CONCEPT DYNAMICS SYSTEM
SOFTWARE DEVELOPMENT
ENVIRONMENT (FDS/SDE)
TUTORIAL

JULY 1986
FLIGHT DYNAMICS SYSTEM
SOFTWARE DEVELOPMENT
ENVIRONMENT (FDS/SDE)
TUTORIAL

JULY 1986

NASA
National Aeronautics and
Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771
FOREWORD

The Software Engineering Laboratory (SEL) is an organization sponsored by the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) and created for the purpose of investigating the effectiveness of software engineering technologies when applied to the development of applications software. The SEL was created in 1977 and has three primary organizational members:

NASA/GSFC (Systems Development and Analysis Branch)
The University of Maryland (Computer Sciences Department)
Computer Sciences Corporation (Flight Systems Operation)

The goals of the SEL are (1) to understand the software development process in the GSFC environment; (2) to measure the effect of various methodologies, tools, and models on this process; and (3) to identify and then to apply successful development practices. The activities, findings, and recommendations of the SEL are recorded in the Software Engineering Laboratory Series, a continuing series of reports that includes this document.

The primary contributors to this document are

John Buell (Computer Sciences Corporation)
Philip Myers (Computer Sciences Corporation)

Single copies of this document can be obtained by writing to

Frank E. McGarry
Code 552
NASA/GSFC
Greenbelt, Maryland 20771
ABSTRACT

A sample development scenario using the Flight Dynamics System Software Development Environment (FDS/SDE) is presented. The SDE uses a menu-driven, fill-in-the-blanks format that provides online help at all steps, thus eliminating lengthy training and allowing immediate use of this new software development tool.
# TABLE OF CONTENTS

Section 1 - Introduction ........................................... 1-1

Section 2 - Basic SDE Terms and Concepts ....................... 2-1

  2.1 SDE Terms ....................................................... 2-1
  2.2 SDE Concepts .................................................... 2-2
  2.3 327x Keyboard and Display Screen ........................... 2-3
  2.4 ISPF Editor ...................................................... 2-3

Section 3 - Sample Development Scenario .......................... 3-1

References

Standard Bibliography of SEL Literature
SECTION 1 - INTRODUCTION

The Flight Dynamics System Software Development Environment (FDS/SDE) provides an exciting new method for developing software. It uses a menu-driven, fill-in-the-blanks format that permits the developer to input, edit, compile, link, and execute software. Online help is always available, thus eliminating a lengthy training period and allowing immediate use of this new tool.

This tutorial demonstrates the use of the SDE following this simple eight-step scenario:

1. A software developer logs on to the IBM timesharing system (TSO) and enters the SDE (pages 3-2 through 3-14).
2. An IBM job card is formatted (pages 3-15 through 3-21).
3. A main program and two subroutines are entered into an existing PANVALET library (pages 3-22 through 3-49).
4. The program and subroutines are compiled with the IBM VS FORTRAN compiler to produce object modules (pages 3-50 through 3-74).
5. The object modules are merged together (link edited) into an executable program (load module) (pages 3-75 through 3-88).
6. The executable program is run (pages 3-89 through 3-97).
7. All listing files are sent to a printer for output (pages 3-98 through 3-105).
8. The developer updates the SDE analysis log, exits the SDE, and logs off TSO (pages 3-106 through 3-109).

At the beginning of the tutorial, each step of the developer's actions is shown on a separate page to highlight the specific action being taken. Later, all actions taken by a developer before pressing the <enter> key are shown on the same page. Shaded areas denote the changes from panel to panel (display to display). The underscore character in some of the shaded areas is not typed in; it simply marks the current cursor position as it actually appears on the screen.

The SDE has many more functions that aid the developer to create programs. All the capabilities of the Interactive System Productivity Facility (ISPF) are detailed in Reference 1. Reference 2 explains the use of all SDE functions that have been added to the basic ISPF.
Before beginning the tutorial, a few basic terms and concepts of ISPF processing must be explained. The terms—panels, commands, help, and messages—and the concepts—levels, navigation, and jumps—are used extensively in discussions of ISPF and the SDE. Brief descriptions of the 327x keyboard and display screen and the ISPF editor (used in the scenario to create FORTRAN source modules) are also presented.

2.1 SDE TERMS

- Panels—Four basic types of panels are used in the tutorial, although more are available on the ISPF.
  - Selection panels display a list of options in a menu format along with an associated option code. The selected option code is entered in a field near the top of the screen that displays OPTION==>. Parameter entry panels are reached after selections have been made. Parameters for a specific function, such as file names or listing options, are entered on these panels.
  - Data entry panels are used almost exclusively by the editor. They allow data to be input to the computer.
  - Help panels provide instructions on how to use a specific function.

- Commands—ISPF and SDE commands (print, down, up, etc.) can be entered in the parameter entry field next to the COMMAND===> or OPTION===> prompts. The ISPF tutorial (Option 8.T) explains all commands.
Help--Help is available online, pertinent to whatever panel is being displayed, whenever needed. Help is obtained by pressing program function key 1 <PFl> or by typing "help" next to the COMMAND ==> or OPTION ==> prompt.

Messages--Messages are displayed in short and/or long format. The short format always appears in the upper right corner of the display. A long message, which further explains the short message, is displayed when <PFl> is pressed. When a short message is being displayed, help is obtained by pressing <PFl> twice (once to get the long message, once more for the help panel).

2.2 SDE CONCEPTS

Levels--The SDE may be visualized as a tree-structured environment. The primary panel is the trunk, which has major branches. Each branch can in turn have more branches, depending on how many subselections are available. For example,

```
Primary
  Option
    Panel
      Browse Edit Language Translation etc.
        Assemble VS FORTRAN Compile Pascal/VS Compile
```

2-2

0253
Navigation--Navigation through the SDE (traversing the tree) is achieved by selecting codes displayed next to the various options on each selection panel.

Jumps--Random access to option nodes can be achieved by entering an equal sign (=) and an option sequence in any parameter entry field, bypassing the intermediate option selections. This capability is demonstrated in scenario step 5.

2.3 327x KEYBOARD AND DISPLAY SCREEN

The keys on the 327x-type terminals are used to facilitate the use of ISPF and SDE functions. Table 2-1 describes the hardware keys used on the 327x keyboard. Table 2-2 describes the default program function (PF) keys used by the ISPF, as defined for this tutorial; they can be changed using option 0. The text symbol is used throughout the scenario to specify a particular key. The keyboard symbol varies slightly among 327x terminals made by different manufacturers.

The displays presented in the sample development scenario in Section 3 are surrounded by a facsimile of a 327x display screen. The lower portion of each display is the 327x system status area. Table 2-3 describes the symbols that may be displayed in this status area throughout the scenario.

2.4 ISPF EDITOR

The sample development scenario uses the ISPF editor to create FORTRAN source modules. Before beginning the scenario, a few basic editor commands should be explained. The ISPF editor is designed to take advantage of the capabilities of the 327x-type display devices. These devices transmit and receive data in block mode, meaning that data are transmitted to and from the computer in large, screen-size blocks.
Table 2-1. Hardware Keys

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Keyboard Symbol</th>
<th>Text Symbol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alt</td>
<td>ALT</td>
<td>&lt;alt&gt;</td>
<td>Used to perform any function illustrated on the front of keys</td>
</tr>
<tr>
<td>Backward tab</td>
<td>←</td>
<td>&lt;BT&gt;</td>
<td>Moves the cursor left to the previous entry field</td>
</tr>
<tr>
<td>Forward tab</td>
<td>➞</td>
<td>&lt;FT&gt;</td>
<td>Moves the cursor right to the next entry field</td>
</tr>
<tr>
<td>New line</td>
<td>←</td>
<td>&lt;NL&gt;</td>
<td>Moves the cursor down to the next entry field</td>
</tr>
<tr>
<td>Home</td>
<td>⇧</td>
<td>&lt;home&gt;</td>
<td>Moves the cursor to the first data entry field on the top left (alternate function of forward tab)</td>
</tr>
<tr>
<td>Cursor select</td>
<td>CURSOR</td>
<td>&lt;csel&gt;</td>
<td>Toggles the cursor between a box and an underscore character</td>
</tr>
<tr>
<td>Clear screen</td>
<td>CLEAR</td>
<td>&lt;cls&gt;</td>
<td>Clears the entire display area and positions the cursor in the top left of the screen (alternate function of cursor select)</td>
</tr>
<tr>
<td>Erase EOF</td>
<td>ERASE</td>
<td>&lt;EEOF&gt;</td>
<td>Erases all characters from the cursor position to the right edge of a data entry field</td>
</tr>
<tr>
<td>Arrow keys</td>
<td>👆👇</td>
<td>-</td>
<td>Moves the cursor up/down/left/right one position, not to next entry field</td>
</tr>
<tr>
<td>Enter</td>
<td>ENTER</td>
<td>&lt;enter&gt;</td>
<td>Transmits all information entered to computer</td>
</tr>
<tr>
<td>Insert</td>
<td>a</td>
<td>&lt;ins&gt;</td>
<td>Puts screen into insert mode; characters typed will not overwrite those characters already displayed on the screen (see &lt;res&gt;)</td>
</tr>
<tr>
<td>Delete</td>
<td>x</td>
<td>&lt;del&gt;</td>
<td>Deletes character displayed at the current cursor position</td>
</tr>
<tr>
<td>Reset</td>
<td>RESET</td>
<td>&lt;res&gt;</td>
<td>Takes the terminal out of insert mode and unlocks the keyboard after an illegal action has been flagged (when the stick figure can be seen in the status area)</td>
</tr>
</tbody>
</table>

0253
<table>
<thead>
<tr>
<th>Key Name</th>
<th>Keyboard Symbol</th>
<th>Text Symbol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help</td>
<td>PF1</td>
<td>&lt;help&gt;</td>
<td>Displays a long message if a short message is being displayed; displays a help panel if no message is present or if pressed a second time while a message is being displayed</td>
</tr>
<tr>
<td>Split screen</td>
<td>PF2</td>
<td>&lt;split&gt;</td>
<td>Splits the screen into a second copy of ISPF/SDE at the current cursor position</td>
</tr>
<tr>
<td>End</td>
<td>PF3</td>
<td>&lt;end&gt;</td>
<td>Ends the current display</td>
</tr>
<tr>
<td>Return</td>
<td>PF4</td>
<td>&lt;rtn&gt;</td>
<td>Returns to the primary option panel</td>
</tr>
<tr>
<td>Find next</td>
<td>PF5</td>
<td>&lt;fnxt&gt;</td>
<td>Finds next occurrence of a pattern in edit or browse</td>
</tr>
<tr>
<td>Replace Next</td>
<td>PF6</td>
<td>&lt;rnxt&gt;</td>
<td>Replaces next occurrence of a pattern in edit only</td>
</tr>
<tr>
<td>Scroll Up</td>
<td>PF7</td>
<td>&lt;up&gt;</td>
<td>Moves the display window up</td>
</tr>
<tr>
<td>Scroll down</td>
<td>PF8</td>
<td>&lt;down&gt;</td>
<td>Moves the display window down</td>
</tr>
<tr>
<td>Swap</td>
<td>PF9</td>
<td>&lt;swap&gt;</td>
<td>Moves the cursor to the opposite window</td>
</tr>
<tr>
<td>Scroll left</td>
<td>PF10</td>
<td>&lt;left&gt;</td>
<td>Moves the display window to the left</td>
</tr>
<tr>
<td>Scroll right</td>
<td>PF11</td>
<td>&lt;right&gt;</td>
<td>Moves the display window to the right</td>
</tr>
<tr>
<td>Cursor</td>
<td>PF12</td>
<td>&lt;crsr&gt;</td>
<td>Moves the cursor to the top left entry position in the current window</td>
</tr>
</tbody>
</table>
Table 2-3. System Status Area Symbols

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock</td>
<td>🕝</td>
<td>Wait</td>
</tr>
<tr>
<td>System</td>
<td>🌐</td>
<td>CPU is processing last request</td>
</tr>
<tr>
<td>Stick</td>
<td>🧵</td>
<td>Something was typed in an area of the display that is not a valid input area; press the &lt;res&gt; key to recover from this problem; use the &lt;F7&gt;, &lt;F8&gt;, or &lt;F9&gt; keys to move the cursor into an input area</td>
</tr>
<tr>
<td>figure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>caret</td>
<td>^</td>
<td>Insert mode is turned on</td>
</tr>
<tr>
<td>Caps lock</td>
<td>🖼</td>
<td>The &lt;caps lock&gt; key has been pressed</td>
</tr>
</tbody>
</table>
The 327x device processes information displayed on the screen and modifies it until the user presses <enter>. At that time, the screen is transmitted to the computer, a response is received from the computer, and the screen is rewritten.

