) for SPF-defined operations, and redefine PF keys 1-12 to edit and browse commands.

PA1 ATTENTION	PA2 RESHOW	PF1 / 13 HELP	PF2 / 14 SPLIT	PF3 / 15 END
		PF4 / 16 RETURN	PF5 / 17 FIND	PF6 / 18 CHANGE
		PF7 / 19 ↑ (UP)	PF8 / 20 ↓ (DOWN)	PF9 / 21 SWAP
		PF10 / 22 ← (LEFT)	PF11 / 23 → (RIGHT)	PF12 / 24 CURSOR

Figure 16. Default Program Key Arrangement

The two PA keys are defined as follows. They may not be redefined by the user.

ATTENTION (PA1) This key is logically disabled whenever the keyboard is unlocked, except during execution of a TSO command under SPF (foreground or command option).

RESHOW (PA2) Redisplays the contents of the screen. May be useful if the ERASE or CLEAR key was pressed accidentally, or if unwanted information has been typed but ENTER (or a PF key) has not yet been pressed. Note that SPF does not support use of the Field Mark character (same key as PA2).

The SPF-defined PF key operations are described below. See Figure 16 for the default key assignments.

- HELP Displays additional information about an error message or tutorial information about SPF commands and options.
- SPLIT Causes split screen mode to be entered, or changes the location of the split line (see "Split Screen").
- END Terminates the current operation and returns to the previous menu. If the primary option menu is displayed, this key terminates SPF.
- RETURN Causes an immediate return to the primary option menu, bypassing any intermediate menus. (Logically equivalent to repeated use of the End key.) May also be used to go directly from one option to another, without displaying the primary option menu, as follows: In any panel input field or primary command line, enter an equal sign (=) followed by a primary option. Then press the Return PF key rather than the ENTER key.
- FIND Repeats the action of the previous FIND command or the FIND part of the most recent CHANGE command (applies to browse and edit only).
- CHANGE Repeats the action of the previous CHANGE command (applies to edit only).
- UP Causes a scroll up (see "Scrolling").
- DOWN Causes a scroll down (see "Scrolling").
- SWAP Moves the cursor to wherever it was previously positioned on the other logical screen (see "Split Screen").
- LEFT Causes a scroll left (see "Scrolling").
- RIGHT Causes a scroll right (see "Scrolling").
- CURSOR Moves the cursor to the first input field on line 2 (normally, the option selection or command input field). Pressing this PF key again causes the cursor to be moved to the second input field on line 2, if any (normally the scroll amount field).
- PRINT Causes a "snapshot" of the screen image to be recorded in the SPF list file.
- PRINT-HI Same as PRINT except that high intensity characters on the screen are printed with overstrikes to simulate the dual intensity display. Should not be used if list output is to be printed on a "local" 328x printer or other device which does not support suppress-space carriage control.
- NOP Causes the PF key to be functionless.

The PRINT, PRINT-HI, and NOP functions have no default PF key assignments.

The only PF key function that is required is the End key. Other keys may be assigned to edit and browse commands, or to PRINT, PRINT-HI, or NOP.

SCROLLING

During edit and browse, the information to be displayed will generally exceed the screen size. Scrolling allows the screen "window" to be moved up, down, left, or right across the information. A member list can also be scrolled up and down, if it exceeds a single screen length.

Four program function (PF) keys are used for scrolling -- one for each direction. Whenever scrolling is allowed, a scroll amount is displayed at the top of the screen (line 2). This determines the number of lines (or columns) scrolled with each use of a Scroll PF key. The user may change the scroll amount by moving the cursor to the scroll field and overtyping the displayed amount. Valid scroll amounts are:

- A number from 1 to 9999 - specifies the number of lines (up or down) or columns (left or right) to be scrolled.
- PAGE - specifies scrolling by one page.
- HALF - specifies scrolling by a half page.
- MAX - specifies scrolling to the top, bottom, left margin, or right margin, depending upon which Scroll PF key is used.
- CSR - specifies scrolling based on the current position of the cursor. The line or column indicated by the cursor is moved to the top, bottom, left margin, or right margin of the screen, depending upon which Scroll PF key is used. If the cursor is not in the body of the data, or if it is already positioned at the top, bottom, left margin, or right margin, a full page scroll will occur.

For scrolling purposes, a "page" is defined as the amount of information currently visible on the logical screen. In split screen mode, for example, a browse display might have 12 lines by 80 columns of scrollable data. In this case, a scroll amount of HALF would move the window up or down by 6 lines, or right or left by 40 columns.

For member lists and browsing, the scroll amount is initialized to PAGE. For editing, the scroll amount is initialized to HALF. When the user overtypes the amount, the new value remains in effect until it is again changed by the user, or until a new member list or new data is displayed. The value MAX is an exception; following a MAX scroll, the scroll amount reverts to its previous value.

Any valid scroll amount can also be entered in the command input field and used in conjunction with a Scroll PF key. For example:

ENTER COMMAND ==> 3

SCROLL ==> HALF

If a Scroll PF key is pressed, the value in the command area will be used, overriding the normal scroll amount, without causing a change to the scroll field. This results in a one-time override; the value in the command area is blanked out after scrolling occurs.

If some key other than a Scroll PF key is pressed, the value in the command area will be interpreted as a command and will probably result in an error message.

OVERTYPING OF SCROLL AND MULTIPLE CHOICE PARAMETERS

To reduce keystrokes, the scroll amount field can be changed by overtyping the first character(s) only:

- To change the scroll amount to PAGE, HALF, MAX, or CSR only the first character need be overtyped with "P", "H", "M", or "C" respectively.
- In the scroll amount field, any alphabetic characters following a number are ignored. For example "3AGE" is interpreted as "3".

The same rule applies to multiple choice parameters in cases where the first letter uniquely defines the option. Examples:

REPLACE MEMBERS	==> NO	(YES OR NO)
DISPOSITION	==> KEEP	(KEEP OR DELETE)

To change these options to YES or DELETE, only the first character need be overtyped with "Y" or "D" respectively.

SPLIT SCREEN

Split screen mode allows the user to partition the display screen into two logical screens. The top and bottom screens are treated as though they were independent terminals. Functions that can be performed simultaneously on two separate terminals can be done on the top and bottom screens. Functions that are in conflict on two separate terminals (editing the same member of an SPF library or other partitioned data set, or editing the same sequential data set) will also be in conflict when simultaneously attempted on the top and bottom screens.

Split screen mode is entered by pressing the Split PF key, but first the user moves the cursor to the desired location for the split. The line containing the cursor will become the split line and will be identified by a row of periods. When the Split PF key is pressed, all lines below the cursor are treated as the bottom screen. The first display to appear on the bottom screen is the primary option menu. An example is shown in Figure 17.

Note: Since only one logical screen is active at a time, the user cannot split the screen while executing a foreground compilation or a TSO command.

The split location may be repositioned at any time by moving the cursor to the desired line and pressing the Split PF key again. Repositioning may be required to bring an entire panel or tutorial page into view.

In split screen mode, one or the other of the logical screens is considered active at any point in time. Any interrupts, such as the operation of program function keys, are interpreted as having meaning for the active screen. The location of the cursor identifies which of the two screens is active. To switch from one screen to the other, the cursor movement keys may be used or the Swap PF key may be pressed.

In addition to moving the cursor to the other logical screen, the Swap PF key will automatically reposition the split if either logical screen is less than five lines long. For example, if the split occurs at the third line down from the top and the cursor is on the bottom screen, pressing the Swap PF key will move the cursor to the top screen and reposition the split to the third line up from the bottom. This allows the use of two screens in "flip-flop" mode, with each logical screen consuming nearly all lines of the physical screen.

```

BROWSE - SPFDemo.MYLIB.PLI(COINS) - 01.04 ----- LINE 00000 COLS 001 080
COMMAND INPUT ==>                               SCROLL ==> PAGE
***** TOP OF DATA *****-CAPS ON-**
COINS:                                           00010001
  PROCEDURE OPTIONS (MAIN);                     00020000
  DECLARE                                       00030000
    COUNT    FIXED BINARY (31) AUTOMATIC INIT (1), 00040000
    HALVES   FIXED BINARY (31),                 00050000
    QUARTERS FIXED BINARY (31),                 00060003
  . . . . .
----- SPF-MVS PRIMARY OPTION MENU -----
SELECT OPTION ==> _
                                USERID - JOSLIN
0  SPF PARMS - SPECIFY TERMINAL AND SPF PARAMETERS  TIME - 12:47
1  BROWSE   - DISPLAY SOURCE DATA OR OUTPUT LISTINGS  TERMINAL - 3277
2  EDIT    - CREATE OR CHANGE SOURCE DATA           PF KEYS - 12
3  UTILITIES - PERFORM SPF UTILITY FUNCTIONS
4  FOREGROUND - COMPILE, ASSEMBLE, LINK EDIT, OR DEBUG
5  BACKGROUND - COMPILE, ASSEMBLE, OR LINK EDIT
6  COMMAND  - ENTER TSO COMMAND OR CLIST
7  SUPPORT  - TEST DIALOG OR CONVERT MENU/MESSAGE FORMATS
T  TUTORIAL - DISPLAY INFORMATION ABOUT SPF
X  EXIT    - TERMINATE SPF USING LIST/LOG DEFAULTS

PRESS END KEY TO TERMINATE SPF

```

Figure 17. Split Screen Example

Split screen mode is terminated by ending SPF processing on either logical screen (i.e., by pressing the End PF key or by entering option "X" when the primary option menu is displayed). The remaining logical screen is then expanded to its full size. Split screen may also be terminated by typing =X (exit option) in the command input area and pressing the Return PF key.

