

Order No. ST34-2085-0 (Formerly SC34-2085-0)
File No. S370/4300-39

Program Product

**Interactive
System Productivity Facility
Dialog Management Services
Examples**

IBM

Order No. ST34-2085-0 (Formerly SC34-2085-0)
File No. S370/4300-39

Program Product

**Interactive
System Productivity Facility
Dialog Management Services
Examples**

Program Number 5668-960

IBM

First Edition (March, 1982)

This edition applies to the Interactive System Productivity Facility (ISPF) Program Product, 5668-960, for use with VM/SP or OS/VS2 MVS Release 3.8 and to all subsequent releases until otherwise indicated by Technical Newsletters.

Information pertaining to VM/SP is for planning purposes until the availability of ISPF for VM/SP.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

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

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Systems Publications, Dept Z59, Bldg 931, P.O. Box 390, Poughkeepsie, New York U.S.A 12602. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

PREFACE

The Interactive System Productivity Facility (ISPF) is a dialog manager for interactive applications. It provides control and services for dialogs executed in the MVS and VM/SP environments.

This publication, which supplements ISPF Dialog Management Services (SC34-2088), provides examples of how dialogs invoke ISPF services. The examples appear in the context of implementations of an "employee records" application and a "telephone book" application. Each application includes a description of the application and listings of dialog components (dialog functions, selection and data entry panel definitions, messages, and file tailoring skeletons) required to process the application under ISPF.

Example dialog functions are provided for both applications in the following command procedures and programming languages:

- Command Procedures
 - CLIST
 - EXEC2
- Programming Languages
 - COBOL
 - FORTRAN
 - PL/I

It is assumed that the reader is an application or systems programmer, engaged in program development, who is familiar with one or more of the command procedures or programming languages listed above.

ISPF Dialog Management Services, SC34-2088, is a prerequisite publication. It contains a detailed description of ISPF services and related information required to develop an interactive application that runs under ISPF.

The following publications contain additional information related to the indicated environment:

<u>MVS</u>	<u>VM/SP</u>	
SC34-2084	SC34-2083	<u>ISPF Installation and Customization</u> provides information needed to install ISPF and the ISPF/Program Development Facility (PDF) and to custom tailor these program products for a particular installation.

SC34-2089 SC34-2090 ISPF/PDF Reference describes how to use the program development facility. PDF includes aids for ISPF dialog creation and testing.

TERMINOLOGY

In this publication the following terms are used to bridge the differences in terminology between the MVS and VM/SP environments:

- Library - A partitioned data set in the MVS environment, or a MACLIB in the VM/SP environment.
- File - A sequential data set in the MVS environment, or a sequential CMS file in the VM/SP environment.
- Command Procedure - A CLIST in the MVS environment, or an EXEC (coded in EXEC2 language) in the VM/SP environment.

CONTENTS

Example 1. Employee Records Application	1
Overview	1
Dialog Components	1
Invocation	2
Description of Processing	2
Dialog Components Applicable to All Implementations	3
First Data Entry Panel EMPLA	4
Panel Definition	4
Panel Display	4
Second Data Entry Panel EMPLB	5
Panel Definition	5
Panel Display	6
Message Definitions EMPX21	7
CLIST Implementation	8
CLIST - Primary Option Menu EMPL	8
Panel Definition	8
Panel Display	8
CLIST - Function EMPLFN	9
EXEC2 Implementation	11
EXEC2 - Primary Option Menu EMPL	11
Panel Definition	11
Panel Display	12
EXEC2 - Function EMPLFN	12
COBOL Implementation	15
COBOL - Primary Option Menu EMPL	15
Panel Definition	15
Panel Display	15
COBOL - Function EMPLFN	16
FORTRAN Implementation	20
FORTRAN - Primary Option Menu EMPL	20
Panel Definition	20
Panel Display	21
FORTRAN - Function EMPLFN	21
PL/I Implementation	25
PL/I - Primary Option Menu Definition EMPL	25
Panel Definition	25
Panel Display	25
PL/I - Function EMPLFN	26
Example 2. Telephone Book Application	29
Overview	29
Dialog Components	30
Invocation	31
Description of Processing	33
Main Function Routine TELEBOOK	33
Combined Selection and Data Entry Panel CPANEL1, FPANEL1, or TPANEL1	33
Subfunction Routine TELESRVC	34