The ISPF editor has the capability to insert, delete, move, copy, and sort single lines or blocks of lines anywhere in a file, at the discretion of the developer. A few basic editing capabilities are described below. The developer is, however, urged to use the online help available to learn all of the editor's capabilities.

- Inserting characters--To insert a character, press the <ins> key on the keyboard (note the ^ in the status area), position the cursor on the screen where characters are to be inserted, and begin typing. Characters to the right of the cursor will be pushed to the right as new characters are inserted. If pushing the characters would make the one on the far right of the line go off the screen, the keyboard will lock. Press <res> to recover. Then use the <right> key to move the display window so that inserted characters and the moving characters are in the display area.

If the editor will not allow characters to be inserted when there are apparently no characters on the line (it displays the stick figure and clicks when any key is struck), it is trying to shift blank characters to the right while new characters are being typed. This can be remedied either by moving the cursor to the right of the last nonblank character on the line and pressing <EEOF> or by typing "nulls on" in the command line area. "Nulls on" tells the editor to pad empty lines with null characters instead of blanks.
Changing a character--To change a character, simply position the cursor on top of the character to be changed and type over it. If insert mode is on (^ is in the status area), press <res> to turn it off.

Deleting characters--To delete a character, position the cursor on the character to be deleted and press the <del> key. All characters on the right of the cursor will be shifted to the left by one column.

Moving a line--To move a line, mark the destination by typing an "a" or "b" on the line number the line is to be moved after (a) or before (b), and mark the source by typing an "m" on the line to be moved. Press <enter>, and the line marked with an "m" will be moved.

Copying a line--To copy a line, mark the destination by typing an "a" or "b" on the line number the line is to be copied after (a) or before (b), and mark the source by typing a "c" on the line to be copied. Press <enter>, and the line marked with a "c" will be copied.

Deleting lines--To delete lines from a file, move the cursor into the left side of the display with the <NL> key. Position it on each line number you want to delete, and type a "d" to mark it. After all lines to be deleted have been marked, press <enter> and all the lines marked will disappear from the file.

Inserting a line--To insert a new line in a file, move the cursor into the numbered area of the display with the <NL> key. Position it on the line number the new line is to be inserted after, and type an "i". Press <enter>, and a line with a row of dots in the line number area will appear. Position the cursor where desired with the <arrow> keys, and type in the new line. When <enter> is pressed,
another row of dots will appear and the process can be repeated. The line insertion process ends when <enter> is pressed but nothing is typed.

- Inserting many lines--To insert many new lines in a file, move the cursor into the numbered area of the display with the <NL> key. Position it on the line number the new lines are to be inserted after and type an "i<n>", where <n> is the number of lines to be inserted. Press <enter>, and <n> rows of dots will appear in the numbered area. Position the cursor where desired with the <arrow> keys, and type in the new information.

- Displaying next page of file--To display the next page of a file, press <PF8> or type "down" in the command line input area.

- Displaying previous page of file--To display the previous page of a file, press <PF7> or type "up" in the command line input area.
SECTION 3 - SAMPLE DEVELOPMENT SCENARIO

Step one of the scenario requires logging on to TSO. The procedure differs depending on whether the terminal is connected directly to the FDF3 at Goddard Space Flight Center (GSFC) or is connected through a modem. Most of the 327x terminals connected to the FDF3 at GSFC are direct-connect terminals. Developers using these terminals should begin the scenario on page 3-2. Developers using terminals that are connected to the FDF3 through a modem should begin the scenario on page 3-4.
Scenario Step 1: Logging on TSO and entering SDE

ENTER LOGON FOR GODDARD SPACE FLIGHT CENTER SYSTEM - FDF3

This is what will appear on the top line of the terminal before you log on to the 4341 computer. If the terminal that you want to use does not display the request seen above, press and hold the <alt> key and the <cse1> key. This performs the alternate function of that key, which is clear-screen. If you still don't get the request message, tell your supervisor.
Scenario Step 1: Logging on TSO and entering SDE

Type in your user identifier (userid), sponsor code, and project code. You can get this information from your supervisor. The userid, sponsor, and project codes above are gjzzp, spon, and test, respectively.

Press <enter>, the screen will clear. . . .

(go to page 3-10)
Scenario Step 1: Logging on TSO and entering SDE

This panel is displayed before any action is taken at the terminal. It is called the VM prompt panel, or the logo panel, and indicates the terminal is connected to the Flight Dynamics Facility (FDF) computer system. There are currently three computers in the FDF, referred to as the F1, F2, and the F3. The Software Development Environment (SDE) is only installed on the F3. The first action to take is to connect the terminal to the F3 computer. This is done by pressing the <enter> key...
Scenario Step 1: Logging on TSO and entering SDE

and this panel is displayed, indicating the terminal is connected to the F1 computer ('FDF-F1VM' is displayed in the lower right corner).

Next we type in ...
Scenario Step 1: Logging on TSO and entering SDE

'dial mvs', then press <enter>, to use the MVS operating system.

The screen will clear ...
Scenario Step 1: Logging on TSO and entering SDE

DIAL TO MVS
DIALED TO MVS

'DIAL TO MVS' is repeated at the top left corner of the screen. 'DIALED TO MVS' and three hexadecimal digits appear briefly two lines lower on the screen; the screen clears again, and . . .
Scenario Step 1: Logging on TSO and entering SDE

\[\text{ENTER LOGON FOR GODDARD SPACE FLIGHT CENTER SYSTEM - FDF1}\]

This message appears on the top line of the terminal, indicating the terminal is connected to the F1 computer under the MVS operating system.

A special logon command, `LOGONF3`, switches the terminal from the F1 to the F3 computer.
Type in LOGONF3, your user identifier (userid), sponsor code, and project code. You can get this information from your supervisor. The userid, sponsor, and project codes above are gjzzp, spons, and test, respectively.

Press <enter>, the screen will clear.
Scenario Step 1: Logging on TSO and entering SDE

USS COMMAND HAS BEEN SUCCESSFULLY EXECUTED

...and this message is displayed briefly after the logon command has been accepted. In a short time the screen will clear again, and ...
Scenario Step 1: Logging on TSO and entering SDE

... you are prompted for your TSO password. Type it in. The letters you type will not be displayed. Press <enter> when you have finished; in a few seconds ...
Scenario Step 1: Logging on TSO and entering SDE

ENTER YOUR PASSWORD

GJZZP LOGON IN PROGRESS AT 14:53:11 ON FEBRUARY 26, 1986

------------------------------------------------------------------------
** IBM/4341 (FDF3)  T S O  MVS/SP 1.3.3 **
** For problem assistance, please call the P.A.C. at 344-6768. **
------------------------------------------------------------------------
** Downtimes ****
** None currently scheduled **
------------------------------------------------------------------------
READY

... these messages will appear. The time and date will differ from the ones shown here, and there may be some downtime scheduled to maintain the F3 computer system, but this basic format will be displayed.

We have successfully logged on to TSO and now must enter the software development environment. To do this...
Scenario Step 1: Logging on TSO and entering SDE

ENTER YOUR PASSWORD

GJZZP LOGON IN PROGRESS AT 14:53:11 ON FEBRUARY 26, 1986

******************************************************
** IBM/4341 (PFD3) ==> T S O ===> MVS/SP 1.3.3 **
** For problem assistance, please call the P.A.C. at 344-6768. **
******************************************************
** ==> Downtimes ===> **
** None currently scheduled **
******************************************************

READY

ex 'pjeoe.sde.clist'

... type in the command above and press <enter>. In a few moments ...
Scenario Step 1: Logging on TSO and entering SDE

This panel is displayed as we enter the Software Development Environment (SDE). It is called the primary option panel and lists all of the major options available to the SDE developer. The underscore character in the upper left is only marking the current cursor position, it never has to be typed in. In order to perform step 2 of our scenario, we have to define an IBM job card. Job cards are used to provide accounting information and several other types of information used to process background jobs.
Scenario Step 2: Defining IBM Job Card

--- FDS/SDE PRIMARY OPTION MENU VERSION 2.2 ---

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DEFAULTS - Specify terminal and user parameters</td>
</tr>
<tr>
<td>1</td>
<td>BROWSE - Display source data or output listings</td>
</tr>
<tr>
<td>2</td>
<td>EDIT - Create or change source data</td>
</tr>
<tr>
<td>3</td>
<td>UTILITIES - Perform utility functions (copy, allocate, rename, list)</td>
</tr>
<tr>
<td>4</td>
<td>COMPILE - Invoke language translators (Asm, Fort, Pascal, GESS)</td>
</tr>
<tr>
<td>5</td>
<td>LINK - Invoke linkage-editor (build load modules)</td>
</tr>
<tr>
<td>6</td>
<td>TSO - Enter TSO command or CLIST</td>
</tr>
<tr>
<td>7</td>
<td>TEST - Perform dialog testing</td>
</tr>
<tr>
<td>8</td>
<td>NEWS/VIEWS - Display news or enter comments about ISPF/PDF/SDE</td>
</tr>
<tr>
<td>9</td>
<td>MGT INFO - Use on-line management information systems</td>
</tr>
<tr>
<td>10</td>
<td>MISC - Miscellaneous Software Development Environment Features</td>
</tr>
<tr>
<td>F</td>
<td>FILE AID - Direct Access data handling utility</td>
</tr>
<tr>
<td>JS</td>
<td>JOB STATUS - Using SPOOL Display and Search Facility (SDSF)</td>
</tr>
<tr>
<td>L</td>
<td>LOG - Update SDE/SEL Data Base log with ISPF log data</td>
</tr>
<tr>
<td>P</td>
<td>PANVALET - Browse, edit, and utilities for Panvalet data sets</td>
</tr>
<tr>
<td>X</td>
<td>EXIT - Terminate ISPF using log and list defaults</td>
</tr>
</tbody>
</table>

Enter END command to terminate ISPF.

Type a 0 in the option selection field to select DEFAULTS and press the <enter> key.
Scenario Step 2: Defining IBM Job Card

The FDS/SDE parameter options selection panel is displayed...
Scenario Step 2: Defining IBM Job Card

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 TERMINAL</td>
<td>Specify terminal characteristics</td>
</tr>
<tr>
<td>2 LOG/LIST</td>
<td>Specify ISPF log and list defaults</td>
</tr>
<tr>
<td>3 PF KEYS</td>
<td>Specify PF keys for 3278 terminal with 24 PF keys</td>
</tr>
<tr>
<td>4 DISPLAY</td>
<td>Specify screen display characteristics</td>
</tr>
<tr>
<td>5 LIST</td>
<td>Specify list data set characteristics</td>
</tr>
<tr>
<td>JC JOB CARD</td>
<td>Specify Batch Job Card to be used</td>
</tr>
</tbody>
</table>

Type in JC to select the job card option, press <enter>, and...
Scenario Step 2: Defining IBM Job Card

Enter/Change Batch Job Cards for GUZZP

Command ===> _

Job Id ===> 
Sponsor Code ===> 
Project Id ===> 
Destination Box ===> 

Job Class ===> Message Class ===> 
Message Level ===> Alloc/Term. ===> 
Time (Minutes) ===> 

Notify Whom when Job Terminates ===> (Blank for No Notify)

Printer to Route Output to ===> (Blank for local print)

Current Job Cards:

//GUZZP JOB (ACCOUNTING INFORMATION)
//*
//*
//*

Press Enter to process changes. End or Return to exit this function

... the parameter entry panel for the change job card function is displayed. The actual job cards displayed at the bottom may differ on your display the first time you enter this function. In a moment they will be very similar.
Scenario Step 2: Defining IBM Job Card

------------- Enter/Change Batch Job Cards for GJZZP -------------

Command >>>

Job Id >>> A
Sponsor Code >>> SPONS
Project Id >>> TEST
Destination Box >>> CCC

Job Class >>> A
Message Class >>> A
Message Level >>>
Alloc/Term. >>>
Time (Minutes) >>> 3

Notify Whom when OdD Terminates >>> GJZZP (Blank for No Notify)

Printer to Route Output to >>> PRJ20 (Blank for local print)

Current Job Cards:

//GJZZP JOB (ACCOUNTING INFORMATION)
//
//
//

Press Enter to process changes. End or Return to exit this function

Pressing <help> will let you read the help panels associated with this function. They explain the various fields and the values that could be entered. Rather than get bogged down with details, use the values for each field as they appear on this panel. <FT> to move the cursor from input field to input field.