HELP INFORMATION

The help function allows the user to obtain additional information about a message that has been displayed in the upper right-hand corner of the screen, or general information about an SPF command or option.

If a message is displayed, pressing the Help PF key causes a one-line explanation to be displayed. If this explanation is not sufficient, the user may obtain further information by pressing the Help key again. This causes an entry into the appropriate section of the tutorial. If a message is not displayed, the Help PF key causes a direct entry into the appropriate section of the tutorial.

Once in the tutorial, the End PF key causes a return to the screen that was being viewed when the Help key was originally pressed.

LIBRARY ACTIVITY STATISTICS

Under user option, the SPF editor will automatically generate and maintain the following activity statistics for each member of an SPF library or other partitioned data set.

Version Number: Initialized to 1 when the member is created.

Modification Level: Number of times this version has been modified.

Creation Date: When this version was created.

Date/Time Modified: When this version was last modified.

Current No. Lines: Current size (number of records).

Initial No. Lines: Initial size of this version.

No. Modified Lines: Number of lines added or changed since this version was created. (Zero for unnumbered data.)

User Id: Who created or last updated this version.

The version number and/or user id may be changed via the "reset SPF statistics" utility (option 3.5). The statistics are displayed next to each member name on member selection lists, and may be printed by requesting an "index listing" via the library utility (option 3.1).

LIST AND LOG FILES

SPF helps the user obtain hardcopy listings of source modules. Under user option, a listing of any source module that is created or modified by the SPF editor may be automatically recorded for printing. Source listings and other types of printed output may also be obtained from the SPF utilities. A screen "snapshot" may be obtained by pressing the Print PF key.

SPF also maintains a log of significant user activities, including a record of data sets (and members) that were modified via edit or utility options, foreground and background processing activity, and command entry.

This information is accumulated in two temporary data sets, referred to as the SPF list and log files. They are allocated automatically when needed. Once allocated, the list and log files remain open throughout the session. When the user terminates SPF, the contents of these files may be printed.

The printed output may be directed to a system output device via submission of a background job, or it may be directed to a local IBM 3284, 3286, 3287, 3288, or 3289 printer. For local printing, the TSO Command Processor "DSPRINT" must be installed.

INVOCATION AND TERMINATION

The SPF program development facility is invoked by entering the command "ISPF". (An SPF command may be established as an alias of ISPF). The first display is the primary option menu (Figure 18). The user may select an option by typing a one-character code in the option field and pressing the ENTER key, e.g.,

```
SELECT OPTION ==> 2
```

to select the edit option.

```
----- SPF-MVS PRIMARY OPTION MENU -----
SELECT OPTION ==> _

0  SPF PARMS  - SPECIFY TERMINAL AND SPF PARAMETERS      USERID  - JOSLIN
1  BROWSE    - DISPLAY SOURCE DATA OR OUTPUT LISTINGS  TIME    - 12:47
2  EDIT      - CREATE OR CHANGE SOURCE DATA            TERMINAL - 3277
3  UTILITIES - PERFORM SPF UTILITY FUNCTIONS           PF KEYS - 12
4  FOREGROUND - COMPILE, ASSEMBLE, LINK EDIT, OR DEBUG
5  BACKGROUND - COMPILE, ASSEMBLE, OR LINK EDIT
6  COMMAND   - ENTER TSO COMMAND OR CLIST
7  SUPPORT   - TEST DIALOG OR CONVERT MENU/MESSAGE FORMATS
T  TUTORIAL  - DISPLAY INFORMATION ABOUT SPF
X  EXIT      - TERMINATE SPF USING LIST/LOG DEFAULTS

PRESS END KEY TO TERMINATE SPF
```

Figure 18. Primary Option Menu

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

```
ISPF 2    - to go directly to edit
ISPF 3.1  - to go directly to utility suboption 1
```

The SPF primary options are:

- **SPF PARMS** - To specify SPF parameters and defaults, including: terminal type, number of program function (PF) keys, default pad character for input fields, allocation parameters and default dispositions for list and log files, and PF key definitions.
- **BROWSE** - To display source data or output listings. Browse is intended primarily for viewing large data sets such as compiler listings or dumps.
- **EDIT** - To create or change source data, including program code, test data, or documentation. Unlike browse, edit reads the selected member (or entire sequential data set) into virtual storage and retains it there during edit operations.
- **UTILITIES** - To print, rename, or delete library members or entire data sets; allocate data sets; move or copy data; display or print catalog listings or VTOCs; reset SPF library statistics; initiate hardcopy output; examine held SYSOUT data; or format SCRIPT/VS documentation.
- **BACKGROUND** - To execute TSO prompters and language processing programs in the foreground, including: Assembler, COBOL, FORTRAN, PL/I (checkout or optimizer), PASCAL, linkage editor, and COBOL or FORTRAN interactive debug.
- **BACKGROUND** - To generate and submit JCL for background execution of IBM language processing programs, including: Assembler, COBOL, FORTRAN, PL/I (checkout or optimizer), PASCAL, and linkage editor.
- **COMMAND** - To enter a TSO command or command procedure (CLIST) during execution of SPF.
- **SUPPORT** - To test a panel or dialog function; set or examine dialog variables; convert old format selection and tutorial menus to new format panels; or convert old format messages to new format.
- **TUTORIAL** - To obtain immediate online instruction in the use of SPF. The tutorial may be viewed sequentially from beginning to end, or randomly by selecting topics from the table of contents or alphabetized index. The tutorial may also be entered from other SPF options by means of the Help PF key.

To terminate SPF, the user must be out of split screen mode. Two termination options are available from the primary option menu:

1. Option "X" may be entered. This terminates SPF with user defaults for processing list and log files, as specified via SPF parms (option 0.2).

Note: Option "X" may also be used with the Return PF key to immediately terminate split screen or, in single screen mode, to immediately terminate SPF from any menu or entry panel.

2. The End PF key may be pressed to display the SPF termination panel for specification of list and log processing.

The termination panel allows the user to specify whether the list and log files are to be printed via a background job or routed to a local printer, or deleted or kept without printing. If the files are to be printed, the destination (SYSOUT class or printer id) must be specified, as well as job statement information for background printing.

Once the termination options have been entered, the ENTER key is pressed to complete termination. The screen is cleared, one or more termination messages is displayed, followed by a READY message, and the user is returned to TSO.

MEMBER LISTS

For any of the following options, a member selection list will be displayed if an SPF library or other partitioned data set is specified but no member name is supplied:

- Browse
- Edit
- Library Utility
- Move/Copy Utility
- Reset SPF Statistics Utility
- SCRIPT/VS Utility

An example of a member list display is shown in Figure 19. The member selection list also shows the SPF statistics for each member of an SPF library.

```
BROWSE - SPFDEMO.MYLIB.PLI -----
COMMAND INPUT ==> -
NAME          VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT  MOD  ID
ACCOUNT       01.00  79/01/09  79/01/09 17:07    21    21    0  HOSTETL
ACCT1         01.01  79/01/09  79/04/23 14:52    99   193    0  HOSTETL
ACCT2         01.00  79/01/09  79/01/09 17:07    20    20    0  HOSTETL
COINS         01.04  79/04/24  79/04/28 16:20    19    19    4  JOSLIN
COMPX         01.00  79/01/09  79/01/09 17:08    44    44    0  JOSLIN
COMPY         01.01  79/01/14  79/01/16 12:30    13    13    1  JOSLIN
DCLS          01.00  79/04/23  79/04/23 15:14    20    20    0  SILVA
LISTNEW       01.02  79/04/23  79/05/06 15:00    17    13    6  EKMAN
MAIN          01.00  79/01/09  79/01/09 17:08     4     4    0  EKMAN
TESTDIR       01.02  79/04/23  79/05/06 17:04    30    43   10  HOSTETL
UPDATE        01.00  79/01/09  79/01/09 17:08    13    13    0  EKMAN
**END**
```

Figure 19. Member List Display

The member list may be scrolled up and down by means of the Scroll PF keys. In addition, a LOCATE command may be entered in the command input field on line 2 of any member list display.

For browse, edit, and the SCRIPT/VS utility, one member at a time may be selected from the list, either by entering the single character "S" in front of the desired member name, or by entering a SELECT command in the command input field on line 2.

For the other utility options, multiple members may be selected from the list. A single character code may be typed in front of one or more member names before pressing the ENTER key. For the move/copy and reset utilities, the selection code is "S". For the library utility, the allowable codes are "P" (for print), "R" (for rename), "D" (for delete), and "B" (for browse).

SPF PARMS (OPTION 0)

The SPF parms option allows the user to display and change a variety of SPF parameters at any time during the session. Changes remain in effect until the user changes the parameter again, and are remembered across sessions. The parameter options menu is shown in Figure 20.

```
----- SPF PARAMETER OPTIONS -----  
SELECT OPTION ==> _  
  
1  TERMINAL - SPECIFY TERMINAL CHARACTERISTICS  
2  LIST/LOG - SPECIFY SPF LIST AND LOG DEFAULTS  
3  PF KEYS  - SPECIFY PF KEYS FOR 3277 TERMINAL WITH 12 PF KEYS
```

Figure 20. SPF Parameter Menu

The SPF parms option allows the user to specify:

1. Terminal Characteristics
 - Type of terminal
 - Number of program function keys on the terminal (12 or 24)
 - Default pad character for input fields.
2. List and Log Defaults
 - Default disposition (print, keep, or delete)
 - Method of printing and destination of output
 - Lines per page
 - Data set allocation parameters.
3. Program Function (PF) Key Assignments
 - For a 3277 (or 3275) with 12 PF keys
 - For a 3278 (or 3276 or 3279) with 12 PF keys
 - For a 3278 (or 3276 or 3279) with 24 PF keys

Before assigning PF keys (option 0.3), the user should ensure that the correct terminal type and number of PF keys has been specified via option 0.1.