Dialog Components Applicable to All Implementations	34
Table Display Panel TPANEL2	35
Panel Definition	35
Panel Display	35
Information Display Panel TPANEL3	36
Panel Definition	36
Panel Display	36
Message Definitions TELB01	37
Message Definitions TELB02	37
File Tailoring Skeleton TELBOOK	38
CLIST Implementation	38
CLIST - Combined Selection and Data Entry Panel CPANEL1	38
Panel Definition	38
Panel Display	40
CLIST - Function TELEBOOK	40
CLIST - Function TELESRVC	42
EXEC2 Implementation	48
EXEC2 - Combined Selection and Data Entry Panel CPANEL1	48
Panel Definition	48
Panel Display	50
EXEC2 - Function TELEBOOK	50
EXEC2 - Function TELESRVC	52
COBOL Implementation	60
COBOL - Combined Selection and Data Entry Panel TPANEL1	60
Panel Definition	60
Panel Display	62
COBOL - Function TELEBOOK	62
COBOL - Function TELESRVC	65
FORTRAN Implementation	73
FORTRAN - Combined selection and Data Entry Panel FPANEL1	74
Panel Definition	74
Panel Display	76
FORTRAN - Function TELEBOOK	76
FORTRAN - Function TELESRVC	78
PL/I Implementation	86
PL/I - Combined Selection and Data Entry Panel TPANEL1	87
Panel Definition	87
Panel Display	88
PL/I - Function TELEBOOK	89
PL/I - Function TELESRVC	91
 Index	 101

FIGURES

1. Organization of Employee Records Application 2
2. Organization of Telephone Book Application 32

EXAMPLE 1. EMPLOYEE RECORDS APPLICATION

This section describes implementations of an "employee records" application dialog. The implementations include dialog functions coded in:

- CLIST
- EXEC2
- COBOL
- FORTRAN
- PL/I

OVERVIEW

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

DIALOG COMPONENTS

Dialog components that implement the employee records application are:

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

INVOCATION

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

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

DESCRIPTION OF PROCESSING

The organization of the dialog is shown in Figure 1.

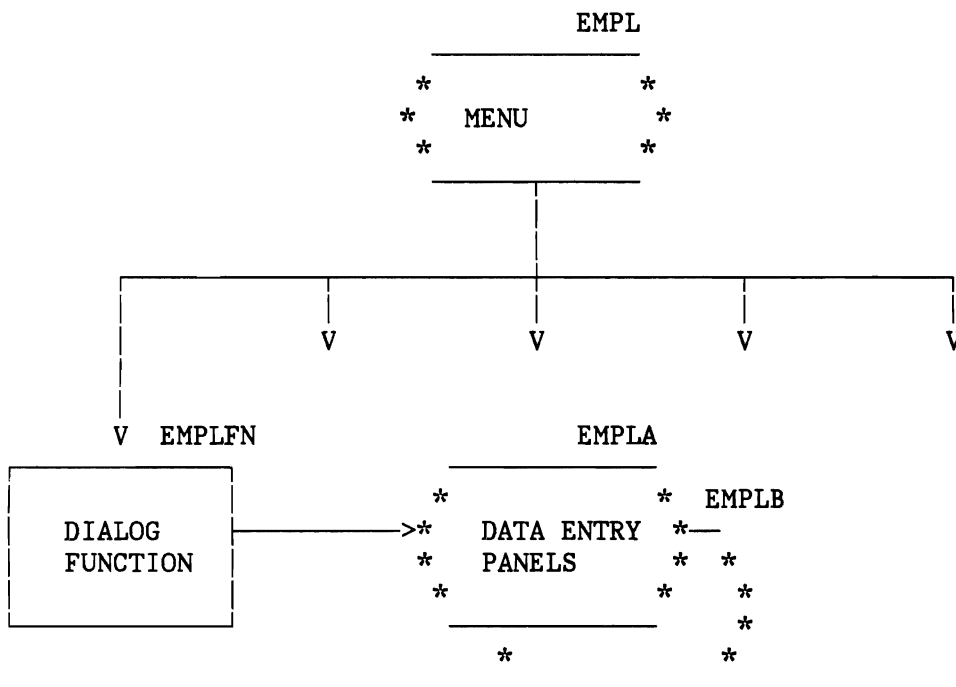


Figure 1. Organization of Employee Records Application

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

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

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

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

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

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

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

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

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

DIALOG COMPONENTS APPLICABLE TO ALL IMPLEMENTATIONS

Dialog components applicable to all of the implementations are:

- First and second data entry panels (EMPLA and EMPLB)
- Messages used by the application (EMPX21)

These components are shown in the following sections.