Press <enter>, and...
Scenario Step 2: Defining IBM Job Card

----------------------- Enter/Change Batch Job Cards for JOB.CARDS.GENERATED

Command ===> _

Job id ===> A
Sponsor Code ===> SPONS
Project Id ===> TEST
Destination Box ===> CCC

Job Class ===> A Message Class ===> A
Message Level ===> 1 Alloc/Term. ===> 1
Time (Minutes) ===> 3

Notify Whom when Job Terminates ===> GJZZP (Blank for No Notify)

Printer to Route Output to ===> PRT23 (Blank for local print)

Current Job Cards:
//GJZZPA JOB (SPONS,TEST,CCC),TIME=0003.
// MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP
//*ROUTE PRINT PRT23
//*

Press Enter to process changes, End or Return to exit this function

... the values typed in each field will be gathered together, formatted and displayed as a valid MVS job card. The short message 'JOB CARDS GENERATED' appears in the upper right corner of the screen.

Silver Spring developers should use FFF instead of CCC, and PRTSS instead of PRT23: your actual sponsor code and project ID will vary; these are fictitious.

when satisfied that the job card is correct, press <end> and ...

3-20
Scenario Step 2: Defining IBM Job Card

FDS/SDE PARAMETER OPTIONS

OPTION ===> _

1 TERMINAL   - Specify terminal characteristics
2 LOG/LIST   - Specify ISPF log and list defaults
3 PF KEYS    - Specify PF keys for 3278 terminal with 24 PF keys
4 DISPLAY    - Specify screen display characteristics
5 LIST       - Specify list data set characteristics
JC JDBCARD   - Specify Batch Job Card to be used

... we will go up one level to the DEFAULTS selection panel. Press <end> again and ...
Scenario Step 3: Creating/Editing Panvalet member

Enter END command to terminate ISPF.

... we are back at the primary option panel.

In order to perform step 3 of our scenario, we have to create a FORTRAN main routine in a Panvalet library. To do this, we should ...
Scenario Step 3: Creating/Editing Panvalet member

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 DEFAULTS</td>
<td>Specify terminal and user parameters</td>
</tr>
<tr>
<td>1 BROWSE</td>
<td>Display source data or output listings</td>
</tr>
<tr>
<td>2 EDIT</td>
<td>Create or change source data</td>
</tr>
<tr>
<td>3 UTILITIES</td>
<td>Perform utility functions (copy, allocate, rename, list)</td>
</tr>
<tr>
<td>4 COMPIL</td>
<td>Invoke language translators (Asm, Fort, Pascal, GESS)</td>
</tr>
<tr>
<td>5 LINK</td>
<td>Invoke linkage-editor (build load modules)</td>
</tr>
<tr>
<td>6 TSO</td>
<td>Enter TSO command or CLIST</td>
</tr>
<tr>
<td>7 TEST</td>
<td>Perform dialog testing</td>
</tr>
<tr>
<td>8 NEWS/VIEWS</td>
<td>Display news or enter comments about ISPF/PDF/SDE</td>
</tr>
<tr>
<td>9 MGT INFO</td>
<td>Use on-line management information systems</td>
</tr>
<tr>
<td>10 MISC</td>
<td>Miscellaneous Software Development Environment Features</td>
</tr>
<tr>
<td>F FILE AID</td>
<td>Direct Access data handling utility</td>
</tr>
<tr>
<td>JS JOB STATUS</td>
<td>Using SPOOL Display and Search Facility (SDSF)</td>
</tr>
<tr>
<td>L LOG</td>
<td>Update SDE/SEL Data Base log with ISPF log data</td>
</tr>
<tr>
<td>P PANVALET</td>
<td>Browse, edit, and utilities for Panvalet data sets</td>
</tr>
<tr>
<td>X EXIT</td>
<td>Terminate ISPF using log and list defaults</td>
</tr>
</tbody>
</table>

Enter END command to terminate ISPF.

... type option 'P' and press the <enter> key, which ...
Scenario Step 3: Creating/Editing Panvalet member

... displays the Panvalet function's selection panel. Panvalet is a database management system that specializes in storing source code. It can also store data or anything else that can be grouped into 80-character records ("card-images", for trivia buffs). Each separate entity is referred to as a 'member' in a Panvalet library. Members can be grouped by using the same first two characters as a prefix in a name.
Scenario Step 3: Creating/Editing Panvalet member

PANVALET PRIMARY MENU

OPTION ==> 2

1 BROWSE  - SCAN PANVALET MEMBERS

2 EDIT    - UPDATE PANVALET MEMBERS

3 UTILITY - PANVALET UTILITIES (COPY, RENAME, STATUS, ETC.)

USE END KEY (PF3) TO BACK OUT OF PANVALET MODE

Select option 2, edit, and press <enter> ...
Scenario Step 3: Creating/Editing Panvalet member

---

**PANVALET EDIT PANEL**

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>STANDARD PANVALET LIBRARY:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJECT</td>
<td>_</td>
</tr>
<tr>
<td>LIBRARY</td>
<td></td>
</tr>
<tr>
<td>TYPE</td>
<td></td>
</tr>
<tr>
<td>MEMBER</td>
<td>(BLANK FOR MEMBER SELECTION LIST)</td>
</tr>
</tbody>
</table>

**NON STANDARD PANVALET LIBRARY:**

<table>
<thead>
<tr>
<th>PANVALET LIB</th>
<th>(IF NOT CATALOGED)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLUME SERIAL</td>
<td></td>
</tr>
</tbody>
</table>

**MEMBER SELECTION LIST OPTIONS:**

- LIST MEMBERS STARTING WITH
- LIST MEMBERS WITH LANG TYPE
- LIST MEMBERS WITH USER CODE
- LIST MEMBERS WITH STATUS

**NEW MEMBER OPTIONS:**

- LANG TYPE
- USER CODE (OPTIONAL)

**PANVALET RETRIEVAL OPTIONS:**

- CONTROL
- PAN/TSO
- EXPAND
- PROFILE
- INITIAL MACRO

... and the Panvalet edit function's parameter entry panel is displayed. This panel is where we 'fill in the blanks' that provide the Panvalet edit function with the information it needs to begin processing.

We are going to create three new members in an existing Panvalet library that is named GJZZP.TEST.PAN. If you do not have a Panvalet library, use option 3.2.2 to create one.
### Scenario Step 3: Creating/Editing Panvalet member

---

#### PANVALET EDIT PANEL

**COMMAND ====>***

**STANDARD PANVALET LIBRARY:**

- **PROJECT ====>** [gizzp]
- **LIBRARY ====>**
- **TYPE ====>**
- **MEMBER ====>** (BLANK FOR MEMBER SELECTION LIST)

**NON STANDARD PANVALET LIBRARY:**

- **PANVALET LIB ====>**
- **VOLUME SERIAL ====>** (IF NOT CATALOGED)

**MEMBER SELECTION LIST OPTIONS:**

- **LIST MEMBERS STARTING WITH ====>**
- **LIST MEMBERS WITH LANG TYPE ====>**
- **LIST MEMBERS WITH USER CODE ====>**
- **LIST MEMBERS WITH STATUS ====>**

**NEW MEMBER OPTIONS:**

- **LANG TYPE ====>**
- **USER CODE ====>** (OPTIONAL) **NOFORMAT ====>** N (Y OR N)

**PANVALET RETRIEVAL OPTIONS:**

- **CONTROL ====>**
- **PAN/TSD ====> PAN** (PAN OR TSD SEQUENCING)
- **ACCESS ====>**
- **EXPAND ====>** N (Y OR N)
- **PROFILE ====>** (DEFAULTS TO MEMBER LANGUAGE TYPE)
- **INITIAL MACRO ====>**

---

Grouping all members of the same system or subsystem by prefix allows us to refer to the group by prefix when using some of the SDE functions. Even though this scenario will not use the functions, we'll prefix the component names with 'TS' (for Test). Use the <NL> key to move the cursor to the PROJECT name qualifier parameter entry field and type in **gizzp** . . .
Scenario Step 3: Creating/Editing Panvalet member

--------------------------- PANVALET EDIT PANEL ---------------------------

COMMAND ===>

STANDARD PANVALET LIBRARY:
PROJECT ===> gjzzp
LIBRARY ===> test
TYPE ===> 
MEMBER ===> (BLANK FOR MEMBER SELECTION LIST)

NON STANDARD PANVALET LIBRARY:
PANVALET LIB ===> 
VOLUME SERIAL ===> (IF NOT CATALOGED)
MEMBER SELECTION LIST OPTIONS: WITH COMMENTS => N ("Y" OR "N")
LIST MEMBERS STARTING WITH ===> 
LIST MEMBERS WITH LANG TYPE ===> 
LIST MEMBERS WITH USER CODE ===> 
LIST MEMBERS WITH STATUS ===> 

NEW MEMBER OPTIONS:
LANG TYPE ===> 
USER CODE ===> (OPTIONAL) NOFORMAT ===> N (Y OR N)

PANVALET RETRIEVAL OPTIONS:
CONTROL ===> PAN/TSD ===> PAN (PAN OR TSO SEQUENCING)
ACCESS ===> EXPAND ===> N (Y OR N)
PROFILE ===> (DEFAULTS TO MEMBER LANGUAGE TYPE)
INITIAL MACRO ===> 

... Use the <NL> key to move to the LIBRARY parameter entry field and type in test...
Scenario Step 3: Creating/Editing Panvalet member

Scenario Step 3: Creating/Editing Panvalet member

--- PANVALET EDIT PANEL ---

COMMAND ==> VERSION - 10.4A

STANDARD PANVALET LIBRARY:
PROJECT ==> gjzzd
LIBRARY ==> test
TYPE ==> MEM
MEMBER ==> (BLANK FOR MEMBER SELECTION LIST)

NON STANDARD PANVALET LIBRARY:
PANVALET LIB ==> VOLUME SERIAL ==> (IF NOT CATALOGED)

MEMBER SELECTION LIST OPTIONS:
LIST MEMBERS STARTING WITH ==> WITH COMMENTS ==> N ("Y" OR "N")
LIST MEMBERS WITH LANG TYPE ==> LIST MEMBERS WITH USER CODE ==> LIST MEMBERS WITH STATUS ==>

NEW MEMBER OPTIONS:
LANG TYPE ==> USER CODE ==> (OPTIONAL) NOFORMAT ==> N (Y OR N)

PANVALET RETRIEVAL OPTIONS:
CONTROL ==> PAN/TSQ ==> PAN (PAN OR TSO SEQUENCING)
ACCESS ==> EXPAND ==> N (Y OR N)
PROFILE ==> (DEFAULTS TO MEMBER LANGUAGE TYPE)
INITIAL MACRO ==>

... Use the <NL> key to move to the TYPE parameter entry field and type in pan ...

3-29
Scenario Step 3: Creating/Editing Panvalet member

----------------------------------------------------- PANVALET EDIT PANEL -----------------------------------------------------

COMMAND >>>

STANDARD PANVALET LIBRARY:

PROJECT >>> gjzzp
LIBRARY >>> test
TYPE >>> pan
MEMBER >>> tssample (BLANK FOR MEMBER SELECTION LIST)

NON STANDARD PANVALET LIBRARY:

PANVALET LIB >>>
VOLUME SERIAL >>> (IF NOT CATALOGED)

MEMBER SELECTION LIST OPTIONS: WITH COMMENTS >> N ("Y" OR "N")

LIST MEMBERS STARTING WITH >>>
LIST MEMBERS WITH LANG TYPE >>>
LIST MEMBERS WITH USER CODE >>>
LIST MEMBERS WITH STATUS >>>

NEW MEMBER OPTIONS:

LANG TYPE >>>
USER CODE >>> (OPTIONAL) NOFORMAT >>> N (Y OR N)

PANVALET RETRIEVAL OPTIONS:

CONTROL >>>
ACCESS >>> EXPAND >>> N (Y OR N)
PROFILE >>> (DEFAULTS TO MEMBER LANGUAGE TYPE)
INITIAL MACRO >>>

... Press <NL> to move to the MEMBER parameter entry field and type in the prefix and name combination tssample ...

3-30
Scenario Step 3: Creating/Editing Panvalet member

--- PANVALET EDIT PANEL ---

COMMAND ==> STANDARD PANVALET LIBRARY:
PROJECT ==> gjzzp
LIBRARY ==> test
TYPE ==> pan
MEMBER ==> tssample

(BLANK FOR MEMBER SELECTION LIST)

NON STANDARD PANVALET LIBRARY:
PANVALET LIB ==> VOLUME SERIAL ==> (IF NOT CATALOGED)

MEMBER SELECTION LIST OPTIONS:
LIST MEMBERS STARTING WITH ==> WITH COMMENTS ==> N ("Y" OR "N")
LIST MEMBERS WITH LANG TYPE ==> LIST MEMBERS WITH USER CODE ==> LIST MEMBERS WITH STATUS ==> NEW MEMBER OPTIONS:
LANG TYPE ==> fortran
USER CODE ==> (OPTIONAL) NOFORMAT ==> N (Y OR N)

PANVALET RETRIEVAL OPTIONS:
CONTROL ==> PAN/TSO ==> PAN (PAN OR TSO SEQUENCING)
ACCESS ==> EXPAND ==> N (Y OR N)
PROFILE ==> (DEFAULTS TO MEMBER LANGUAGE TYPE)
INITIAL MACRO ==>

... and finally <NL> seven times to the language type parameter entry field and type in fortran. The language type entry is only required when a new Panvalet library member is to be created.