The PF key assignment panel for a 24-key terminal is shown in Figure 21. The user may equate a PF key to an edit or browse primary command, or to an edit line command, in the following manner:

>string Causes the specified PF key, when used in edit or browse, to simulate the entering of a primary command. Example:

PF10 ==> >FIND ABC

Pressing PF10 when browsing or editing data will have the same effect as entering a FIND ABC primary command.

:string Causes the specified PF key, when used in edit, to simulate the entering of a line command. Example:

PF11 ==> :TF

Pressing PF11 when editing data will have the same effect as entering the "TF" (text flow) line command on whichever line the cursor is positioned when the key is pressed.

```

----- PROGRAM FUNCTION KEY DEFINITION -----
ENTER/VERIFY PARAMETERS BELOW FOR 3278 TERMINAL WITH 24 PF KEYS:

PF1 ==> HELP_
PF2 ==> SPLIT
PF3 ==> END
PF4 ==> RETURN
PF5 ==> FIND
PF6 ==> CHANGE
PF7 ==> UP
PF8 ==> DOWN
PF9 ==> SWAP
PF10 ==> LEFT
PF11 ==> RIGHT
PF12 ==> CURSOR

PF13 ==> HELP
PF14 ==> SPLIT
PF15 ==> END
PF16 ==> RETURN
PF17 ==> FIND
PF18 ==> CHANGE
PF19 ==> UP
PF20 ==> DOWN
PF21 ==> SWAP
PF22 ==> LEFT
PF23 ==> RIGHT
PF24 ==> CURSOR

VALID OPTIONS: (BLANK RESETS TO DEFAULT OPTION)
HELP SPLIT END RETURN FIND CHANGE
UP DOWN SWAP LEFT RIGHT CURSOR
NOP PRINT PRINT-HI

OR EQUATE TO EDIT/BROWSE COMMAND:
>CMD... (PRIMARY COMMAND) EXAMPLE: PF10 ==> >TABS ON
:CMD... (EDIT LINE COMMAND) EXAMPLE: PF11 ==> :I

```

Figure 21. PF Key Assignment Panel (For 24-key Terminals)

BROWSE (OPTION 1)

The browse option allows the user to display source data and listings stored in SPF libraries or other partitioned or sequential data sets with the following characteristics:

Record Format (RECFM):

- Fixed, variable (non-spanned), or undefined
- Blocked or unblocked
- With or without printer control characters

Logical Record Length (LRECL): Up to 32,767

When browse is selected, an entry panel is displayed to allow the user to specify an SPF library or other data set, volume serial (if the data set is not cataloged) and data set password (if the data set is protected). See Figure 22.

For SPF libraries and other partitioned data sets, the user can supply the name of the member to be browsed, or leave the member name blank to request a member list from which a member may be selected.

The browse data display is shown in Figure 23. Two lines are reserved at the top of the screen for title information, short messages, command entry, and the scroll amount. The remainder of the screen contains the data.

During browse, four-way scrolling is available via the Scroll PF keys. The FIND and LOCATE commands may also be used to scroll to a particular character string, line number, or symbolic label.

Browse provides the following commands, which may be entered in the command input field on line 2:

- LOCATE Scroll to the specified line number or label.
- .xxxxx Establish a label (xxxxx) for the LOCATE command.
- COLS Display a column identification line.
- RESET Remove column identification line.
- FIND Find a specified character string.
- CAPS Set CAPS mode on or off (affects FIND command strings).
- HEX Display data in hexadecimal format.

```

----- BROWSE - ENTRY PANEL -----
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
  PROJECT ==> SPFDEMO
  LIBRARY ==> XXX
  TYPE    ==> COBOL
  MEMBER  ==> CBLMAIN_ (BLANK FOR MEMBER SELECTION LIST)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
  DATASET NAME ==>
  VOLUME SERIAL ==> (IF NOT CATALOGED)

DATASET PASSWORD ==> (IF PASSWORD PROTECTED)

```

Figure 22. Browse - Entry Panel

```

BROWSE - SPFDEMO.XXX.COBOL(CBLMAIN) - 01.01 ----- LINE 00000 COLS 001 080
COMMAND INPUT ==> _ SCROLL ==> PAGE
***** TOP OF DATA *****-CAPS ON-**
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. 'F20D1000'.
000300 DATE-COMPILED. AUG. 20, 1979
000400 ENVIRONMENT DIVISION.
000500 CONFIGURATION SECTION.
000600 SOURCE-COMPUTER. IBM-370.
000700 OBJECT-COMPUTER. IBM-370.
000800 INPUT-OUTPUT SECTION.
000900 FILE-CONTROL.
001000     SELECT OLD-COMREC ASSIGN TO DA-S-DD1.
001100     SELECT D1-REPORTS ASSIGN TO UR-S-D1OUT.
001200     SELECT OPTION-CARD-FILE ASSIGN TO UR-S-SYSIN.
001300 DATA DIVISION.
001400 FILE SECTION.
001500 FD  OLD-COMREC
001600     LABEL RECORD IS STANDARD
001700     RECORDING MODE IS F
001800     BLOCK CONTAINS 0 RECORDS
001900     DATA RECORD IS COMREC1.
002000 01  COMREC1.
002100     02 DUMMY             PICTURE X(520).
002200 FD  D1-REPORTS
002300     LABEL RECORD IS OMITTED
002400     RECORDING MODE IS F
002500     BLOCK CONTAINS 133 CHARACTERS

```

Figure 23. Browse - Data Display

EDIT (OPTION 2)

The edit option allows the user to create, display, and modify source data (program code, test data, documentation, etc.) stored in SPF libraries or other partitioned or sequential data sets with the following characteristics:

Record Format (RECFM):

- Fixed or variable (non-spanned)
- Blocked or unblocked
- With or without printer control characters

Logical Record Length (LRECL):

- From 10 to 255, inclusive, for fixed length records
- From 14 to 259, inclusive, for variable length records

The edit entry panel is shown in Figure 24. For edit, a concatenated sequence of SPF libraries may be specified. The concatenation applies to the fetching of members to be edited. The libraries are searched in the designated order to find the member and bring it into working storage. When the edited member is saved, it is placed (or replaced) in the first library in the concatenation sequence regardless of which library it came from.

The panel also allows specification of a profile name, which may be entered to override the default edit profile. See description under "Edit Modes and Profiles."

Space for the selected data set must have been previously allocated, but it may be empty. Selection of an empty sequential data set or a nonexistent member of a partitioned data set allows creation of new source data.

The selected member or sequential data set is read into virtual storage, wherein it is updated during edit operations. Use of virtual storage for edit work space results in high performance, but may require a large user region.

The edit data display is similar to a browse display except that each line consists of a 6-column line-number field followed by a 72-column data field (see Figure 25). The line-number field reflects the contents of the sequence numbers in the data, if the data is numbered. For unnumbered data, the line numbers start at 1 and are incremented by 1.

```

----- EDIT - ENTRY PANEL -----
ENTER/VERIFY PARAMETERS BELOW:

SPF LIBRARY:
PROJECT ==> SPFDEMO
LIBRARY ==> MYLIB      ==> MASTER      ==>
TYPE ==> PLI
MEMBER ==> _          (BLANK FOR MEMBER SELECTION LIST)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==>
VOLUME SERIAL ==>          (IF NOT CATALOGED)

DATASET PASSWORD ==>          (IF PASSWORD PROTECTED)

PROFILE NAME ==>          (BLANK DEFAULTS TO DATASET TYPE)

```

Figure 24. Edit - Entry Panel

```

EDIT --- SPFDEMO.MYLIB.PLI(COINS) - 01.04 ----- COLUMNS 001 072
COMMAND INPUT ==> _          SCROLL ==> HALF
***** ***** TOP OF DATA *****
000100 COINS:
000200 PROCEDURE OPTIONS (MAIN);
000300 DECLARE
000400     COUNT    FIXED BINARY (31) AUTOMATIC INIT (1),
000500     HALVES   FIXED BINARY (31),
000600     QUARTERS FIXED BINARY (31),
000700     DIMES    FIXED BINARY (31),
000800     NICKELS  FIXED BINARY (31),
000900     SYSPRINT FILE STREAM OUTPUT PRINT;
001000     DO HALVES = 100 TO 0 BY -50;
001100         DO QUARTERS = (100 - HALVES) TO 0 BY -25;
001200             DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;
001300                 NICKELS = 100 - HALVES - QUARTERS - DIMES;
001400                     PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);
001500                         COUNT = COUNT + 1;
001600                             END;
001700                                 END;
001800                                     END;
001900                                         END COINS;
***** ***** BOTTOM OF DATA *****

```

Figure 25. Edit - Data Display

BASIC EDIT OPERATIONS

Under edit, four-way scrolling is available via the Scroll PF keys. The 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. Several lines may be modified before pressing the ENTER key.

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. Frequently used line commands include:

- I - Insert new line.
- D - Delete line.
- R - Repeat (duplicate) line.
- C - Copy line.
- M - Move line.

- A (after) - Specify destination of copy or move.
- B (before) - Specify destination of copy or move.
- (or < - Shift line left.
-) or > - Shift line right.

Single character line commands operate on individual lines (e.g., D to delete a line). Double character line commands operate on blocks of lines (e.g., DD on two different lines to indicate the first and last lines to be deleted). In most cases, a number may follow the line command to indicate multiple occurrences (e.g., I3 to insert three lines). 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 on line 2. Frequently used primary commands include:

- LOCATE Scroll to the specified line number.
- RENUM Renumber all lines.
- UNNUM Unnumber all lines (blank out sequence numbers).
- RESET Reset excluded lines and messages.
- SUBMIT Submit edit data to the job stream.
- SAVE Save the data.
- CANCEL Cancel edit without saving.
- FIND Find a specified character string.
- CHANGE Change a specified character string to another string.
- COPY Copy data from another member or data set.
- MOVE Copy data and then delete the member or data set.
- CREATE Create another member or data set from the edit data.
- REPLACE Replace another member or data set from the edit data.