FIRST DATA ENTRY PANEL EMPLA

Definition and display of data entry panel EMPLA are shown below.

Panel Definition

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

Panel Display

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

SECOND DATA ENTRY PANEL EMPLB

Definition and display of data entry panel EMPLB are shown below.

Panel Definition

```
%----- EMPLOYEE RECORDS -----
%COMMAND ==> _ZCMD
+
%   EMPLOYEE SERIAL: &EMP SER
+
+   TYPE OF CHANGE%==> _TYPECHG +   (NEW, UPDATE, OR DELETE)
+
+   EMPLOYEE NAME:
+     LAST   %==> _LNAME           +
+     FIRST  %==> _FNAME           +
+     INITIAL%==> _I+&RBL
+
+   HOME ADDRESS:
+     LINE 1 %==> _ADDR1           +
+     LINE 2 %==> _ADDR2           +
+     LINE 3 %==> _ADDR3           +
+     LINE 4 %==> _ADDR4           +
+
+   HOME PHONE:
+     AREA CODE   %==> _PHA+
+     LOCAL NUMBER%==> _PHNUM   +
+
)INIT
  .CURSOR = TYPECHG
  IF (&PHA = ' ')
    &PHA = 914
  &TYPECHG = TRANS(&TYPECHG N,NEW U,UPDATE D,DELETE)
)PROC
  &TYPECHG = TRUNC (&TYPECHG,1)
  VER (&TYPECHG,LIST,N,U,D,MSG=EMPX210)
  IF (&TYPECHG = N)
    IF (&CHKTYPE ^= N)
      .MSG = EMPX211
  IF (&TYPECHG ^= N)
    IF (&CHKTYPE = N)
      .MSG = EMPX212
  VER (&LNAME,ALPHA)
  VER (&FNAME,ALPHA)
  VER (&I,ALPHA)
  VER (&PHA,NUM)
  VER (&PHNUM,PICT,'NNN-NNNN')
```

```
IF (&TYPECHG = N,U)
  VER (&LNAME, NONBLANK, MSG=EMPX214)
  VER (&FNAME, NONBLANK, MSG=EMPX213)
  VER (&ADDR1, NONBLANK, MSG=EMPX215)
  VER (&ADDR2, NONBLANK, MSG=EMPX215)
  VER (&ADDR3, NONBLANK, MSG=EMPX215)
)END
```

Panel Display

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

EMPLOYEE SERIAL: 121212

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

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

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

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

MESSAGE DEFINITIONS EMPX21

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

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

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

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

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

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

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

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

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

CLIST IMPLEMENTATION

The components listed under "Dialog Components Applicable to All Implementations," and the following are required for the CLIST implementation of the dialog:

- Primary option menu definition EMPL
- Function EMPLFN

CLIST - PRIMARY OPTION MENU EMPL

Definition and display of primary option menu EMPL are shown below.

Panel Definition

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

Panel Display

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

SELECT OPTION ==> 1_

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

ENTER END COMMAND TO TERMINATE.
```

CLIST - FUNCTION EMPLFN

```
PROC 0
CONTROL MAIN

/* SET CONTROL TO RETURN TO CLIST FOR ALL ISPF SERVICE ERRORS
ISPEXEC CONTROL ERRORS RETURN

/* INITIALIZE LOOP CONTROL AND MESSAGE VARIABLES
SET &STATE = 1
SET &MSG =

/* OPEN EMPLOYEE TABLE
ISPEXEC TBOPEN EMPLTBL

/* IF EMPLOYEE TABLE DOES NOT EXIST, CREATE IT
IF &LASTCC = 0 THEN +
  ISPEXEC TBCREATE EMPLTBL KEYS(EMP SER) +
  NAMES(LNAME FNAME I ADDR1 ADDR2 ADDR3 ADDR4 PHA PHNUM)