Now, press the <enter> key ...
Scenario Step 3: Creating/Editing Panvalet member

... and the Panvalet edit data entry panel is displayed. The dots on the left indicate new line positions. Now we can begin entering the program into memory.

Press <FT> at the end of each line...
Scenario Step 3: Creating/Editing Panavlet member

PVEDIT -- GUZZP тест. PANTSSAMPLE -- ------------------- COLUMNS 001 072
COMMAND ==>

program sample

cc

purpose: show the use of the basic sde capabilities by

cc

writing, compiling, linking and running a program that will ask

cc

a user for two real numbers, then call 2 subroutines that will

cc

compute and display their sum, difference, product, and quotient.

---

and the cursor will move to the far left of the next line. Press the <FT> key again...
Scenario Step 3: Creating/Editing Panvalet member

```plaintext
COMMAND ===> SCROLL ===> 0018

program sample
purpose: show the use of the basic sde capabilities by
writing, compiling, linking and running a program that will ask
a user for two real numbers, then call 2 subroutines that will
compute and display their sum, difference, product, and quotient.

... and the cursor will move to the first column of the data entry area.
Continue to type the program into the computer.

When the last line of the program has been typed in, press <enter> and ...
```

3-34
we notice two things have happened. All of the text has been converted to upper case, and line numbers have appeared where the dots were. The program must now be saved. Press the <end> key...
Scenario Step 3: Creating/Editing Panvalet member

and the Panvalet edit parameter entry panel will be redisplayed. Note
the message 'TSSAMPLE SAVED' in the upper right corner of the screen. This
is an example of a short message.
Next we want to create the two subroutines that will be called by our main routine. Type in the prefix/name combination `tsadsub`. ...
Scenario Step 3: Creating/Editing Panvalet member

---

PANVALET EDIT PANEL

---

TSSAMPLE SAVED

Command ==>

Standard Panvalet library:

Project ==> GJZZP
Library ==> TEST
Type ==> PAN
Member ==> tsaOdSUD

(Blank for member selection list)

Non standard Panvalet library:

Panvalet lib ==> 
Volume serial ==> 

(Member selection list options: with comments => N ("Y" or "N")
List members starting with ==> 
List members with lang type ==> 
List members with user code ==> 
List members with status ==> 

New member options:

Lang type ==> fortran
User code ==> (Optional) Noformat ==> N (Y or N)

Panvalet retrieval options:

Control ==> Pan/Tso ==> PAN (PAN or TSO sequencing)
Access ==> Expand ==> N (Y or N)
Profile ==> 
Initial macro ==> 

... seven times to the language type field; type in fortran; and press <enter>...
Scenario Step 3: Creating/Editing Panvalet member

... and we are again in the Panvalet edit data entry mode. Type in the subroutine just like we did the main routine; press <enter> ...
Scenario Step 3: Creating/Editing Panvalet member

PVEDIT --- GJZZP.TEST.PAN(TSADDSUB)------------------------------- COLUMNS 001 072
COMMAND ====> SCROLL ====> 0018

-------- ******* TOP OF DATA ******* *******

000001 SUBROUTINE ADDSUB(FIRST,SECOND)
000002 CC
000003 C PURPOSE: COMPUTE AND DISPLAY THE SUM AND DIFFERENCE OF TWO REAL
000004 C NUMBERS.
000005 C
000006 C
000007 C compute the sum
000008 SUM = FIRST + SECOND
000009 C compute the difference
000010 DIFFER = FIRST - SECOND
000011 C write out the sum
000012 WRITE(6,1000) FIRST, SECOND, SUM
000013 1000 FORMAT(’ THE SUM OF ’,F8.2,’ AND ’,FB.2,’ IS ’,F10.2)
000014 C
000015 C write out the difference
000016 WRITE(6,2000) FIRST, SECOND, DIFFER
000018 C
000019 RETURN
000020 END

-------- ******* BOTTOM OF DATA ******* *******

and we have our first subroutine finished. Press <end> . . . .

3-40
Scenario Step 3: Creating/Editing Panvalet member

--------- PANVALET EDIT PANEL --------- TSADDSUB SAVED

COMMAND ===> 

STANDARD PANVALET LIBRARY:
  PROJECT ===> GJZZP
  LIBRARY ===> TEST
  TYPE ===> PAN
  MEMBER ===> _  (BLANK FOR MEMBER SELECTION LIST)

NON STANDARD PANVALET LIBRARY:
  PANVALET LIB ===> 
  VOLUME SERIAL ===>  (IF NOT CATALOGED)

MEMBER SELECTION LIST OPTIONS: 
  LIST MEMBERS STARTING WITH ===> 
  LIST MEMBERS WITH LANG TYPE ===> 
  LIST MEMBERS WITH USER CODE ===> 
  LIST MEMBERS WITH STATUS ===> 

NEW MEMBER OPTIONS:
  LANG TYPE ===> 
  USER CODE ===> (OPTIONAL)  NOFORMAT ===> N (Y OR N)

PANVALET RETRIEVAL OPTIONS:
  CONTROL ===> 
  PAN/TSO ===> PAN  (PAN OR TSO SEQUENCING)
  ACCESS ===> 
  EXPAND ===> N (Y OR N)
  PROFILE ===>  (DEFAULTS TO MEMBER LANGUAGE TYPE)
  INITIAL MACRO ===> 

... and we are back in the Panvalet parameter entry panel. Note the 'TSADDUS SAVED' message in the upper right of the panel.
Type in the prefix/name combination of the last subroutine, tsmuld, press <enter>, and ...
Scenario Step 3: Creating/Editing Panvalet member

--- PANVALET EDIT PANEL --- NEW MBR. NEED LANG TYPE

COMMAND ==> 

STANDARD PANVALET LIBRARY:

PROJECT ==> GUZZP
LIBRARY ==> TEST
TYPE ==> PAN
MEMBER ==> TSMULDIV (BLANK FOR MEMBER SELECTION LIST)

NON STANDARD PANVALET LIBRARY:

PANVALET LIB ==> 
VOLUME SERIAL ==> (IF NOT CATALOGED)

MEMBER SELECTION LIST OPTIONS:

LIST MEMBERS STARTING WITH ==> 
LIST MEMBERS WITH LANG TYPE ==> 
LIST MEMBERS WITH USER CODE ==> 
LIST MEMBERS WITH STATUS ==> 

NEW MEMBER OPTIONS:

LANG TYPE ==> USER CODE ==> (OPTIONAL) NOFORMAT ==> N (Y OR N)

PANVALET RETRIEVAL OPTIONS:

CONTROL ==> PAN/TSO ==> PAN (PAN OR TSO SEQUENCING)
ACCESS ==> EXPAND ==> N (Y OR N)
PROFILE ==> (DEFAULTS TO MEMBER LANGUAGE TYPE)
INITIAL MACRO ==> 

OOPS! the console beeps and we have a message in the short message area. It's a new member and we forgot to specify the type. Notice the cursor is automatically positioned at the data entry field that is required.
Scenario Step 3: Creating/Editing Panvalet member

------------- PANVALET EDIT PANEL -- NEW MBR. NEED LANG TYPE -------------

COMMAND >>>

STANDARD PANVALET LIBRARY:

PROJECT >>> GUZZP
LIBRARY >>> TEST
TYPE >>> PAN
MEMBER >>> TSMULDIV (BLANK FOR MEMBER SELECTION LIST)

NON STANDARD PANVALET LIBRARY:

PANVALET LIB >>>
VOLUME SERIAL >>> (IF NOT CATALOGED)

MEMBER SELECTION LIST OPTIONS:

LIST MEMBERS STARTING WITH >>>
LIST MEMBERS WITH LANG TYPE >>>
LIST MEMBERS WITH USER CODE >>>
LIST MEMBERS WITH STATUS >>>

NEW MEMBER OPTIONS:

LANG TYPE >>> fortran
USER CODE >>> (OPTIONAL)  NOFORMAT >>> N (Y OR N)

PANVALET RETRIEVAL OPTIONS:

CONTROL >>> PAN/TSD >>> PAN (PAN OR TSO SEQUENCING)
ACCESS >>> EXPAND >>> N (Y OR N)
PROFILE >>> (DEFAULTS TO MEMBER LANGUAGE TYPE)

INITIAL MACRO >>>

Type in fortran: press <enter> again . . .
Scenario Step 3: Creating/Editing Panvalet member

and we are at the Panvalet edit data entry panel. By now you should have no problem typing in the last subroutine.

Remember that help is available by typing 'help' in the 'COMMAND===' field or by pressing <help>. The edit help panels explain all of the bells and whistles available to the edit user.
Scenario Step 3: Creating/Editing Panvalet member

PVEDIT --- GJZZP.TEST.PAN(TSMULDIV)----------------------------- COLUMNS 001 072
COMMAND ===>
SCROLL ===>

******** *************** TOP OF DATA ***************

000001 SUBROUTINE MULDIV(R1,R2)

000002 C

000003 C PURPOSE: COMPUTE AND DISPLAY THE PRODUCT AND QUOTIENT OF TWO

000004 C REAL NUMBERS

000005 C INITIALIZE VARIABLES

000006 C QUOT = 0.0

000007 C COMPUTE THE PRODUCT

000008 C PROD = R1 * R2

000009 C COMPUTE THE QUOTIENT

000010 C IF (R2 .NE. 0) QUOT = R1 / R2

000011 C WRITE OUT THE PRODUCT

000012 C WRITE(6,1000) R1, R2, PROD

000013 1000 FORMAT(' THE PRODUCT OF ',F8.2,' AND ',F8.2,' IS ',F14.4)

000014 C

000015 C WRITE OUT THE QUOTIENT

000016 C WRITE(6,2000) R1, R2, QUOT

000017 2000 FORMAT(' THE QUOTIENT OF ',F8.2,' OVER ',F8.2,' IS ',F10.2)

000018 C

000019 C RETURN

000020 C END

******** *************** BOTTOM OF DATA ***************

Once again, this is what the display looks like when it is typed in and the <enter> key is pressed. Press <end> to return to the edit parameter entry panel....
Scenario Step 3: Creating/Editing Panvalet member

Note the short message 'TSMULDIV SAVED' and press <end> to go up one level.
Scenario Step 3: Creating/Editing Panvalet member

--- PANVALET PRIMARY MENU ---

OPTION ==> _

1 BROWSE - SCAN PANVALET MEMBERS
2 EDIT - UPDATE PANVALET MEMBERS
3 UTILITY - PANVALET UTILITIES (COPY, RENAME, STATUS, ETC.)

USE END KEY (PF3) TO BACK OUT OF PANVALET MODE

... to the Panvalet selection panel. Press <end> to go up one more level...
Scenario Step 3: Creating/Editing Panvalet member