Additional edit commands are described briefly in the following sections.

EDIT MODES AND PROFILES

Edit is capable of operating in the following modes:

- **NUMBER Mode** - controls validation and generation of sequence numbers.
- **AUTONUM Mode** - controls automatic renumbering when the data is saved.
- **STATS Mode** - controls generation of SPF library statistics.
- **PRINT Mode** - controls automatic recording of source listings in the SPF list file.
- **CAPS Mode** - controls translation of alphabetic characters to uppercase.
- **NULLS Mode** - controls replacement of trailing blanks with null characters on the display screen.
- **TABS Mode** - controls use of "hardware" or "logical" tabs.
- **HEX Mode** - controls display of data in hexadecimal format.
- **RECOVERY Mode** - controls automatic recording of edit transactions to facilitate recovery following a system failure.

Each mode may be turned on or off independently of the other modes by means of the following primary commands: NUMBER, AUTONUM, STATS, PRINT, CAPS, NULLS, TABS, HEX, and RECOVERY.

The current settings of the modes, together with the current MASK, TABS, and BOUNDS lines, are maintained in an edit profile which may be displayed at any time via the PROFILE command. When edit is terminated, the current profile is automatically saved.

Up to 25 different edit profiles are maintained for each user. This allows different mode settings and different MASK, TABS, and BOUNDS lines to be remembered and used as the initial settings for different types of source data.

Each profile is normally associated with the data "type" (the SPF library type or last qualifier in the data set name). When the user edits COBOL data, for example, the default profile contains whatever modes were in effect the last time the user finished editing COBOL data. Additional profiles may be created and used by specifying a profile name on the edit entry panel, or as an operand on the PROFILE command.

SPECIAL FEATURES

Following is a brief discussion of some of the special features provided by the SPF editor.

- **Column Identification Line** - A special line may be displayed anywhere in the data area of the screen (via the COLS line command) to determine column positions. Format of the line is:

-----+-----1-----+-----2-----+-----3-----etc.

where "1" indicates column 10, "2" indicates column 20, etc.

- **Insert Mask** - Information may be pre-entered on inserted lines by defining a mask (via the MASK line command). The contents of the mask remain in effect until changed by the user. Initially, the mask contains all blanks. The mask is saved as part of the edit profile.

- **Excluded Lines** - Designated lines of data may be temporarily excluded from display (via the X or XX line command) to facilitate visual verification of program control structure. Excluded lines may also be used to limit the scope of the FIND and CHANGE commands. Excluded lines may be redisplayed via the S (show), F (first), and L (last) line commands. The excluded lines are removed from display, but not deleted from the data.
- **Tabs** - Three types of tabs may be defined (via the TABS line command): "software" tabs which control cursor repositioning when the ENTER key is pressed, "hardware" tabs which cause attribute bytes to be inserted at user-defined positions on each line of the screen, or "logical" tabs which cause SPF to reposition data based on a user-defined special tab character. The tab definitions remain in effect until changed by the user, and are saved as part of the edit profile.
- **Bounds** - The column boundaries, which delimit the extent of the shift line commands and the FIND and CHANGE primary commands, may be changed at any time (via the BOUNDS line command). Normally, these bounds encompass the "statement body" of each data record, excluding the sequence number field. The bounds are saved as part of the edit profile.
- **Null Characters** - Trailing blanks in each data field may be displayed as null characters (via the NULLS primary command) to facilitate use of the 3270 INSERT key. Normally, all blanks are displayed as blanks.
- **Hexadecimal Display** - Data may be displayed and updated using hexadecimal representation. The format may be "data" (two character positions per byte) or "vertical" (one character position, but two lines per byte). The user may change characters by overtyping the hex representation and by issuing CHANGE commands with hex syntax.
- **Document Preparation Support** - Document preparation is aided by commands which facilitate: the bulk entry of text without regard for the end of the line, the insertion of new material into existing text, and the "reflowing" of text.
- **Edit Recovery** - Under user option, the SPF editor will automatically maintain a history of change activity. If there is a system failure, the user can recover the edit session up to the point of failure. A special panel is displayed when edit is reentered to indicate that automatic recovery is in effect.

EDIT TERMINATION

Normally, edit is terminated by pressing the End PF key, which causes the data to be saved if any changes have been made. The user may save the data without terminating edit via the SAVE command. The user may also terminate editing without saving via the CANCEL command.

Note: The Return PF key, which is logically equivalent to repeated use of the End PF key, will also cause the data to be saved if any changes have occurred.

UTILITIES (OPTION 3)

The utility option provides a variety of utility functions for: library, data set, and catalog maintenance; moving and copying data; resetting SPF library statistics; initiating hardcopy output; displaying or printing VTOC entries for a DASD volume; browsing and printing held SYSOUT data; and formatting SCRIPT/VS documentation.

The utility selection menu is shown in Figure 26.

```
----- UTILITY SELECTION MENU -----  
SELECT OPTION ==> _  
  
  1 LIBRARY  - LIBRARY UTILITY:  
                PRINT INDEX LISTING OR ENTIRE DATASET  
                PRINT, RENAME, DELETE, OR BROWSE MEMBERS  
                COMPRESS DATASET  
  2 DATASET  - DATASET UTILITY:  
                DISPLAY DATASET INFORMATION  
                ALLOCATE, RENAME, OR DELETE ENTIRE DATASET  
                CATALOG OR UNCATALOG DATASET  
  3 MOVE/COPY - MOVE OR COPY MEMBERS OR DATASETS  
  4 CATALOG  - CATALOG MANAGEMENT:  
                DISPLAY OR PRINT CATALOG ENTRIES  
                INITIALIZE OR DELETE USER CATALOG ALIAS  
  5 RESET    - RESET STATISTICS FOR MEMBERS OF SPF 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 SCRIPT/VS - FORMAT, DISPLAY, AND OPTIONALLY PRINT SCRIPT TEXT
```

Figure 26. Utility Selection Menu

For each of the utility options, a panel is displayed which allows the user to select a function and enter the appropriate library or data set information. These panels allow both option selection and data entry in a single panel format. An example is shown in Figure 27.

```

----- LIBRARY UTILITY -----
SELECT OPTION ==> _

C - COMPRESS DATASET                P - PRINT MEMBER
X - PRINT INDEX LISTING             R - RENAME MEMBER
L - PRINT ENTIRE DATASET           D - DELETE MEMBER
BLANK - DISPLAY MEMBER LIST        B - BROWSE MEMBER

SPF LIBRARY:
PROJECT ==> SPFDEMO
LIBRARY ==> XXX
TYPE   ==> COBOL
MEMBER ==>                (IF OPTION "P", "R", "D", OR "B" SELECTED)
NEWNAME ==>                (IF OPTION "R" SELECTED)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==>
VOLUME SERIAL ==>          (IF NOT CATALOGED)

DATASET PASSWORD ==>       (IF PASSWORD PROTECTED)

```

Figure 27. Library Utility Panel

LIBRARY UTILITY

The library utility options are shown in Figure 27.

This utility is intended primarily for maintenance of SPF libraries and other partitioned data sets, but options X (index) and L (list) also apply to sequential data sets.

Option C (compress) is operable only if the installation has installed a special procedure to allow the IEBCOPY (which requires authorization) to run in the foreground under SPF.

The three print options (X, L, and P) cause information to be recorded in the SPF list file. An index listing includes general information about the data set followed by a member list. For a sequential data set, the index listing includes general information only.

Option B (browse) provides a direct interface to browse. All browse commands can be executed. When browse is terminated by pressing the End PF key, the user is returned to this utility.

The default option (selected by leaving the option field blank), allows the user to select members for printing, renaming, deleting, or browsing by entering "P", "R", "D", or "B" in front of one or more member names on the member list. For renaming, a new member name must also be entered in the field immediately following the current member name.

DATA SET UTILITY

The data set utility provides the following options:

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

Any DASD-resident sequential or partitioned data set may be specified for option A (allocate). Any DASD-resident, non-VSAM data set may be specified for the other options.

Option A allows specification of volume serial (may specify a real DASD volume or an IBM 3850 virtual volume), data set characteristics, and space allocation parameters. Default values are pre-entered based on the last allocation or the last "display data set information" request (whichever occurred most recently).

Option D (delete) displays a confirmation panel to insure that the user did not select this option by mistake. As directed in the panel, the user must press either the ENTER key to confirm or the End PF key to cancel.

The default option (blank) causes a display of volume serial, data set characteristics, current space allocation, and current space utilization for the selected data set. This information may be used to precondition the defaults for option A (allocate), so that an existing data set may be used as the model for allocating a new data set.

MOVE/COPY UTILITY

The move/copy utility provides the following options:

- CP - Copy data set or member(s) and print
- MP - Move data set or member(s) and print
- C - Copy without print
- M - Move without print

The first panel displayed allows the user to select one of the options listed above, and specify the "from" library or data set. When the ENTER key is pressed, a second panel is displayed which allows the user to specify the "to" library or data set. The following options must also be specified on the second panel:

- Replace like-named members (YES or NO), if the "to" data set is partitioned.
- "To" data set disposition (OLD or MOD), if the "to" data set is sequential.

Both data sets must already exist. The utility will not automatically allocate space for a new "to" data set.