/* DISPLAY DATA ENTRY PANEL TO OBTAIN EMPLOYEE NUMBER FOR PROCESSING
/* UNTIL END OR RETURN COMMAND RECEIVED IN RESPONSE TO DISPLAY
/* (&STATE = 4)
DO WHILE &STATE = 4
  ISPEXEC DISPLAY PANEL(EMPLA) MSG(&MSG)

/* IF RESPONSE IS END OR RETURN SET &STATE = 4 AND GO CHECK FOR MESSAGE
IF &LASTCC=8 THEN SET &STATE =4

/* ELSE, PROCESS EMPLOYEE NUMBER ENTERED
ELSE DO

/* RESET CONTROL VARIABLE (&STATE) AND MESSAGE VARIABLE (&MSG)
SET &MSG =
SET &STATE =2

/* GET ROW SPECIFIED BY EMPLOYEE NUMBER FROM THE TABLE
ISPEXEC TBGET EMPLTBL

/* IF THE ROW WAS FOUND SET THE TYPE OF CHANGE TO UPDATE (U)
/* AND SKIP INITIALIZATION PROCESSING FOR A NEW ROW
IF &LASTCC=0 THEN SET &TYPECHG = U
```

```

/* ELSE SET TYPE OF CHANGE TO NEW (N) AND INITIALIZE THE TABLE VARIABLES
  ELSE DO
    SET  &TYPECHG = N
    SET  &LNAME   =
    SET  &FNAME   =
    SET  &I       =
    SET  &ADDR1   =
    SET  &ADDR2   =
    SET  &ADDR3   =
    SET  &ADDR4   =
    SET  &PHA     =
    SET  &PHNUM   =
  END

/* SAVE TYPE OF CHANGE FOR VALIDITY CHECKING ON DATA ENTRY PANEL
  SET &CHKTYPE = &TYPECHG

/* DISPLAY DATA ENTRY PANEL TO OBTAIN EMPLOYEE DATA
  ISPEXEC  DISPLAY  PANEL(EMPLB)

/* IF END OR RETURN NOT RECEIVED IN RESPONSE ROW PROCESSING
  IF  &LASTCC  ^= 8  THEN DO

/* IF TYPE OF CHANGE IS NEW, ADD NEW ROW TO TABLE AND
/* SET MESSAGE TO INDICATE EMPLOYEE ADDED TO TABLE
  IF  &TYPECHG = N  THEN DO
    ISPEXEC  TBADD  EMPLTBL
    SET  &MSG  = EMPX217
  END

/* IF TYPE OF CHANGE IS UPDATE, PUT NEW EMPLOYEE DATA TO TABLE TO
/* EXISTING ROW FOR THAT EMPLOYEE AND SET THE ROW UPDATED MESSAGE
  ELSE DO
    IF  &TYPECHG = U  THEN DO
      ISPEXEC  TBPUT  EMPLTBL
      SET  &MSG  = EMPX218
    END

/* IF USER CHANGED TYPE OF CHANGE ON DATA ENTRY PANEL TO OTHER THAN
/* NEW (N) OR UPDATE (U), DELETE THE ROW FROM THE TABLE AND SET THE
/* ROW DELETED MESSAGE
  ELSE DO
    ISPEXEC  TBDELETE  EMPLTBL
    SET  &MSG  = EMPX219
  END

  END
  END
  END

/* LOG ANY OUTSTANDING MESSAGE AND GO CHECK FOR ANOTHER EMPLOYEE NUMBER
  IF  &MSG  ^=      THEN  ISPEXEC  LOG  MSG(&MSG)

```

```
/* END OF DO WHILE LOOP
END
```

```
/* USER HAS RESPONDED WITH END OR RETURN TO EMPLOYEE NUMBER OR
/* EMPLOYEE DATA PANEL, WRITE THE PROCESSED EMPLOYEE TABLE TO
/* DASD AND CLOSE IT
ISPEXEC TBCLOSE EMPLTBL
```

```
/* RETURN TO THE SELECTION MENU FOR THE EMPLOYEE APPLICATION (EMPL)
EXIT CODE (0)
END
```

EXEC2 IMPLEMENTATION

In addition to the components listed under "Dialog Components Applicable to All Implementations," the following components are required for the EXEC2 implementation of the dialog:

- Primary option menu definition EMPL
- Function EMPLFN

EXEC2 - PRIMARY OPTION MENU EMPL

Definition of primary option menu EMPL is shown below, followed by the panel as displayed.

Panel Definition

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

Panel Display