---

FDS/SDE PRIMARY OPTION MENU VERSION 2.2

**OPTION \(=**\)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DEFAULTS - Specify terminal and user parameters</td>
</tr>
<tr>
<td>1</td>
<td>BROWSE - Display source data or output listings PF KEYS - 24</td>
</tr>
<tr>
<td>2</td>
<td>EDIT - Create or change source data</td>
</tr>
<tr>
<td>3</td>
<td>UTILITIES - Perform utility functions (copy, allocate, rename, list)</td>
</tr>
</tbody>
</table>
| 4      | COMPIL<

```plaintext
USERID - GJZZP
TIME - 15:05
TERMINAL - 3278
PF KEYS - 24
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>LINK - Invoke linkage-editor (build load modules)</td>
</tr>
<tr>
<td>6</td>
<td>TSO - Enter TSO command or CLIST</td>
</tr>
<tr>
<td>7</td>
<td>TEST - Perform dialog testing</td>
</tr>
<tr>
<td>8</td>
<td>NEWS/VIEWS - Display news or enter comments about ISPF/PDF/SDE</td>
</tr>
<tr>
<td>9</td>
<td>MGT INFO - Use on-line management information systems</td>
</tr>
<tr>
<td>10</td>
<td>MISC - Miscellaneous Software Development Environment Features</td>
</tr>
<tr>
<td>F</td>
<td>FILE AID - Direct Access data handling utility</td>
</tr>
<tr>
<td>JS</td>
<td>JOB STATUS - Using SPOOL Display and Search Facility (SDSF)</td>
</tr>
<tr>
<td>L</td>
<td>LOG - Update SDE/SEL Data Base log with ISPF log data</td>
</tr>
<tr>
<td>P</td>
<td>PANVALET - Browse, edit, and utilities for Panvalet data sets</td>
</tr>
<tr>
<td>X</td>
<td>EXIT - Terminate ISPF using log and list defaults</td>
</tr>
</tbody>
</table>

Enter END command to terminate ISPF.

---

... to the primary option panel. We have just completed step 3 of our scenario.
Scenario Step 4: Compiling program and subroutines

--- FDS/SDE PRIMARY OPTION MENU VERSION 2.2 ---

OPTION >>> 4

0 DEFAULTS - Specify terminal and user parameters
1 BROWSE - Display source data or output listings
2 EDIT - Create or change source data
3 UTILITIES - Perform utility functions (copy, allocate, rename, list)
4 COMPILE -Invoke language translators (Asm, Fort, Pascal, GESS)
5 LINK - Invoke linkage-editor (build load modules)
6 TSO - Enter TSO command or CLIST
7 TEST - Perform dialog testing
8 NEWS/VIEWS - Display news or enter comments about ISPF/PDF/SDE
9 MGT INFO - Use on-line management information systems
10 MISC - Miscellaneous Software Development Environment Features
F FILE AID - Direct Access data handling utility
J JOB STATUS - Using SPOOL Display and Search Facility (SDSF)
L LOG - Update SDE/SEL Data Base log with ISPF log data
P PANVALET - Browse, edit, and utilities for Panvalet data sets
X EXIT - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.

Step 4 requires that we compile these three routines. To do this, type 4, then <enter> ...
Scenario Step 4: Compiling program and subroutines

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System assembler</td>
</tr>
<tr>
<td>2</td>
<td>VS FORTRAN compiler</td>
</tr>
<tr>
<td>3</td>
<td>PASCAL/VS compiler</td>
</tr>
<tr>
<td>4</td>
<td>GE5S interactive display builder</td>
</tr>
</tbody>
</table>

... and we arrive at the compiler/translator function's selection panel. We wish to perform a FORTRAN compilation, so ...
Scenario Step 4: Compiling program and subroutines

------------------------
COMPILER/TRANSLATOR SELECTION MENU
------------------------

OPTION ===> 2

BATCH ===> (Job will be submitted for batch processing if non-blank)

1 - System assembler
2 - VS FORTRAN compiler
3 - PASCAL/VS compiler
4 - GESS interactive display builder

Type 2, then press <enter>, and ...
Scenario Step 4: Compiling program and subroutines

--- FOREGROUND VS FORTRAN COMPILE ---

**COMMAND ====>**

**ISPF SOURCE FILE LIBRARY:**
- **PROJECT ====>** _
- **GROUP ====>**
- **TYPE ====>**
- **MEMBER ====>** (Blank for member selection list)

**OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:**
- **DATA SET NAME ====>**
- **PANVALET MEMBER NAME ====>** **MEMBER PREFIX ====>** (optional)

**COMPILER INCLUDE FILE LIBRARY:**
- **DATA SET NAME ====>**

**ALTERNATE PANVALET INCLUDE FILE LIBRARY:**
- **DATA SET NAME ====>**

- **LIST ID ====>** **Automatic Browse? ====> NO** **Automatic Print? ====> NO**

**COMPILER OPTIONS:**
- **====>**

... the foreground VS FORTRAN compile parameter entry panel is displayed. <NL>

four times to the DATA SET NAME parameter entry field ...

3-53
Scenario Step 4: Compiling program and subroutines

--------------- FOREGROUND VS FORTRAN COMPILE ---------------

COMMAND 

ISPF SOURCE FILE LIBRARY:
PROJECT 
GROUP 
TYPE 
MEMBER (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
DATA SET NAME 
PANVALET MEMBER NAME MEMBERS PREFIX (optional)

COMPILER INCLUDE FILE LIBRARY:
DATA SET NAME

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
DATA SET NAME

LIST ID Automatic Browse? NO Automatic Print? NO

COMPILER OPTIONS:

... and type in the name of the Panvalet library (even though our TSO user id is GUZZP, we are explicitly defining our data set by using quotes). <FT> once to the member name field ...

3-54
Scenario Step 4: Compiling program and subroutines

----------------------------- FOREGROUND VS FORTRAN COMPIL ---------------

COMMAND ===> 

ISPF SOURCE FILE LIBRARY:
PROJECT ===> 
GROUP ===> 
TYPE ===> 
MEMBER ===> (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
DATA SET NAME ===> 'gjzzp.test.pan'
PANVALET MEMBER NAME ===> SAMPLE MEMBER PREFIX ===> (optional)

COMPILER INCLUDE FILE LIBRARY:
DATA SET NAME ===> 

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
DATA SET NAME ===> 

LIST ID ===> Automatic Browse? ===> NO Automatic Print? ===> NO

COMPILER OPTIONS: 
 ===> 

... and type in the Panvalet member name, SAMPLE. <FT> once more to the member prefix field...
Scenario Step 4: Compiling program and subroutines

FOREGROUND VS FORTRAN Compile

COMMAND 

ISPF SOURCE FILE LIBRARY:
- PROJECT 
- GROUP 
- TYPE 
- MEMBER (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
- DATA SET NAME 
- PANVALET MEMBER NAME 
- MEMBER PREFIX (optional)

COMPILER INCLUDE FILE LIBRARY:
- DATA SET NAME

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
- DATA SET NAME

LIST ID Automatic Browse? NO Automatic Print? NO

COMPILER OPTIONS:

... and type in the Panvalet member prefix, TS. The INCLUDE FILES are not required, <help> will explain them if you wish to know about them. The listing ID will default to the prefix and member name.

Note that the automatic browse and automatic print answers are both 'NO'. We should change this for a first compile. <FT> to these fields ...

3-56
Scenario Step 4: Compiling program and subroutines

--- FOREGROUND VS FORTRAN COMPIL **E --- **---

**COMMAND ***=>**

**ISPF SOURCE FILE LIBRARY:**

**PROJECT ***=>**
**GROUP ***=>**
**TYPE ***=>**
**MEMBER ***=>** (Blank for member selection list)

**OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:**

**DATA SET NAME ***=>** 'gjzzp.test.pan'
**PANVALET MEMBER NAME ***=>** sample  **MEMBER PREFIX ***=>** ts (optional)

**COMPILER INCLUDE FILE LIBRARY:**

**DATA SET NAME ***=>**

**ALTERNATE PANVALET INCLUDE FILE LIBRARY:**

**DATA SET NAME ***=>**

**LIST ID ***=>**  **Automatic Browse? ***=>** y  **Automatic Print? ***=>** y

**COMPILER OPTIONS:**

***=>

... and change them both to 'y'. SDE will accept y, ye, or yes in most cases; blank defaults to 'NO'. The entries can be in upper or lower case.

Now <FT> to the compiler options field ...
Scenario Step 4: Compiling program and subroutines

<table>
<thead>
<tr>
<th>FOREGROUND VS FORTRAN COMPILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND:</td>
</tr>
</tbody>
</table>

**ISPF SOURCE FILE LIBRARY:**
- **PROJECT**: 
- **GROUP**: 
- **TYPE**: 
- **MEMBER**: (Blank for member selection list)

**OTHER PARTITIONED, SEQUENTIAL, OR PANVALET DATA SET:**
- **DATA SET NAME**: 'gjzzp.test.pan'
- **PANVALET MEMBER NAME**: 'sample' **MEMBER PREFIX**: 'ts' (optional)

**COMPILER INCLUDE FILE LIBRARY:**
- **DATA SET NAME**: 

**ALTERNATE PANVALET INCLUDE FILE LIBRARY:**
- **DATA SET NAME**: 

**LIST ID**: Automatic Browse? **y** Automatic Print? **y**

**COMPILER OPTIONS:**
- ```
  **magenta**
  `timeaccount(80).xref`
  ```

... and type in a couple of compiler options. The object module output by the compiler will be named 'GJZZP.TS.SAMPLE.OBJ'. Press <enter> and the compile will begin. The screen will clear...
Scenario Step 4: Compiling program and subroutines

RETRIEVING TSSAMPLE AS GUZZP.TS.SAMPLE.FORT
DONE 21 STMT(S) LEVEL 001
VS FORTRAN COMPILER ENTERED. 15:56:24
*STATISTICS* SOURCE STATEMENTS = 11, PROGRAM SIZE = 810 BYTES, PROGRAM NAME = SAMPLE PAGE: 1.
*STATISTICS* NO DIAGNOSTICS GENERATED.
**SAMPLE** END OF COMPILATION 1 *****
VS FORTRAN COMPILER EXITED. 15:56:25

and these messages will be displayed as the compile progresses. A few seconds after the VS FORTRAN COMPILER EXITED message is displayed...
Scenario Step 4: Compiling program and subroutines

BROWSE - GUZP.TS.SAMPLE.LIST ........................ LINE 000000 CDL 001 080
COMMAND ===> _ SCROLL ===> 00:18
                         TOP OF DATA

LEVEL 1.3.1 (FEB 1984) VS FORTRAN DATE: FEB 26, 1986 TIME:
REQUESTED OPTIONS (EXECUTE): LINECOUNT(80),XREF
OPTIONS IN EFFECT: NOLIST NOMAP XREF NOGOSTMT NODECK SOURCE TERM OBJECT
                           OPT(O) LONGLONG(77) NOFIPS FLAG(I) NAME(MAIN ) LINECDU
                           *....*1........2........3........4........5........6.....
                           C DATA SET TSSAMPLE AT LEVEL 001 AS OF 02/26/86

ISN 1 PROGRAM SAMPLE
    CC
    C PURPOSE: SHOW THE USE OF THE BASIC FDS/SDE CAPABILITIES BY
    C WRITING, COMPIILING, LINKING AND RUNNING A PROGRAM THAT WILL
    C A USER FOR TWO REAL NUMBERS, THEN CALL 2 SUBROUTINES THAT W
    C COMPUTE AND DISPLAY THEIR SUM, DIFFERENCE, PRODUCT, AND QUO
    C PROMPT THE USER FOR TWO REAL N

ISN 2 WRITE(6,1000)
ISN 3 1000 FORMAT(' ENTER YOUR FIRST REAL NUMBER: ')
ISN 4 READ(5,1010) R1
ISN 5 1010 FORMAT(F6.2)

ISN 6 WRITE(6,2000)
ISN 7 2000 FORMAT(' ENTER YOUR SECOND REAL NUMBER: ')

... this display will be seen. It is a "browse" of the listing file. Note
the listing file name has defaulted to the prefix and member as anticipated.

If you wish you can use <up>, <down>, <left>, or <right> to scroll the display
window around the listing file. The listing file is 132 characters wide.
Scenario Step 4: Compiling program and subroutines

BROWSE - GUZZP.TS.SAMPLE.LIST

LEVEL 1.3.1 (FEB 1984) VS FORTRAN

REQUESTED OPTIONS (EXECUTE): LINECOUNT(80), XREF

OPTIONS IN EFFECT: NOLIST NODMAP XREF NOGOSTMT NODECK SOURCE TERM OBJECT

OPT(O) LANGLEVEL(77) NOFIPS FLAG(1) NAME(MAIN) LINECOUNT

DATA SET TSSAMPLE AT LEVEL 001 AS OF 02/26/86

ISN 1 PROGRAM SAMPLE

ISN 2 WRITE(6,1000)

ISN 3 1000 FORMAT(' ENTER YOUR FIRST REAL NUMBER: ')

ISN 4 READ(5,1010) R1

ISN 5 1010 FORMAT(F6.2)

ISN 6 WRITE(6,2000)

ISN 7 2000 FORMAT(' ENTER YOUR SECOND REAL NUMBER: ')

ISN 8 READ(5,1010) R2

Type 'find error' in the command line field of the display, press <enter>.

3-61
Scenario Step 4: Compiling program and subroutines

BROWSE - GJ2ZP.TS.SAMPLE.LIST .............................................. NO CHARS 'error' FOUND
COMMAND ===> _ SCROLL ===> 00:18

----------------- TOP OF DATA ----------------
LEVEL 1.3.1 (FEB 1984) VS FORTRAN DATE: FEB 26, 1986 TIME:
REQUESTED OPTIONS (EXECUTE): LINECOUNT(80), XREF
OPTIONS IN EFFECT: NOLIST NOMAP XREF NOGOSTMT NODECK SOURCE TERM OBJECT
OPT(0) LANGlvl(77) NOFIPS FLAG(I) NAME(MAIN ) LINEDDU
*...*1...........2...........3...........4...........5...........6...
C DATA SET TSSAMPLE AT LEVEL 001 AS OF 02/26/86

ISN 1 PROGRAM SAMPLE

CC
C PURPOSE: SHOW THE USE OF THE BASIC FDS/SDE CAPABILITIES BY
C WRITING, COMPILING, LINKING AND RUNNING A PROGRAM THAT WILL
C A USER FOR TWO REAL NUMBERS, THEN CALL 2 SUBROUTINES THAT W
C COMPUTE AND DISPLAY THEIR SUM, DIFFERENCE, PRODUCT, AND QUO
C
ISN 2 WRITE(6,1000)
ISN 3 1000 FORMAT(' ENTER YOUR FIRST REAL NUMBER: ')
ISN 4 READ(5,1010) R1
ISN 5 1010 FORMAT(F6.2)
C
ISN 6 WRITE(6,2000)
ISN 7 2000 FORMAT(' ENTER YOUR SECOND REAL NUMBER: ')
ISN 8 READ(S,1010) R2

... and the message NO CHARS 'error' FOUND will appear in the upper right of
the display. If the word "error" was in the file, as in an error message, the
cursor would be positioned at that location in the file. This lets us see the
lines that are wrong without scanning through the entire listing.

3-62
Scenario Step 4: Compiling program and subroutines

BROWSE - GUZZP.TS.SAMPLE.LIST

COMMAND → down max

SCROLL → oo18

LEVEL 1.3.1 (FEB 1984) VS FORTRAN
DATE: FEB 26, 1986
TIME:

REQUESTED OPTIONS (EXECUTE): LINECOUNT(80), XREF

OPTIONS IN EFFECT: NOLIST NOMAP XREF NOGOSTMT NODECK SOURCE TERM OBJECT

OPT(O) LANGLEVEL(77) NOFIPS FLAG(I) NAME(MAIN ) LINECOU

DATA SET TSSAMPLE AT LEVEL 001 AS OF 02/26/86

ISN 1

PROGRAM SAMPLE

PURPOSE: SHOW THE USE OF THE BASIC FDS/SDE CAPABILITIES BY
WRITING,-compiling, linking and running a program that will
A USER FOR TWO REAL NUMBERS, THEN CALL 2 SUBROUTINES THAT W
COMPUTE AND DISPLAY THEIR SUM, difference, PRODUCT, AND QUO
PROMPT THE USER FOR TWO REAL N

ISN 2

WRITE(6,1000)

ISN 3

1000 FORMAT( ' ENTER YOUR FIRST REAL NUMBER: ' )

ISN 4

READ(5,1010) R1

ISN 5

1010 FORMAT(F6.2)

ISN 6

WRITE(6,2000)

ISN 7

2000 FORMAT( ' ENTER YOUR SECOND REAL NUMBER: ' )

ISN 6

READ(5,1010) R2

Type 'down max' in the command line field and press <enter> . . .
Scenario Step 4: Compiling program and subroutines

and we will skip to the bottom of the listing field. In this way we can
review the processing statistics, in case they flashed by when we weren’t
looking.

When you are finished scrolling around the listing file, press <end> . . .
Scenario Step 4: Compiling program and subroutines

FOREGROUND PRINT OPTIONS

OPTION ==> 
PK - Print data set and keep
PD - Print data set and delete
K - Keep data set (without printing)
D - Delete data set (without printing)

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

DATA SET NAME: GJZZP.TS.SAMPLE.LIST

SYSOUT CLASS ==> A  (For system printer)
PRINTER ID ==>  (For 32Bx printer)

JOB STATEMENT INFORMATION:  (Required for system printer)