If a move option is selected (M or MP) and the "from" data set is partitioned, the selected members will be deleted in the "from" data set after they have been successfully copied. Move from a sequential data set causes the entire "from" data set to be deleted after its contents have been successfully copied to the "to" data set. If a copy option is selected (C or CP), no deletions will occur.

If a print option is selected (CP or MP), source listings will be recorded in the SPF list file.

If a member listing is requested, the user may select members to be moved or copied by entering an "S" in front of one or more member names. If the "to" data set is partitioned, the members may also be renamed by entering a new member name in the field immediately following the current name.

CATALOG MANAGEMENT UTILITY

The catalog management utility provides the following options:

- I - Initialize user catalog alias
- D - Delete catalog alias
- P - Print catalog entries
- blank - Display catalog entries

Option I (initialize) is valid only if there are no data sets currently allocated with the specified project name (or other high-level data set qualifier). It causes all subsequent catalog entries for this project name to be placed in the designated user catalog, pointed to by a single entry in the master catalog.

Option D (delete) causes the alias for the designated project name to be deleted from the master catalog.

Option P (print) causes a catalog listing for the designated project name to be recorded in the SPF file. The default option (blank) causes the catalog listing to be displayed via direct entry to browse.

RESET SPF STATISTICS UTILITY

The reset SPF statistics utility provides the following options:

- R - Create/update statistics, conditionally reset sequence numbers
- N - Create/update statistics, no reset of sequence numbers
- D - Delete SPF statistics

Option R is used to create statistics in an SPF library or other partitioned data set which does not currently have them, or to update existing statistics. The data is scanned to determine if sequence numbers are present in all records. If so, the data is renumbered and the modification flags (the last two digits of each sequence number) are set to zeros. If sequence numbers are not present, renumbering is not done.

Option N is also used to create or update statistics; however, the data is not renumbered. This option should be used if the data already contains SPF statistics and the user wants to update the user id and/or version information without renumbering the data.

Option D is used to delete SPF statistics.

If a version number is specified, the statistics are created or reset as follows:

- | | |
|---------------------------|------------------------------------|
| Version number: | set to the specified value |
| Modification level: | set to zero |
| Creation date: | set to current date |
| Date/time last modified: | set to current date and time |
| Current number of lines: | set to the current number of lines |
| Initial number lines: | set to the current number of lines |
| Number of modified lines: | set to zero |

If member listing is requested, the user may select members to be reset by entering an "S" in front of one or more member names.

HARDCOPY UTILITY

The hardcopy utility is used to initiate printing or punching of data which is already in the proper format for output. This utility provides the following options:

- J - Generate JCL to print or punch data set
- L - Route data set to local printer

Any DASD-resident data set may be specified except for the SPF list and log files. Since list and log remain open throughout the session, they may only be printed upon SPF termination.

For sequential data sets, the user may specify whether the data set is to be kept or deleted after printing. (Partitioned data set members are always kept.)

For option J, a SYSOUT class and job statement parameters must be specified. The SYSOUT class may specify a printer or punch.

For option L, a printer id must be specified. When the user enters the desired information and presses the ENTER key, SPF invokes the "DSPRINT" TSO command processor to print the data set on the specified local printer.

DSPRINT is an IBM Field Developed Program that must be installed to use the "L" option.

LIST VTOC UTILITY

The list volume table of contents (VTOC) utility provides the following options:

- P - Print VTOC entries
- blank - Display VTOC entries

Option P (print) causes a VTOC listing for the designated volume to be recorded in the SPF file.

The default option (blank) causes the VTOC listing to be displayed at the terminal via direct entry to browse.

For either the print or display options, the VTOC listing may be limited to data sets which begin with a designated high-level qualifier (specified via project name on the list VTOC panel).

OUTLIST UTILITY

The outlist utility provides access to job output which is in a held SYSOUT queue. This utility has the following options:

- L - List job names via the TSO STATUS command
- D - Delete job output
- P - Print job output
- R - Requeue job output to a new output class
- blank - Display job output

Option L displays a list of job names and job id's for currently held jobs. If no job name is supplied, or if the job name is the user's id plus one character, status is displayed for all jobs which begin with the user's id. If any other job name is supplied, status for that exact job is displayed.

Option D deletes the held output for a specific job from the specified SYSOUT queue.

Option P removes the held output for a specific job from the SYSOUT queue for printing. After the ENTER key is pressed, a panel is displayed on which the user enters information about how and where the job output is to be printed.

Option R requeues the held output for a specific job to another SYSOUT class.

The default option (blank) displays held output for the specified job via direct entry to browse. The data remains in the SYSOUT queue. When the user presses the End PF key to terminate browse, the outlist utility panel is redisplayed and the user may then choose to print, requeue, or delete the job output.

SCRIPT/VS UTILITY

The SCRIPT/VS utility displays a sequence of panels to allow formatting, displaying, and printing of text maintained in SPF libraries or other partitioned or sequential data sets. Use of this utility requires installation of the Document Composition Facility (SCRIPT/VS), Program Product 5748-XX9, with the Foreground Environment Feature.

This utility has the following options:

- 1 - Draft document options
- 2 - Final document options

For the draft document options, it is assumed that the primary interest of the user is to assure that the content, syntax, and spelling within the document are correct. The final document options provide additional flexibility and control for the user interested in specific document formats.

The draft document options allow the user to specify:

Spelling Check (Yes or No)	Other SCRIPT Parms
Uppercase Only (Yes or No)	SCRIPT Profile
Line Numbering (Yes or No)	SCRIPT Macro Library

The final document options allow the user to specify:

Spelling Check (Yes or No)	3800 Type Fonts
Uppercase Only (Yes or No)	Binding (Margins)
Device Type	Other SCRIPT Parms
Lines Per Inch	SCRIPT Profile
Form Width	SCRIPT Macro Library
Form Length	

The first two panels displayed by this utility allow the user to select one of the options and enter the appropriate processing parameters. The SCRIPT/VS formatter is then invoked. When it completes, the formatted document is displayed via direct entry to browse.

When the user terminates browse by pressing the End PF key, a print panel is displayed to allow the formatted output to be printed, kept, or deleted.

FOREGROUND (OPTION 4)

The foreground option provides an interface to standard language processors for foreground compilation, assembly, link edit, or debugging of programs stored in SPF libraries. For foreground processing of other partitioned or sequential data sets, the appropriate TSO prompter commands may be entered directly via the command option (primary option 6).

The foreground selection menu is shown in Figure 28.

```
----- FOREGROUND SELECTION MENU -----  
SELECT OPTION ==> _  
  
1 - SYSTEM ASSEMBLER  
2 - OS/VS COBOL COMPILER  
3 - FORTRAN IV (G1) COMPILER  
4 - PL/I CHECKOUT COMPILER  
5 - PL/I OPTIMIZING COMPILER  
6 - LINKAGE EDITOR  
7 - COBOL INTERACTIVE DEBUG  
8 - FORTRAN INTERACTIVE DEBUG  
9 - PASCAL/VS COMPILER
```

Figure 28. Foreground Selection Menu

When a language processor has been selected, an entry panel is displayed to allow the user to enter the appropriate library and member names, library concatenation sequence, and processor options. Figure 29 shows an example for Assembler. The project name, first library name, type qualifier, member name, and list id must be specified. The other parameters are optional.

From the entry panel, the user may press either:

- The ENTER key, which causes the foreground processor to be invoked without entering Session Manager mode, or
- Any one of the SPF Scroll keys, which causes foreground execution under control of the Session Manager (if the Session Manager, Program Product 5740-XE2, is installed).

If the Session Manager is not invoked, the terminal is operated in normal TSO fashion. Any communication with the foreground processor is in line-I/O mode. Upon completion of foreground processing, three asterisks (***) are displayed. The user may then press ENTER to redisplay the entry panel.

Under the Session Manager, the terminal is operated according to Session Manager procedures. Upon completion of foreground processing the user is prompted to enter a null line to return to SPF control and redisplay the entry panel.

```
----- FOREGROUND ASSEMBLY -----
ENTER/VERIFY PARAMETERS BELOW:

PROJECT ==> SPFDEMO
LIBRARY ==> XXX      ==> A      ==> MASTER  ==>
TYPE    ==> ASM
MEMBER  ==> TOP

LIST ID ==> LISTASM_      PASSWORD ==>

ASSEMBLER OPTIONS:
    ==> LIST,TEST,TERM,RENT

CAUTION: DO NOT SPECIFY OPTIONS LIB, LOAD, OR PRINT. THEY ARE GENERATED
AUTOMATICALLY BY SPF.
```

Figure 29. Foreground - Assembler Example

The Assembler entry panel shown in Figure 29 is typical of the language processor panels. In the figure, a concatenation sequence of three SPF libraries has been specified:

```
SPFDEMO.XXX.ASM
SPFDEMO.A.ASM
SPFDEMO.MASTER.ASM
```

Before the foreground processor is invoked, the concatenated sequence of libraries is scanned to find the member to be assembled or compiled (member TOP in this example). If, in this example, member TOP first appears in data set SPFDEMO.A.ASM, the following command would be generated.

```
ASM 'SPFDEMO.A.ASM(TOP)'
    LIB('SYS1.MACLIB',
        'SPFDEMO.XXX.ASM',
        'SPFDEMO.A.ASM',
        'SPFDEMO.MASTER.ASM')
    LOAD('SPFDEMO.XXX.OBJ(TOP)')
    PRINT(LISTASM)
    LIST,TEST,TERM,RENT
```

The commands generated for the other language processors are similar to the Assembler example, except that:

- The macro library 'SYS1.MACLIB' is not included in the concatenation sequence passed via the LIB parameter.
- For COBOL, an optional TEST parameter may be generated, as follows:

```
TEST ('project-name.lib1-name.SYM(member-name)').
```

The user may control the generation of this parameter via a field on the panel.