```
----- EMPLOYEE RECORDS -----  
  
SELECT OPTION ==> 1_  
  
 1  MODIFY      - ADD, UPDATE, OR DELETE EMPLOYEE RECORDS  
 2  (FUTURE)    - FUNCTION NOT YET AVAILABLE  
 3  (FUTURE)    - FUNCTION NOT YET AVAILABLE  
 4  (FUTURE)    - FUNCTION NOT YET AVAILABLE  
 5  (FUTURE)    - FUNCTION NOT YET AVAILABLE  
  
ENTER END COMMAND TO TERMINATE.
```

EXEC2 - FUNCTION EMPLFN

```
&TRACE  
* SET UP EXEC2 AND SUBCOMMAND ENVIRONMENT FOR ISPF DIALOG FUNCTION  
&PRESUME &SUBCOMMAND ISPEXEC  
  
* SET UP CONTROL FOR RETURN TO FUNCTION ON ISPF SERVICE ERRORS  
* 12 OR GREATER.  
ISPEXEC CONTROL ERRORS RETURN  
  
* INITIAL LOOP CONTROL VARIABLE AND MESSAGE VARIABLE  
&STATE = 1  
&MSG = &BLANK  
  
* SET UP TABLE NAME VARIABLES NAME LIST FOR TBCREATE  
&NAME1 = &CONCAT OF ( &BLANK LNAME &BLANK FNAME &BLANK I &BLANK ADDR1  
&NAME2 = &CONCAT OF &BLANK ADDR2 &BLANK ADDR3 &BLANK ADDR4 &BLANK  
&NAME3 = &CONCAT OF PHA &BLANK PHNUM &BLANK )  
&NAMES = &CONCAT OF &NAME1 &NAME2 &NAME3  
  
* OPEN EMPLOYEE TABLE  
ISPEXEC TBOPEN EMPLTBL  
  
* IF TABLE DOESN'T EXIST, CREATE NEW ONE  
&IF &RC = 0 &GOTO -ESISTE  
ISPEXEC TBCREATE EMPLTBL KEYS ( EMPSER ) NAMES &NAMES REPLACE  
  
-ESISTE  
* CONTINUE TO DISPLAY DATE ENTRY PANEL TO OBTAIN EMPLOYEE NUMBER FROM  
* USER UNTIL THE USER RESPONDS WITH END OR RETURN.  
&LOOP -FLOOP WHILE &STATE ^= 4  
ISPEXEC DISPLAY PANEL(EMPLA) MSG ( &MSG )
```

```

* IF USER RESPONDS TO PANEL WITH END OR RETURN, GO SET LOOP CONTROL
* VARIABLE (&STATE) TO 4.
  &IF &RC = 8 &GOTO -STATE4

* OTHERWISE REINITIALIZE THE LOOP CONTROL AND MESSAGE VARIABLES AND
* GET THE ROW FOR THE SPECIFIED EMPLOYEE NUMBER FROM THE TABLE
  &MSG = &BLANK
  &STATE = 2
  ISPEXEC TBGET EMPLTBL

-FIRPAN
* IF THE ROW FOR THE SPECIFIED EMPLOYEE NUMBER IS FOUND GO SET UP
* AN UPDATE TYPE OF CHANGE TO THE TABLE
  &IF &RC = 0 &GOTO -UPDATE2

* OTHERWISE, SET THE TYPE OF CHANGE TO NEW ROW, INITIALIZE THE TABLE
* VARIABLES TO BLANK, AND GO GET THE ROW DATA.
  &TYPECHG = N
  &LNAME = &BLANK
  &FNAME = &BLANK
  &I = &BLANK
  &ADDR1 = &BLANK
  &ADDR2 = &BLANK
  &ADDR3 = &BLANK
  &ADDR4 = &BLANK
  &PHA = &BLANK
  &PHNUM = &BLANK
  &GOTO -NEW