```plaintext
===> //GJZZPM JOB (SPONS,TEST,CCC),TIME=0003.
===> // MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP
===> /*ROUTE PRINT PRT23
===> */
```

... and the print option panel is displayed. Note the job control information is the same as that which we entered in option O.JC. That's why we did that first.
Scenario Step 4: Compiling program and subroutines

------------------------ FOREGROUND PRINT OPTIONS ------------------------

OPTION => (PK)

PK - Print data set and keep       K - Keep data set (without printing)
PD - Print data set and delete     D - Delete data set (without printing)

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

DATA SET NAME: GUZZP. TS. SAMPLE.LIST

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

JOB STATEMENT INFORMATION:          (Required for system printer)
  => //GUZZPM JOB (SPONS,TEST,CCC),TIME=0003,
  => // MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GUZZP
  => /*ROUTE PRINT PRT23
  => /*

Type in the option PK to print and keep the listing data set. We want to keep it in order to show you another option later in this scenario.

Press <enter> and . . .
Scenario Step 4: Compiling program and subroutines

The VS FORTRAN compile parameter entry panel is redisplayed. Note the short message in the upper right corner of the display. This message indicates the compiler function ended with a return code of zero, meaning no errors were found. The next thing to compile is the ADDSUB subroutine, so we position the cursor at the Panvalet member name field and ...
Scenario Step 4: Compiling program and subroutines

------------------------ FOREGROUND VS FORTRAN COMPIL FUNCTION ENDED RC=0

COMMAND ===> 

ISPF SOURCE FILE LIBRARY:
PROJECT ===> 
GROUP ===> 
TYPE ===> 
MEMBER ===> (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
DATA SET NAME ===> 'GUZZP.TEST.PAN'
PANVALET MEMBER NAME ===> ADDSUB MEMBER PREFIX ===> TS (optional)

COMPILER INCLUDE FILE LIBRARY:
DATA SET NAME ===> 

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
DATA SET NAME ===> 

LIST ID ===> Automatic Browse? ===> YES Automatic Print? ===> 

COMPILER OPTIONS:
 ===> LINECOUNT(80).XREF

... type ADDSUB over top of SAMPLE. We will print the listing file later using another option, so <FT> to the automatic print question and blank out the entry with the <EEOF> key. All blanks are converted to NO by the SDE.

Press <enter> . . .
and just like the last compile, the screen will clear, the messages shown above will be displayed, and shortly after the last message...
Scenario Step 4: Compiling program and subroutines

BROWSE - GJZZP.TS.ADDSUB.LIST ---------------------- LINE 000000 COL 001 080
COMMAND ===> _ SCROLL ===> 0018

LEVEL 1.3.1 (FEB 1984) VS FORTRAN DATE: FEB 26, 1986 TIME:
REQUESTED OPTIONS (EXECUTE): LINECOUNT(80), XREF
OPTIONS IN EFFECT: NOLIST NOMAP XREF NOGOSTMT NODECK SOURCE TERM OBJECT

OPT(0) LANGLEVEL(77) NOSFIPS FLAG(1) NAME(MAIN ) LINELIMIT

DATA SET TSADDSUB AT LEVEL 002 AS OF 02/26/86

ISN 1 SUBROUTINE ADDSUB(FIRST,SECOND)
CC
C PURPOSE: COMPUTE AND DISPLAY THE SUM AND DIFFERENCE OF TWO
C NUMBERS.

ISN 2 SUM = FIRST + SECOND
CC
C COMPUTE THE SUM

ISN 3 DIFFER = SECOND - FIRST
CC
C COMPUTE THE DIFFERENCE

ISN 4 WRITE(6,1000) FIRST, SECOND, SUM

ISN 5 1000 FORMAT(' THE SUM OF ',F8.2,' AND ',F8.2,' IS ',F10.2)

we will automatically browse this listing file. Press <end>, and...

3-70
Scenario Step 4: Compiling program and subroutines

FOREGROUND VS FORTRAN COMPIL FUNCTION ENDED RC=0

COMMAND ===>

ISPF SOURCE FILE LIBRARY:

PROJECT ===> 

GROUP ===> 

TYPE ===> 

MEMBER ===> (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:

DATA SET NAME ===> 'GJZZP.TEST.PAN'

PANVALET MEMBER NAME ===> ADDSUB MEMBER PREFIX ===> TS (optional)

COMPILER INCLUDE FILE LIBRARY:

DATA SET NAME ===> 

ALTERNATE PANVALET INCLUDE FILE LIBRARY:

DATA SET NAME ===> 

LIST ID ===> Automatic Browse? ===> YES Automatic Print? ===> NO

COMPILER OPTIONS:

====> LINECOUNT(80),XREF

... and the VS FORTRAN compiler parameter entry panel is again displayed.

Note the short message indicates no errors were found (return code = 0). Object module 'GJZZP.TS.ADDSUB.OBJ' (or TS.ADDSUB.OBJ) has been created.
Scenario Step 4: Compiling program and subroutines

FOREGROUND VS FORTRAN COMPIL FUNCTION ENDED RC=0

COMMAND ===> 

ISPF SOURCE FILE LIBRARY:
PROJECT ===> 
GROUP ===> 
TYPE ===> 
MEMBER ===> (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
DATA SET NAME ===> 'GUZZP.TEST.PAN'
PANVALET MEMBER NAME ===> MULDIV MEMBER PREFIX ===> TS (optional)

COMPILER INCLUDE FILE LIBRARY:
DATA SET NAME ===> 

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
DATA SET NAME ===> 

LIST ID ===> Automatic Browse? ===> NO Automatic Print? ===> NO

COMPILER OPTIONS:
 ===> LINECOUNT(80),XREF

Now it's time to create the last object module, TS.MULDIV.OBJ. Type in MULDIV as the member name. <FT> to the automatic browse question and blank it out; then press <enter>. 

3-72
RETRIEVING TSMULDIV AS GUZZP.TS.MULDIV.FORT
DONE 20 STMT(S) LEVEL 001
VS FORTRAN COMPILER ENTERED. 16:10:28
*STATISTICS* SOURCE STATEMENTS = 10, PROGRAM SIZE = 934 BYTES, PROGRAM NAME = MULDIV PAGE: 1.
*STATISTICS* NO DIAGNOSTICS GENERATED.
**MULDIV** END OF COMPILATION 1 -------
VS FORTRAN COMPILER EXITED. 16:10:29

The messages for the compile will be displayed, but this time three asterisks will appear at the bottom of the screen. Three asterisks are displayed until we acknowledge having seen the results. Press <enter> to acknowledge, and ...
Scenario Step 4: Compiling program and subroutines

FOREGROUND VS FORTRAN COMPIL FUNCTION ENDED RC=0

COMMAND ===>

ISPF SOURCE FILE LIBRARY:
PROJECT ===>
GROUP ===>
TYPE ===>
MEMBER ===> (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
DATA SET NAME ===> 'GJZZP.TEST.PAN'
PANVALET MEMBER NAME ===> MULDIV MEMBER PREFIX ===> TS (optional)

COMPILER INCLUDE FILE LIBRARY:
DATA SET NAME ===>

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
DATA SET NAME ===>

LIST ID ===>
Automatic Browse? ===> YES
Automatic Print? ===> NO

COMPILER OPTIONS:
====> LINECOUNT(80),XREF

... and the VS FORTRAN compile parameter entry screen is redisplayed. Note by the short message that the function once again ended with a return code of 0. Boy are we good (actually this scenario was rigged).

This concludes step 4 of our scenario. Now it’s time to learn a handy trick.
Scenario Step 5: Linking Program

------------- FOREGROUND VS FORTRAN COMPIL FUNCTION ENDED RC=0
COMMAND ===> 

ISPF SOURCE FILE LIBRARY:
PROJECT ===> 
GROUP ===> 
TYPE ===> 
MEMBER ===> (Blank for member selection list)

OTHER PARTITIONED, SEQUENTIAL, or PANVALET DATA SET:
DATA SET NAME ===> 
PANVALET MEMBER NAME ===> MULDIV MEMBER PREFIX ===> TS (optional)

COMPILER INCLUDE FILE LIBRARY:
DATA SET NAME ===> 

ALTERNATE PANVALET INCLUDE FILE LIBRARY:
DATA SET NAME ===> 

LIST ID ===> Automatic Browse? ===> YES Automatic Print? ===> NO

COMPILER OPTIONS:
====> LINECOUNT(80),XREF

What we’re going to do is jump directly to the link edit function, without pressing <end> to back up to the primary option panel and descending down the path to the link edit function. To do this, position the cursor in any input field and type in =5 (this means "select option 5 from the primary option panel"). Press <enter> . . .
Scenario Step 5: Linking program

---------- LINKAGE EDITOR/PROGRAM BUILDER ----------

OPTION ===>  

BATCH ===> (job will be submitted for batch processing if non-blank)

1 - Simple Output Load Module (no external references resolved, i.e. NCAL)

2 - Assembler Source (w/ Utility Libraries)

3 - VS Fortran Source (w/ Utility and Language Libraries)

4 - VS Pascal Source (w/ Utility and Language Libraries)

5 - VS Fortran & VS Pascal Source (w/ Utility and Language Libraries)

6 - Link Edit with User Defined Output Library

7 - GESS Driver entry point

... and we scoot to the linkage editor selection panel. This is option 5 from the primary option panel. We want to build a program using a temporary library that we will define, so...
Scenario Step 5: Linking program

---------- LINKAGE EDITOR/PROGRAM BUILDER ----------

OPTION = 6

BATCH ==> (job will be submitted for batch processing if non-blank)

1 - Simple Output Load Module (no external references resolved, i.e. NCAL)

2 - Assembler Source (w/ Utility Libraries)

3 - VS Fortran Source (w/ Utility and Language Libraries)

4 - VS Pascal Source (w/ Utility and Language Libraries)

5 - VS Fortran & VS Pascal Source (w/ Utility and Language Libraries)

6 - Link Edit with User Defined Output Library

7 - GESS Driver entry point

... select option 6 and press <enter>.
Scenario Step 5: Linking program

The user-defined libraries linkage edit parameter entry panel is displayed. Note the automatic browse and automatic print answers are the same as we saw in the VS FORTRAN compile panel. The answers to these two questions will always be retained between functions and between SDE sessions.
Scenario Step 5: Linking program

FOREGROUND LINKAGE EDIT WITH USER DEFINED LIBRARIES

COMMAND ==> 

LOAD LIBRARY ==> TUTORIAL.LOAD
MEMBER ==> EXAMPLE

INPUT FILE ==> TS.SAMPLE.OBJ
PANVALET MEMBER ==> 
PANVALET PREFIX ==> 

LIST ID ==> Automatic Browse? ==> NO Automatic Print? ==> NO

LINKAGE EDITOR OPTIONS:
==> MAP.LIST.LET

Use Additional Object Modules List? ==> Y Display/Change it? ==> Y
Use Additional Object Libraries List? ==> Display/Change it? ==> Display/Change it? ==> 
Use Additional Load Libraries List? ==> Display/Change it? ==> Display/Change it? ==> 

Use VS FORTRAN Libraries? ==> Y
Use Pascal/VS Libraries? ==> 

<FT> to each position indicated by a shaded area and type in the corresponding parameter. The program will be called EXAMPLE and will be saved in a library called GUZZP.TUTORIAL.LOAD. The list ID for link-edits defaults to <userid>.<member name>.LINKLIST. GUZZP.EXAMPLE.LINKLIST in this case.

Press <enter> when finished. The screen will clear, and soon . . .
Scenario Step 5: Linking program

FOREGROUND LINKAGE EDITOR OBJECT MODULE LIST

COMMAND ===> _

Data sets entered into this table will be placed into the linkage editor input stream after the primary input object module.

Press ENTER to execute changes made; press END (PF3) to continue link.

New data set name ===> 

Options: Move, Copy, After, Before, Repeat, Delete, Update (default)

<table>
<thead>
<tr>
<th>Option</th>
<th>Object Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

.......................... BOTTOM OF DATA ..........................

... this panel appears. This panel is displayed in response to the Y in the "Display/Change it" question about the Additional Object Modules List. We want to change the list (because there is nothing in it yet).
Scenario Step 5: Linking program

Data sets entered into this table will be placed into the linkage editor input stream after the primary input object module.

Press ENTER to execute changes made; press END (PF3) to continue link.

New data set name ===> TS.MULDIV.OBJ

Options: Move, Copy, After, Before, Repeat, Delete, Update (default)

Option .......................... Object Module Name

............................................................ BOTTOM OF DATA .............................................................

<FT> to position the cursor in the NEW DATA SET NAME field of the panel. Type in the name of one of the subroutines and press <enter>.
Scenario Step 5: Linking program

Data sets entered into this table will be placed into the linkage editor input stream after the primary input object module.

Press ENTER to execute changes made; press END (PF3) to continue link.

New data set name =>

Options: Move, Copy, After, Before, Repeat, Delete, Update (default)

<table>
<thead>
<tr>
<th>Option</th>
<th>Object Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TS.MULDIV.OBJ</td>
</tr>
</tbody>
</table>

The name will be converted to all caps and will be entered into the list. The cursor will be positioned in the option field for that member name.
Scenario Step 5: Linking program

FOREGROUND LINKAGE EDITOR OBJECT MODULE LIST - ROW 1 OF 1

COMMAND ===> SCROLL ===> PAGE

Data sets entered into this table will be placed into the linkage editor input stream after the primary input object module.

Press ENTER to execute changes made; press END (PF3) to continue link.

New data set name ===> 

Options: M ove, C opy, A fter, B efore, R epeat, D elete, U pdate (default)

<table>
<thead>
<tr>
<th>Option</th>
<th>Object Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TS.MULDIV.OBJ</td>
</tr>
</tbody>
</table>

Type an R (Repeat) as the option for this list name, press <enter>, and ...
Scenario Step 5: Linking program

Data sets entered into this table will be placed into the linkage editor input stream after the primary input object module.

Press ENTER to execute changes made; press END (PF3) to continue link.

New data set name ***>

Options: Move, Copy, After, Before, Repeat, Delete, Update (default)

<table>
<thead>
<tr>
<th>Option</th>
<th>Object Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS.MULDIV.OBJ</td>
<td></td>
</tr>
<tr>
<td>TS.MULDIV.OBJ</td>
<td></td>
</tr>
</tbody>
</table>

... the name will be repeated. Now move the cursor...
Scenario Step 5: Linking program

Data sets entered into this table will be placed into the linkage editor input stream after the primary input object module.

Press ENTER to execute changes made; press END (PF3) to continue link.

New data set name ===> 

Options: Move, Copy, After, Before, Repeat, Delete, Update (default)

Option    Object Module Name
          ----------------------------------------
TS.ADDSUB.OBJ
TS.MULDIV.OBJ

... and type ADDSUB over MULDIV. Press <enter> to update the list, then press <end> to continue the link.
Scenario Step 5: Linking program

The screen will clear and soon this message will be displayed.

In a short time...
Scenario Step 5: Linking program

...... FOREGROUND LINKAGE EDIT WITH USER DEFIN
COMMAND ===> 

LOAD LIBRARY ===> TUTORIAL.LOAD
MEMBER ===> EXAMPLE

INPUT FILE ===> TS.SAMPLE.OBJ
PANVALET MEMBER ===> -
PANVALET PREFIX ===> 

LIST ID ===> Automatic Browse? ===> NO Automatic Print? ===> NO

LINKAGE EDITOR OPTIONS:
 ===> MAP, LIST, LET

Use Additional Object Modules List? ===> Y Display/Change it? ===> Y
Use Additional Object Libraries List? ===> Display/Change it? ===> 
Use Additional Load Libraries List? ===> Display/Change it? ===> 

Use VS FORTRAN Libraries? ===> Y
Use Pascal/VS Libraries? ===> }

... the link edit parameter entry panel will be redisplayed. This function should end with a return code of 0, too. We're batting 1000.
Scenario Step 5: Linking program

---------- FOREGROUND LINKAGE EDIT WITH USER DEFIN FUNCTION ENDED RC=0

COMMAND ===> E

LOAD LIBRARY ===> TUTORIAL.LOAD
MEMBER ===> EXAMPLE

INPUT FILE ===> TS.SAMPLE.OBJ
PANVALET MEMBER ===> 
PANVALET PREFIX ===> 

LIST ID ===> Automatic Browse? ===> YES Automatic Print? ===> NO

LINKAGE EDITOR OPTIONS:
 ===> MAP, LIST, LET

Use Additional Object Modules List? ===> Y Display/Change it? ===> Y
Use Additional Object Libraries List? ===> Display/Change it?
Use Additional Load Libraries List? ===> Display/Change it?

Use VS FORTRAN Libraries? ===> Y
Use Pascal/VS Libraries? ===> 

This completes step 5 of the scenario. Now let's jump to the TSO function to run our program.

Use the <home> key, to put the cursor in the command field. Type in =6 and press <enter>. 
Scenario Step 6: Running program

Here we are in the TSO command processing function of SDE. We need to allocate logical unit 5 for input, logical unit 6 for output, and then fire up the program that we have just entered, compiled, and linked.
Scenario Step 6: Running program

To allocate unit 5 to the keyboard for input, we type in this command and press <enter>.