- For FORTRAN, there is no LIB parameter. The concatenation sequence is still used to locate the member to be compiled.
- For the two PL/I compilers and PASCAL, an OBJECT parameter (rather than LOAD) is generated to designate the destination of the object module.
- For the PL/I Checkout compiler, the OBJECT parameter is optional (controlled by a field on the panel). If the user specifies OBJECT, the following PLIC parameter is generated:

```
OBJECT('project-name.lib1-name.OBJ(member-name)'  
      'project-name.lib1-name.ITEXT')
```

If the user specifies NOOBJECT, neither OBJ nor ITEXT data is generated.

In addition, the foreground option automatically generates the following optional parameters:

For ASM:	LIB, LOAD, PRINT
For COBOL:	LIB, LOAD, PRINT, TEST
For FORT:	LOAD, PRINT
For PLIC:	LIB, OBJECT, PRINT
For PLI:	LIB, OBJECT, PRINT
For LINK:	LIB, LOAD, PRINT
For TESTCOB:	LOAD, PRINT, PARM
For TESTFORT:	LIB, PRINT, SOURCE
For PASCAL:	LIB, OBJECT, PRINT

For the linkage editor, the entry panel allows specification of up to two system subroutine libraries. The concatenation sequence used to locate the member for input to the linkage editor is:

```
project-name.lib1-name.type  
project-name.lib2-name.type  
(etc.)
```

where "type" is whatever the user specifies on the panel (may be OBJ or may be some other type containing linkage editor control statements). If the type is not OBJ, an "OBJECT" DDNAME is automatically allocated to facilitate use of the following linkage editor control statements.

```
INCLUDE OBJECT(member-name)
```

The concatenation sequence passed to the linkage editor via the LIB parameter has a type qualifier of LOAD and includes the user-specified system libraries, as follows:

```
LIB('project-name.lib1-name.LOAD',  
    'project-name.lib2-name.LOAD',  
    ...  
    'syslib1-name',  
    'syslib2-name')
```

This concatenation sequence is used by the linkage editor to resolve "automatic call" references.

BACKGROUND (OPTION 5)

The background option provides an interface to standard language processors for background compilation, assembly, or link edit of programs stored in SPF libraries. For other background jobs, the SUBMIT command may be entered via the command option (primary option 6) or via the SUBMIT command under edit.

The background selection menu is shown in Figure 30. It allows the user to enter job statement information, as well as select a language processor.

As with other secondary menus, it is possible to bypass the background selection menu by entering two numbers, separated by a decimal point, on the primary option menu. Note, however, that use of this procedure does not allow verification or changes to the job statement parameters, nor does it allow the generation of multiple compilations and/or link edits within the same job.

```
----- BACKGROUND SELECTION MENU -----
SELECT OPTION ==> _

      1 - SYSTEM ASSEMBLER                4 - PL/I CHECKOUT COMPILER
      2 - OS/VS COBOL COMPILER            5 - PL/I OPTIMIZING COMPILER
      3 - FORTRAN IV (61) COMPILER        6 - LINKAGE EDITOR
                                          9 - PASCAL/VS COMPILER

JOB STATEMENT INFORMATION: (VERIFY BEFORE PROCEEDING)
==> //SILVA7  JOB (U602,B043),'SILVA RJ',CLASS=B,
==> //          MSGLEVEL=2,
==> //          NOTIFY=SILVA
==>
```

Figure 30. Background Selection Menu

The data entry panels for background processing are similar to those for foreground, except that the user may enter either a list id (if the output listing is to be stored in a data set) or SYSOUT class (if the listing is to be printed as part of the background job). If both list id and SYSOUT class are specified, list id is used.

When the user has filled in the entry panel and pressed the ENTER key, the appropriate JCL statements are generated. The background selection menu is then redisplayed (unless that menu was bypassed) with a "JCL GENERATED" message displayed in the short message area (line 1). If the background selection menu was bypassed, the generated JCL is submitted and the user is returned to the primary option menu.

When the background menu is redisplayed, the job statement parameters are shown for information only -- they are no longer intensified and may not be overtyped, since the JOB statement has already been generated. At this point, the user may:

- Select another processor to cause more JCL (additional job steps) to be generated,
- Enter CANCEL to return to the primary option menu without submitting the job, or
- Press the End (or Return) PF key to cause the generated JCL to be submitted for execution.

COMMAND (OPTION 6)

The command option allows direct entry of a TSO command or command procedure (CLIST) under SPF. When this option is selected, a blank panel is displayed with the following prompt:

```
ENTER TSO COMMAND OR CLIST BELOW:  
===>
```

The command or CLIST is entered to the right of the arrow, using standard TSO syntax. It may be continued on the next two lines, if necessary.

The following commands may not be entered under SPF:

- TEST
- LOGON, LOGOFF
- SPF
- Authorized commands (e.g., RACF commands)
- CALL to an authorized program

After the desired command has been typed, the user may press either:

- The ENTER key, which causes the command to be executed without entering Session Manager mode, or
- Any one of the SPF Scroll keys, which causes the command to be executed under control of the Session Manager (if the Session Manager, Program Product 5740-XE2, is installed).

If the Session Manager is not invoked, the terminal is operated in normal TSO fashion. Any communication with the command is in line-I/O mode. Upon completion of the command, three asterisks (***) are displayed. The user may then press ENTER to redisplay the command panel.

Under the Session Manager, the terminal is operated according to Session Manager procedures. Upon completion of command processing, the user is prompted to enter a null line to return to SPF control and redisplay the command panel.

When the command panel is redisplayed, the command which was just executed is pre-entered to the right of the arrow. The user may then enter another command, or press the End PF key to return to the primary option menu.

SUPPORT (OPTION 7)

The support option provides test aids and conversion utilities for the development of applications to be run under the SPF dialog manager. The support selection menu is shown in Figure 31.

```
----- SUPPORT SELECTION MENU -----  
SELECT OPTION ==> _  
  
1 TEST PANEL - DISPLAY PANEL AS USER WOULD SEE IT  
2 TEST FUNCTION - INVOKE DIALOG FUNCTION OR SELECTION MENU  
3 TEST VARIABLES - SET OR DISPLAY VARIABLES FOR TEST FUNCTION  
4 CONVERT MENUS - CONVERT SELECTION/TUTORIAL MENUS TO NEW FORMAT  
5 CONVERT MSGS - CONVERT MESSAGES TO NEW FORMAT
```

Figure 31. Support Selection Menu

The test panel option allows a panel definition to be tested by displaying it as the end user would see it. The name of the panel and, optionally, a message id and initial cursor location may be specified. These are the same parameters which may be specified (from the dialog function) when invoking the display service.

The test function option allows a dialog function to be tested without having to build "scaffolding" code. The name of the function and parameters that may be passed correspond exactly to what may be specified when the function is invoked from a selection menu or from another function.

The test variables option allows dialog variables to be set and/or displayed. It is intended for use with the test panel and test function options.

The convert menus option provides automated conversion from old format SPF selection menus and tutorial pages to new format panel definitions. This option allows specification of the "from" and "to" libraries in a manner similar to the move/copy utility (option 3.3).

The convert messages option provides automated conversion from old format SPF message definitions to new format message definitions. It works the same as the convert menus option.

TUTORIAL (OPTION T)

The tutorial option provides immediate online reference and instruction on how to use the SPF program development facility. It may be invoked from the primary option menu or via the Help PF key. The tutorial may be viewed sequentially from beginning to end, or it may be viewed randomly by selecting specific topics from an alphabetic index or table of contents.

When the tutorial is invoked from the primary option menu, introductory pages are displayed to explain how the tutorial works (Figure 32). Following the introduction, a table of contents is displayed from which the user may select a topic by entering the desired section number.

When the tutorial is invoked via the Help PF key, the appropriate section of the tutorial is entered based on what the user was doing when the Help key was pressed.

When viewing the tutorial, the user may select topics by number, or simply press the ENTER key to view the next topic. On any panel, the user may also enter the following commands:

BACK or B - to back up to the previously viewed page
SKIP or S - to skip to the next topic
UP or U - to display a higher level list of topics
TOP or T - to display the table of contents
INDEX or I - to display the tutorial index.

During execution of the tutorial, the four scroll PF keys are interpreted as follows:

UP - display higher level list of topics
DOWN (skip) - skip to the next topic
LEFT (back) - display previous tutorial page
RIGHT (next) - display next tutorial page

The ENTER key also means "next."

The Help PF key may be pressed at any time to display a one-page summary of how to use the tutorial.

The tutorial function is terminated by pressing the End PF key. This causes a return either to the primary option menu or to the display from which the user requested help.

```

----- SPF TUTORIAL -----
                                STRUCTURED PROGRAMMING FACILITY
                                TUTORIAL

THIS TUTORIAL PROVIDES ON-LINE INFORMATION ABOUT THE FEATURES AND OPERATION
OF SPF. YOU MAY VIEW THE TUTORIAL SEQUENTIALLY, OR YOU MAY CHOOSE SELECTED
TOPICS FROM LISTS THAT ARE DISPLAYED ON MANY OF THE TUTORIAL PAGES.

THE TABLE OF CONTENTS CONTAINS A LIST OF MAJOR TOPICS. SUBSEQUENT PAGES
CONTAIN ADDITIONAL LISTS THAT LEAD YOU TO MORE SPECIFIC LEVELS OF DETAIL.
TOPICS MAY ALSO BE SELECTED FROM THE TUTORIAL INDEX.

THE NEXT TWO PAGES CONTAIN A DESCRIPTION OF HOW TO USE THIS TUTORIAL.

PRESS ENTER TO PROCEED TO THE NEXT PAGE, OR
PRESS UP KEY TO GO DIRECTLY TO THE TABLE OF CONTENTS, OR
PRESS END KEY TO RETURN TO THE PRIMARY OPTION MENU.

```

```

TUTORIAL ----- INTRODUCTION ----- TUTORIAL
NEXT SELECTION ==> _

YOU MAY VIEW THE TUTORIAL SEQUENTIALLY BY SIMPLY LEAVING THE "NEXT SELECTION"
FIELD BLANK AND PRESSING THE ENTER KEY. ALTERNATIVELY, YOU MAY SELECT TOPICS
FROM LISTS THAT ARE DISPLAYED ON MANY OF THE TUTORIAL PAGES. FOR EXAMPLE,
ENTER:  NEXT SELECTION ==> 3  TO SELECT TOPIC 3.

YOU MAY ALSO ENTER ONE OF THE FOLLOWING CODES IN THE "NEXT SELECTION" FIELD
ON ANY TUTORIAL PAGE:
    BACK OR B - TO BACK UP TO THE PREVIOUSLY VIEWED PAGE.
    SKIP OR S - TO SKIP THE CURRENT TOPIC AND GO ON TO THE NEXT TOPIC.
    UP    OR U - TO DISPLAY A HIGHER LEVEL LIST OF TOPICS.
    TOP   OR T - TO DISPLAY THE TABLE OF CONTENTS.
    INDEX OR I - TO DISPLAY THE TUTORIAL INDEX.

                (CONTINUED ON NEXT PAGE)

```

Figure 32. Tutorial - First Two Pages

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

SPF operates as a TSO command processor under the Time Sharing Option of VS2 Release 3.8 (MVS). SPF is written in PL/S and translated into Assembler Language. The BPAM and BSAM access methods are required by SPF for reading and writing data sets, and TSO/TCAM or TSO/VTAM are required for terminal communication.

One of the following teleprocessing access methods is required:

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

The SPF program development facility provides interfaces to the following IBM processing programs for foreground and background execution:

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

The appropriate processing programs and TSO prompters must be installed to use the foreground and background options.

An interface to SCRIPT/VS is also provided to allow formatting, display, and printing of text maintained in SPF libraries or other data sets. Use of this feature requires installation of the following IBM program product:

Document Composition Facility (SCRIPT/VS) with the Foreground Environment Feature	5748-XX9
--	----------

Internal interfaces are provided to the following IBM programs. These programs are not required to operate SPF. However, if an IBM 3284, 3286, 3287, 3288, or 3289 printer is used for SPF output, the appropriate DSPRINT command processor must be installed on the system.

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

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 MVS with the Time Sharing Option (TSO).

Approximately 512K bytes of the pageable link pack area (PLPA) are required for SPF program occupancy. The virtual storage requirements for each user's region will vary depending upon the application being executed, the size of tables and data sets, and the use of split screen mode. The minimum region size is 512K bytes, but a 2048K byte region is recommended. An even larger region may be required under certain circumstances.

Approximately 20 cylinders of IBM 3330 disk storage (or an equivalent amount of space for other types of devices) are required for the program, panel, message, and skeleton libraries distributed with SPF.

TERMINAL REQUIREMENTS

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

- 3275 Models 2 and 12
- 3276 Models 2, 3, 4, 12, 13, and 14
- 3277 Model 2 (local or remote attachment)
- 3278 Models 2, 3, 4, and 5 (local or remote attachment)
- 3279 All models (local or remote attachment), when operated in 4-color compatibility mode.

Each display station must be equipped with one of the following keyboards:

For 3275 or 3277 Display Stations:

- 78 Key Operator Console (feature 4632)
- 78 Key EBCDIC Typewriter (feature 4633)
- 78 Key ASCII Typewriter (feature 4635)
- 78 Key EBCDIC Typewriter/APL (feature 4638), when operated with APL switch off.

For 3276, 3278, or 3279 Display Stations:

- 75 Key EBCDIC Typewriter (feature 4621)
- 75 Key ASCII Typewriter (feature 4624)
- 87 Key EBCDIC Typewriter (feature 4627)
- 87 Key ASCII Typewriter (feature 4628)
- 87 Key EBCDIC Typewriter/APL (feature 4626), when operated with APL switch off.
- 87 Key EBCDIC Typewriter/Text (feature 4629), when operated with TEXT switch off.

The standard character set (94 graphics plus blank and null) is supported on 3276, 3278, and 3279 Display Stations.

The following are supported, but 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 is strongly recommended to enhance usability. The alarm is sounded whenever a warning or error message is displayed.

Use of the IBM 3284, 3286, 3287, 3288, and 3289 printers requires installation of the appropriate DSPRINT command processor.

CUSTOMER RESPONSIBILITIES

The installation of SPF requires a properly configured IBM System/370, appropriate terminals and other required devices, and MVS with the Time Sharing Option (TSO). Also, appropriate processing programs and TSO prompters must be installed to use the SPF foreground and background options, and the Document Composition Facility must be installed to use the SCRIPT/VS utility.

To install SPF, operations personnel must be knowledgeable in OS/VS JCL, TSO, and the System Modification Program (SMP). Installation procedures are described in the SPF Installation and Customization Guide.

If SPF 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 SPF 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 MVS, TSO, and appropriate programming languages, and should review the SPF Dialog Development Guide. A user of the SPF Program Development Facility must be familiar with MVS and TSO, and should review the SPF Program Reference Manual.

Special SPF training or courses are not required for programming personnel familiar with the IBM 3270, the programming and machine systems, and the language processors in use.

CONVERSION OF MENUS AND MESSAGES

Installations which have previously extended or custom tailored SPF may need to convert old format selection menus, tutorials, and messages to the new panel and message formats. Conversion utilities have been provided for this purpose in the program development facility (new option 7 on the primary option menu).

All selection menus (displayed by program SPFUTIL in the previous SPF products) must be converted to new format to run under the SPF dialog manager. Conversion of tutorial pages is optional, but recommended, since support for old format tutorial pages may be dropped in the future. No attempt should be made to convert foreground and background entry panels; these are supported in old format only.

RELATED DOCUMENTS

The following documents are planned for availability when SPF is released.

- SPF Dialog Development Guide - Provides detailed information on how to develop interactive applications to run under SPF.
- SPF Program Reference Manual - Provides detailed information on how to use the SPF program development facility.
- SPF Quick Reference Summary (Card) - Provides a summary of SPF program development facility options and command formats.
- SPF Installation and Customization Guide - Provides detailed information on how to install and custom tailor SPF.
- SPF Program Logic Manual - Describes SPF internal logic, program structure, and data areas.
- SPF Program Listings (Microfiche) - Provides compiler and assembler listings of SPF programs.

Two sets of documents will be available: One for SPF with MVS Time Sharing Option (SPF-MVS), and the other for SPF with VM/370 Conversational Monitor System (SPF-VM).

Two PF key overlays are also planned for availability: One for use with IBM 3275 or 3277 terminals, and the other for use with IBM 3276, 3278, or 3279 terminals. The overlays are the same for SPF-MVS and SPF-VM.

CHAPTER 5. SYSTEM PRODUCTIVITY FACILITY: SUMMARY

Program Name: System Productivity Facility

Program Number: 5668-009

The System Productivity Facility is a program development tool designed to take advantage of the characteristics of IBM 3270 display terminals, and to increase programmer productivity in an interactive environment.

The System Productivity Facility replaces the Structured Programming Facility program products. It includes significant new functions which simplify the development of interactive applications.

New services are provided to define and control symbolic variables, display predefined screen images and messages, build and maintain permanent tables of user information, generate output files for job submission or other processing, interface to edit and browse, and log hardcopy output.

Testing Period: The testing period is one month.

License: A separate license is required for each designated machine on which the licensed program materials will be used, except as otherwise provided by IBM.

Program Services: Central Service and Local Service will be available until discontinued by IBM upon six month's written notice. Local Service will be provided by IBM Field Engineering for DPD and IBM Customer Engineering for E/ME/A and A/FE.

Warranty: The System Productivity Facility is warranted to conform to its Licensed Program Specifications when shipped to the customer if properly used in the specified operating environment.

Licensed Program Specifications may be updated from time to time, and such updates may constitute a change in specifications.

Following the discontinuance of all program services, this program will be distributed on an "As Is" basis without warranty of any kind either express or implied.

Availability: The estimated availability dates at PID (Program Information Department) are December 1980 for SPF-MVS, and April 1981 for SPF-VM.

APPENDIX A. SPF-MVS AND SPF-VM DIFFERENCES

The System Productivity Facility (SPF) supports two environments:

- MVS Time Sharing Option (SPF-MVS)
- VM/370 Conversational Monitor System (SPF-VM)

SPF provides equivalent function in the two environments. Interfaces to dialog services, display formats, and operation are fully compatible except for those features which are explicitly oriented to the particular environment. Following is a summary of the differences.

DIALOG MANAGER

- Under SPF-MVS a dialog function may be a program module or a CLIST. Under SPF-VM a dialog function may be a program module or an EXEC (coded in EXEC2 language). The CLIST and EXEC2 languages are native to the environment, and differ in syntax.
- Under SPF-MVS a module function must be link edited; it is invoked via LINK. Under SPF-VM a module function is not link edited; it is invoked via LOAD, and other modules which it references are automatically loaded with it.
- The edit and browse interface services under SPF-MVS require specification of a data set name and, optionally, a volume serial and OS password. Under SPF-VM these services require specification of a CMS file id and, optionally, a member name (for MACLIBs or TXTLIBs).
- In SPF-VM, the system variable ZPREFIX (user prefix) has the same value as ZUSER (user id).
- In SPF-VM, the "CONTROL DISPLAY SM" (Session Manager) service has the same meaning as "CONTROL DISPLAY LINE".
- Before entering SPF, certain libraries must be set up (panels, messages, skeletons, and tables). The procedures differ in the two environments. For SPF-MVS, concatenated partitioned data sets must be allocated to specified DDNAMEs. For SPF-VM, concatenated MACLIBs must be specified with FILEDEF commands.