```plaintext
#pragma allocate file(ft05f001) dataset(*)
```
Scenario Step 6: Running program

---

TSO COMMAND PROCESSOR

ENTER TSO COMMAND OR CLIST BELOW:

... allocate file(ft05f001) dataset(*)

---

To allocate unit 6 to the display tube for output, move the cursor; change the 5 to a 6; and press <enter>.

FT05F001 is the IBM default for logical unit 5, FT06F001 is the IBM default for logical unit 6.
Scenario Step 6: Running program

To execute our program in the foreground, we type in a call to the library and member, and press <enter>.
Scenario Step 6: Running program

---

TSO COMMAND PROCESSOR

ENTER TSO COMMAND OR CLIST BELOW:

***> call tutorial(example)

ENTER YOUR FIRST REAL NUMBER:

Our program will prompt us for a number...
Scenario Step 6: Running program

```
- - - - - - - - - - - - - - TSO COMMAND PROCESSOR - - - - - - - - - - - - - -

ENTER TSO COMMAND OR CLIST BELOW:

*** call tutorial(example)

ENTER YOUR FIRST REAL NUMBER:

323.456
```

We type it in. Then our program will prompt us for another number...
Scenario Step 6: Running program

---------- TSO COMMAND PROCESSOR ----------

ENTER TSO COMMAND OR CLIST BELOW:

****> call tutorial(example)

ENTER YOUR FIRST REAL NUMBER:

123.456

ENTER YOUR SECOND REAL NUMBER:

234.567

Type it in too. After we press <enter> this time . . .
Scenario Step 6: Running program

----------------------------------- TSO COMMAND PROCESSOR -----------------------------------

ENTER TSO COMMAND OR CLIST BELOW:

***> call tutorial(example)

ENTER YOUR FIRST REAL NUMBER:
123.456

ENTER YOUR SECOND REAL NUMBER:
234.567

THE SUM OF 123.45 AND 234.56 IS 358.01
THE DIFFERENCE OF 123.45 AND 234.56 IS 111.11
THE PRODUCT OF 123.45 AND 234.56 IS 28956.4297
THE QUOTIENT OF 123.45 OVER 234.56 IS 0.53

***

... Our program will display the results.
Three asterisks are displayed until we acknowledge having seen the results.
Press <enter> to acknowledge, and ...
Scenario Step 6: Running program

--- TSO COMMAND PROCESSOR ---

ENTER TSO COMMAND OR CLIST BELOW:

```shell
====>
call tutorial(example)
```

the TSO command panel is redisplayed. Press <end>, and...
Scenario Step 7: Printing listing files

--- FDS/SDE PRIMARY OPTION MENU VERSION 2.2 ---

**OPTION ****> _

0 DEFAULTS - Specify terminal and user parameters
1 BROWSE - Display source data or output listings
2 EDIT - Create or change source data
3 UTILITIES - Perform utility functions (copy, allocate, rename, list)
4 COMPILATE - Invoke language translators (Asm, Fort, Pascal, GESS)
5 LINK - Invoke linkage-editor (build load modules)
6 TSO - Enter TSO command or CLIST
7 TEST - Perform dialog testing
8 NEWS/VIEWS - Display news or enter comments about ISPF/PDF/SDE
9 MGT INFO - Use on-line management information systems
10 MISC - Miscellaneous Software Development Environment Features
F FILE AID - Direct Access data handling utility
JS JDB STATUS - Using SPOOL Display and Search Facility (SDSF)
L LOG - Update SDE/SEL Data Base log with ISPF log data
P PANVALET - Browse, edit, and utilities for Panvalet data sets
X EXIT - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.

Lo and behold, we are back at the primary option panel. Just a few more things to do and we can quit.

As you can see, Utility functions are option 3. Take it on faith that listing utilities are utility option 4; and that hardcopy listings are produced as listing option 2. Type 3.4.2 in the option input area. This will jump option 3 (utilities), suboption 4 (listings), sub-suboption 2 (hardcopy). Press <enter>, and ...
Scenario Step 7: Printing listing files

--- HARDCOPY UTILITY ---

**OPTION**

PK - Print/punch and keep data set  
PD - Print/punch and delete data set

**DATA SET NAME**

VOLUME SERIAL  
DATA SET PASSWORD  

**SYSOUT CLASS**  
LOCAL PRINTER ID

**JOB STATEMENT INFORMATION:** (If not to local printer, verify before proceeding)

```plaintext
//GJZZPN JOB (SPONS,TEST,CCC), 'Z. Z. PITTS HARDCOPY',  
// MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,  
// TIME=003  
//=ROUTE PRINT PRT23
```

... we get to the harcopy utility parameter entry panel.
Scenario Step 7: Printing listing files

--- HARDCOPY UTILITY ---

**OPTION**: PK

PK - Print/punch and keep data set
PD - Print/punch and delete data set

**DATA SET NAME**: TS.SAMPLE.LIST
**VOLUME SERIAL**: (If not cataloged)
**DATA SET PASSWORD**: (If password protected PDS)

**SYSOUT CLASS**: A
**LOCAL PRINTER ID**: 

**JOB STATEMENT INFORMATION**: (If not to local printer, verify before proceeding)