PROGRAM DEVELOPMENT FACILITY

- On the primary option menu, option 6 is:
 SPF-MVS: Enter TSO Command or CLIST
 SPF-VM : Enter CMS Command or EXEC
- Panels and tutorials which refer to "data sets" in SPF-MVS refer to "files" in SPF-VM. The first part of an entry panel (browse or edit, for example) allows specification of an SPF library. The format is identical in SPF-MVS and SPF-VM:

```
SPF LIBRARY:
PROJECT ==>
LIBRARY ==>
TYPE    ==>
MEMBER  ==>
                (BLANK FOR MEMBER SELECTION LIST)
```

- The second part of an entry panel differs as follows. For SPF-MVS:

```
OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==>
VOLUME SERIAL ==>
                (IF NOT CATALOGED)
```

For SPF-VM:

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

- The third part of an entry panel contains an (OS) password field for SPF-MVS. For SPF-VM, this part of the panel may contain a read password field, an update password field, or both, depending on the type of function. In SPF-MVS, passwords pertain to data sets; in SPF-VM, they pertain to mini-disks.
- Implementation of SPF libraries is accomplished in SPF-MVS using partitioned data sets, while SPF-VM uses sequential CMS files and/or MACLIBs and TXTLIBs (under user option).
- SPF-MVS supports printing on either:
 System printer (via submission of a background job)
 328x printer (via interface to DSPRINT command)

SPF-VM supports printing via the CMS PRINT command. The output may be directed to the system printer or spooled to another user (or machine), which may be at a remote node in the network.
- Specification of "job statement information" occurs in SPF-MVS on the background selection menu and all panels pertaining to hardcopy output. SPF-VM provides job information on the batch selection menu only. All other output is accomplished with the CMS PRINT, PUNCH, or DISK DUMP commands, rather than via job submission.

Additionally, job statement information syntax and keywords are different, reflecting differences between CMS batch job streams and VS2 JCL.

- Differences in foreground processing (option 4) and background (batch) processing (option 5) are as follows:

SPF-MVS: Compiled/assembled output (object module) goes into an SPF library whose "type" is OBJ. Link edit output (load module) goes into an SPF library whose "type" is LOAD.

SPF-VM: Compiled/assembled output (object module) goes into an SPF library whose "type" is TEXT. Linkage editing is not supported, but the LOAD command is supported.

- Option 3.2 (dataset/file utility). The following suboptions of this utility in SPF-MVS:

A - Allocate new data set
C - Catalog data set
U - Uncatalog data set

are replaced in SPF-VM with:

S - Specify new SPF library
U - Unspecify SPF library

There is no need for an "allocate file" capability under SPF-VM, since space allocation is handled automatically by VM/CMS. There is, however, a need to pre-specify SPF library identifiers and file characteristics to SPF-VM.

- Option 3.4. In SPF-MVS this is the catalog management utility. In SPF-VM it is replaced with the project utility, which provides a similar function for SPF libraries.
- Option 3.6. In SPF-MVS this is the hardcopy utility. In SPF-VM it is replaced with the spool utility, which provides similar functions for printing and punching, and also allows spooling to another user (or machine), which may be at a remote node in the network.
- Option 3.7. In SPF-MVS this is the list VTOC utility, which has no equivalent in SPF-VM. It is replaced in SPF-VM with the reader utility.
- Option 3.8. In SPF-MVS this is the outlist utility, which has no equivalent in SPF-VM. It is replaced in SPF-VM with the retrieve utility.

INDEX

Allocate New Data Set 50
Attention PA Key 7, 30

Background
 Background Printing 29, 37, 52, 57
 Background Processing 57

Browse 41
Browse Interface Service 20
Browse Member 49

Catalog Data Set Name 50
Catalog Management Utility 51
Change
 Repeat Change PF Key 30

Compress Data Set 49
Concatenation of SPF Libraries 25, 26
Control Service 20
Convert Manus/Messages 60, 65
Copy Data Set or Member (Move/Copy Utility) 50
Cursor
 Move Cursor PF Key 8, 30

Customer Responsibilities 65

Data Set
 Characteristics 27, 28, 41, 43
 Security 28
 SPF List and Log Files 32

Data Set Utility 50

Delete
 Delete Catalog Alias 51
 Delete Data Set 50
 Delete Job Output 52
 Delete Member 49
 Delete SPF Statistics 51

Dialog 3
Dialog Functions 3
Dialog Manager 1, 3, 69
Dialog Services
 Browse Interface Service 20
 Control Service 20
 Display Service 11
 Edit Interface Service 20
 File Tailoring Services 18
 Select Service 4, 5
 Log Service 20
 Table Services 16
 Variable Services 9

Dialog Test Aids 60
Dialog Variables 9

Display
 Display Catalog Entries 51
 Display Data Set Information 50
 Display Document (SCRIPT/V5) 53
 Display Job Names/ID's 52
 Display Job Output 52
 Display Member Selection List 27, 38
 Display Variables (for Testing) 60
 Display VTOC Listing 52
 SPF Display Formats 22

Display Service 11
Display Table Service 17
Documents, Related 66

Edit 43
Edit Interface Service 20
Edit Modes and Profiles 46
End PF Key 7, 30

File Tailoring Services 18
 Find
 Repeat Find PF Key 30
 Foreground Processing 54
 Functions 3

 Hardcopy Utility 52
 Help
 Help PF Key 7, 30
 Help Information 34

 Initialize User Catalog 51
 Invocation of SPF 4, 36
 ISPF Command 4, 36

 Job
 Job Statement Information 29
 Job Submission for Printing 28, 37, 52, 57
 Job Submission for Processing 57

 Language Processors 54, 57, 63
 Libraries
 Library Statistics 35, 51
 SPF Library Definition 25
 Library Utility 49
 List File 35, 37
 List Job Names/ID's 52
 List VTOC Utility 52
 Log File 35, 37
 Log Service 20
 Logical Screen 7, 33

 Machine Requirements 64
 Member Selection List 27, 38
 Menus 2, 4
 Messages 3, 15
 Move Cursor PF Key 8, 30
 Move Data Set or Member (Move/Copy Utility) 50

 Outlist Utility 52

 Page Size
 For Listings 39
 For Scrolling 32
 Panels 2, 3, 11
 Passwords 28
 PCF 28, 63
 Printing
 Automatic Printing of Edited Data 46
 Print Catalog Entries 51
 Print Data Set or Member 49
 Print Document (SCRIPT/VS) 53
 Print Index Listing 49
 Print Job Output 52
 Print Option of Move/Copy 50
 Print PF Key 8, 30
 Print VTOC Listing 52
 Route to Local Printer 37, 52, 63, 64
 Submit Print Job to Background 29, 37, 52, 57

 Process
 Background Processing 57
 Foreground Processing 54
 Language Processors 54, 57, 63
 SCRIPT/VS Document Processing 53
 Program Access (PA) Keys 7, 30
 Program Development Facility 1, 21, 70
 Program Function (PF) Keys 7, 30
 Programming Libraries 25
 Programming Requirements 63
 Protection, Data Set 28

 RACF Passwords 28, 63
 Related Documents 66

Rename
 Rename Data Set 50
 Rename Member 49

Repeat
 Repeat Change PF Key 30
 Repeat Find PF Key 30

Requeue Job Output 52

Requirements
 Programming Requirements 63
 Machine Requirements 64
 Terminal Requirements 64

Reset SPF Statistics Utility 51

Reshow PA Key 7, 30

Return PF Key 7, 30

Route to Local Printer 37, 52, 63, 64

SCRIPT/VS Utility 53

Scrolling 32
 Scroll PF Keys 7, 30, 31

Security, Data Set 28

Select Service 4, 5

Services (see Dialog Services)

Session Manager 20, 54, 59, 63

Set Variables (for Testing) 60

Skeletons 3, 18

SPF Parameters 29, 39

Split Screen 7, 33
 Split PF Key 7, 30
 Swap PF Key 7, 30

Statistics, Library 35, 51

Structured Programming 21

Submit Background Job (see Job)

Summary 67

Support 60

Swap PF Key 7, 30

System Requirements (see Requirements)

Tables 3, 16

Table Services 16

Terminal Requirements 64

Termination of SPF 37

Test Dialog Function 60

Test Panel 60

TSO Command Entry 59

Tutorial 6, 61

Uncatalog Data Set Name 50

User Parameters
 Saving of Parameters 29
 Specifying SPF Parameters 39

Utilities
 Catalog Management 51
 Data Set 50
 Hardcopy 52
 Library 49
 List VTOC 52
 Move/Copy 50
 Outlist 52
 Reset 51
 SCRIPT/VS 53

Variables 9

Variable Services 9

System Productivity
Facility for MVS
General Information

GC34-2039-0

READER'S
COMMENT
FORM

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

System Productivity Facility for MVS: General Information (File No. S370/4300-39)

Printed in U.S.A.

GC34-2039-0

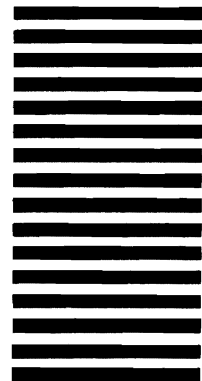
Fold and tape

Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



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

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department Z59, Building 931
P. O. Box 390
Poughkeepsie, New York 12602

Fold and tape

Please Do Not Staple

Fold and tape



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601