```
//GZ2ZPN JOB (SPONS,TEST,CCC),'Z. Z. PETTS HARDCOPY',
// MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GZ2ZPN,
// TIME=003
// ROUTE PRINT PRT23
```

Type in PK for a selection option, then <NL> to the data set name parameter entry field. TS.SAMPLE.LIST is the name of the compile output file. Type it in and press <enter>.
Scenario Step 7: Printing listing files

---------- HARDCOPY UTILITY -----------

OPTION ===> PK

PK - Print/punch and keep data set
PD - Print/punch and delete data set

DATA SET NAME ===> IS.SAMPLE.LIST
VOLUME SERIAL ===> (If not cataloged)
DATA SET PASSWORD ===> (If password protected PDS)

SYSOUT CLASS ===> A
LOCAL PRINTER ID ===> 

JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)

==> //GJZPPN JOB (SPONS,TEST,CCC), 'Z. Z. PITTS HARDCOPY',
==> // MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZPP,
==> // TIME=003
==> /*ROUTE PRINT PRT23

The short message, JCL GENERATED, is displayed, indicating that a job control language file has been created and instructions that will print the data set have been put into it. The cursor will move to the beginning of the data set name parameter entry field.
Scenario Step 7: Printing listing files

--------------------------  HARDCOPY UTILITY  --------------------------  JCL GENERATED

OPTION ==> PK

PK - Print/punch and keep data set
PD - Print/punch and delete data set

DATA SET NAME ==> TS.ADDSUB_LIST

VOLUME SERIAL ==> (If not cataloged)
DATA SET PASSWORD ==> (If password protected PDS)

SYSOUT CLASS ==> A
LOCAL PRINTER ID ==> 

JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)

==> //GJZZPN JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY'.
==> // MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP.
==> // TIME=003
==> /*ROUTE PRINT PRT23

Type in the name of the ADDSUB subroutine and press <enter> . . .

3-102
Scenario Step 7: Printing listing files

--- HARDCOPY UTILITY --- JCL GENERATED

OPTION ==> PK

PK - Print/punch and keep data set
PD - Print/punch and delete data set
CANCEL - Exit without submitting job

Enter END command to submit job.

DATA SET NAME ==> TS.MULDIV.LIST
VOLUME SERIAL ==> (If not cataloged)
DATA SET PASSWORD ==> (If password protected PDS)

SYSOUT CLASS ==> A

JOB STATEMENT INFORMATION:
//GJZZPN JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',.
// MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,.  
// TIME=003
//ROUTE PRINT PRT23

... and the JCL GENERATED message remains. The JCL to print the ADDSUB
listing file has been added to the file created for SAMPLE. Move the cursor and
type MULDIV over top of ADDSUB; then press <enter> ...
Scenario Step 7: Printing listing files

------------- HARDCOPY UTILITY -------------- JCL GENERATED

OPTION ===> PK

PK - Print/punch and keep data set
PD - Print/punch and delete data set

DATA SET NAME ===> EXAMPLE.LINKLIST

VOLUME SERIAL ===> (If not cataloged)
DATA SET PASSWORD ===> (If password protected PDS)

SYSOUT CLASS ===> A
LOCAL PRINTER ID ===> 

JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)

====> //GUZPNN JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY',
====> // MSGLEVEL=*1.1,CLASS=A,MSGCLASS=A,NOTIFY=GUZPPN,
====> // TIME=003
====> //ROUTE PRINT PRT23

... and instructions to print the MULDIV listing file are added to the JCL file. Now move the cursor and add the linkage editor listing file to the list of files to be printed. Press <enter> and the JCL GENERATED message remains.
Press <end> and ...

3-104
Scenario Step 7: Printing listing files

--- HARDCOPY UTILITY --- JCL GENERATED ---

OPTION ===> PK

PK - Print/punch and keep data set
PD - Print/punch and delete data set

DATA SET NAME ===> EXAMPLE.LINKLIST.
VOLUME SERIAL ===> (If not cataloged)
DATA SET PASSWORD ===> (If password protected PDS)

SYSSUT CLASS ===> A
LOCAL PRINTER ID ===> 

JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)

====> //GJZZPN JOB (SPONS,TEST,CCC),'Z. Z. PITTS HARDCOPY'.
====> // MSGLEVEL=(1,1),CLASS=A,MSGCLASS=A,NOTIFY=GJZZP,
====> // TIME=003

JOB GJZZPN(JOB000162) SUBMITTED

... and a job submission message is displayed. Three asterisks remain at the bottom of the screen until we press <enter> to acknowledge. At that time ...
Scenario Step 8: Updating SDE log and exiting the SDE

<table>
<thead>
<tr>
<th>OPTION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DEFAULTS</td>
</tr>
<tr>
<td>1</td>
<td>BROWSE</td>
</tr>
<tr>
<td>2</td>
<td>EDIT</td>
</tr>
<tr>
<td>3</td>
<td>UTILITIES</td>
</tr>
<tr>
<td>4</td>
<td>COMPIL</td>
</tr>
<tr>
<td>5</td>
<td>LINK</td>
</tr>
<tr>
<td>6</td>
<td>TSO</td>
</tr>
<tr>
<td>7</td>
<td>TEST</td>
</tr>
<tr>
<td>8</td>
<td>NEWS/VIEWS</td>
</tr>
<tr>
<td>9</td>
<td>MGT INFO</td>
</tr>
<tr>
<td>10</td>
<td>MISC</td>
</tr>
<tr>
<td>F</td>
<td>FILE AID</td>
</tr>
<tr>
<td>J</td>
<td>JOB STATUS</td>
</tr>
<tr>
<td>L</td>
<td>LOG</td>
</tr>
<tr>
<td>P</td>
<td>PANVALET</td>
</tr>
<tr>
<td>X</td>
<td>EXIT</td>
</tr>
</tbody>
</table>

Enter END command to terminate ISPF.

---

...we are back at the primary option panel.

Type L to select the log update function. This copies our ISPF session log to an archive file so the SDE development team can see which options get the most use and which ones get the most errors. They will provide more help panels and/or redesign functions that generate a lot of error messages.

3-106
Scenario Step 8: Updating SDE log and exiting the SDE

------------------ FDS/SDE PRIMARY OPTION MENU VERSION 6.1 - ARCADE SUCCESSFUL ------------
OPTION [ ]

0 DEFAULTS - Specify terminal and user parameters
1 BROWSE - Display source data or output listings
2 EDIT - Create or change source data
3 UTILITIES - Perform utility functions (copy, allocate, rename, list)
4 COMPILE - Invoke language translators (Asm, Fort, Pascal, GESS)
5 LINK - Invoke linkage-editor (build load modules)
6 TSO - Enter TSO command or CLIST
7 TEST - Perform dialog testing
8 NEWS/VIEWS - Display news or enter comments about ISPF/PDF/SDE
9 MGT INFO - Use on-line management information systems
10 MISC - Miscellaneous Software Development Environment Features
F FILE AID - Direct Access data handling utility
JS JOB STATUS - Using SPOOL Display and Search Facility (SDSF)
L LOG - Update SDE/SEL Data Base log with ISPF log data
P PANVALET - Browse, edit, and utilities for Panvalet data sets
X EXIT - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.

In a minute or two, the message ARCHIVE SUCCESSFUL will be displayed in the short message area.
Scenario Step 8: Updating SDE log and exiting the SDE

---------- FDS/SDE PRIMARY OPTION MENU VERSION ----------

OPT 0 DEFAULTS - Specify terminal and user parameters
1 BROWSE   - Display source data or output listings
2 EDIT     - Create or change source data
3 UTILITIES- Perform utility functions (copy, allocate, rename, list)
4 COMPILE  - Invoke language translators (Asm, Fort, Pascal, GESS)
5 LINK     - Invoke linkage-editor (build load modules)
6 TSO      - Enter TSO command or CLIST
7 TEST     - Perform dialog testing
8 NEWS/VIEWS- Display news or enter comments about ISPF/PDF/SDE
9 MGT INFO- Use on-line management information systems
F FILE AID - Direct Access data handling utility
JS JOB STATUS- Using SPOOL Display and Search Facility (SDSF)
L LOG      - Update SDE/SEL Data Base log with ISPF log data
P PANVALET - Browse, edit, and utilities for Panvalet data sets
X EXIT     - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.

Type X as the option to exit. The screen will clear and soon ...

3-108
Scenario Step 8: Updating SDE log and exiting the SDE

the TSO READY acknowledgement will appear. Now, type in logoff and you will be off the system. Best of luck in your software development adventures aided by the SDE.
The technical papers, memorandums, and documents listed in this bibliography are organized into two groups. The first group is composed of documents issued by the Software Engineering Laboratory (SEL) during its research and development activities. The second group includes materials that were published elsewhere but pertain to SEL activities.

**SEL-ORIGINATED DOCUMENTS**

SEL-76-001, *Proceedings From the First Summer Software Engineering Workshop*, August 1976


SEL-77-003, *Structured FORTRAN Preprocessor (SFORT)*, B. Chu and D. S. Wilson, September 1977

SEL-77-004, *GSFC NAVPAK Design Specifications Languages Study*, P. A. Scheffer and C. E. Velez, October 1977


SEL-78-005, *Proceedings From the Third Summer Software Engineering Workshop*, September 1978


B-1


SEL-79-005, Proceedings From the Fourth Summer Software Engineering Workshop, November 1979


SEL-80-005, A Study of the Musa Reliability Model, A. M. Miller, November 1980

SEL-80-006, Proceedings From the Fifth Annual Software Engineering Workshop, November 1980


SEL-81-012, The Rayleigh Curve As a Model for Effort Distribution Over the Life of Medium Scale Software Systems, G. O. Picasso, December 1981

SEL-81-013, Proceedings From the Sixth Annual Software Engineering Workshop, December 1981

SEL-81-014, Automated Collection of Software Engineering Data in the Software Engineering Laboratory (SEL), A. L. Green, W. J. Decker, and F. E. McGarry, September 1981

SEL-81-101, Guide to Data Collection, V. E. Church, D. N. Card, F. E. McGarry, et al., August 1982

SEL-81-102, Software Engineering Laboratory (SEL) Data Base Organization and User's Guide Revision 1, P. Lo and D. Wyckoff, July 1983


SEL-82-003, Software Engineering Laboratory (SEL) Data Base Reporting Software User's Guide and System Description, P. Lo, September 1982

SEL-82-004, Collected Software Engineering Papers: Volume 1, July 1982

SEL-82-007, Proceedings From the Seventh Annual Software Engineering Workshop, December 1982

SEL-82-008, Evaluating Software Development by Analysis of Changes: The Data From the Software Engineering Laboratory, V. R. Basili and D. M. Weiss, December 1982

SEL-82-102, FORTRAN Static Source Code Analyzer Program (SAP) System Description (Revision 1), W. A. Taylor and W. J. Decker, April 1985

SEL-82-105, Glossary of Software Engineering Laboratory Terms, T. A. Babst, F. E. McGarry, and M. G. Rohleder, October 1983

SEL-82-306, Annotated Bibliography of Software Engineering Laboratory Literature, D. N. Card, Q. L. Jordan, and F. E. McGarry, November 1985


SEL-83-007, Proceedings From the Eighth Annual Software Engineering Workshop, November 1983

SEL-83-105, Software Engineering Laboratory (SEL) Data Base Retrieval System (DARES) System Description, P. Lo, W. J. Decker, and W. Miller, August 1984


SEL-84-003, Investigation of Specification Measures for the Software Engineering Laboratory (SEL), W. W. Agresti, V. E. Church, and F. E. McGarry, December 1984

SEL-84-004, Proceedings From the Ninth Annual Software Engineering Workshop, November 1984


SEL-86-002, General Object-Oriented Software Development, E. Seidewitz and M. Stark, July 1986


SEL-RELATED LITERATURE

Agresti, W. W., Definition of Specification Measures for the Software Engineering Laboratory, Computer Sciences Corporation, CSC/TM-84/6085, June 1984


Basili, V. R., "SEL Relationships for Programming Measurement and Estimation," University of Maryland, Technical Memorandum, October 1979


B-7

0253


Higher Order Software, Inc., TR-9, A Demonstration of AXES for NAVPAK, M. Hamilton and S. Zeldin, September 1977 (also designated SEL-77-005)


B-8

0253


---

1This article also appears in SEL-85-003, Collected Software Engineering Papers: Volume III, November 1985.

2This article also appears in SEL-83-003, Collected Software Engineering Papers: Volume II, November 1983.

3This article also appears in SEL-82-004, Collected Software Engineering Papers: Volume I, July 1982.
REFERENCES

1. IBM Corporation, Interactive System Productivity Facility/Program Development Facility, Version 2, Program Reference SC34-2139-0, March 1984