-UPDATE2
* SET THE TYPE OF CHANGE FOR THE ROW TO UPDATE
  &TYPECHG = U

-NEW
* SAVE TYPE OF CHANGE FOR VALIDITY CHECKING IN DATA ENTRY PANEL
  &CHKTYPE = &TYPECHG

* DISPLAY DATA ENTRY PANEL TO OBTAIN NEW OR UPDATED EMPLOYEE DATA
  ISPEXEC DISPLAY PANEL(EMPLB)

* IF RESPONSE TO REQUEST FOR EMPLOYEE DATA IS END OR RETURN, GO
* CHECK FOR ANOTHER EMPLOYEE NUMBER TO BE PROCESSED
  &IF &RC = 8 &GOTO -LATER

* ELSE, IF THE TYPE OF CHANGE IS NOT NEW, GO CHECK FOR UPDATE
  &IF &TYPECHG ^= N &GOTO -NOTNEW

* OTHERWISE, ADD ROW FOR NEW EMPLOYEE TO TABLE, SET ROW ADDED
* MESSAGE, AND GO LOG MESSAGE
  ISPEXEC TBADD EMPLTBL
  &MSG = EMPX217
  &GOTO -LOGMSG

```

-NOTNEW

* IF TYPE OF CHANGE IS NOT UPDATE, GO PROCESS DELETE FUNCTION
&IF &TYPECHG ^= U &GOTO -UPDATE1

* FOR UPDATED ROW, PUT UPDATED EMPLOYEE DATA TO THE TABLE, SET

* ROW UPDATED MESSAGE, AND GO LOG MESSAGE

ISPEXEC TBPUR EMPLTBL

&MSG = EMPX218

&GOTO -LOGMSG

-UPDATE1

* USER CHANGED TYPE OF CHANGE ON EMPLOYEE DATA PANEL TO DELETE, SO

* DELETE SPECIFIED EMPLOYEE ROW FROM TABLE, SET DELETE MESSAGE,

* AND GO LOG MESSAGE

ISPEXEC TBDELETE EMPLTBL

&MSG = EMPX219

&GOTO -LOGMSG

-STATE4

* SET LOOP CONTROL VARIABLE TO 4 (END OR RETURN RECEIVED)

* TO THE TOP OF THE LOOP

&STATE = 4

-LOGMSG

* IF THERE IS AN OUTSTANDING MESSAGE LOG IT

&IF &MSG = &BLANK &GOTO -LATER

ISPEXEC LOG MSG (&MSG)

-LATER

* GO CHECK FOR ANOTHER EMPLOYEE NUMBER FOR PROCESSING

&GOTO -ESISTE

-FLOOP

* WRITE THE EMPLOYEE TABLE TO DASD AND CLOSE IT

ISPEXEC TBCLOSE EMPLTBL

* RETURN TO THE EMPLOYEE APPLICATION SELECTION MENU

&EXIT &RC

COBOL IMPLEMENTATION

The components listed under "Dialog Components Applicable to All Implementations," and the following are required for the COBOL implementation of the dialog:

- Primary option menu definition EMPL
- Function EMPLFN

COBOL - PRIMARY OPTION MENU EMPL

Definition and display of primary option menu EMPL are shown below.

Panel Definition

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

Panel Display

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

SELECT OPTION ==> 1_

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

ENTER END COMMAND TO TERMINATE.
```


COBOL - FUNCTION EMPLFN

```
000001 ID DIVISION.
000002 PROGRAM-ID.      EMPLFN
000004 ENVIRONMENT DIVISION.
000005 DATA DIVISION.
000006 WORKING-STORAGE SECTION.
000007
000008* DEFINE STORAGE FOR DIALOG VARIABLES.
000009 01 EMPSER PIC X(6) VALUE SPACES.
000010 01 FNAME PIC X(16) VALUE SPACES.
000011 01 LNAME PIC X(16) VALUE SPACES.
000012 01 I PIC X VALUE SPACE.
000013 01 ADDR1 PIC X(40) VALUE SPACES.
000014 01 ADDR2 PIC X(40) VALUE SPACES.
000015 01 ADDR3 PIC X(40) VALUE SPACES.
000016 01 ADDR4 PIC X(40) VALUE SPACES.
000017 01 PHA PIC X(3) VALUE SPACES.
000018 01 PHNUM PIC X(8) VALUE SPACES.
000019 01 MSG PIC X(8) VALUE SPACES.
000020 01 TYPECHG PIC X VALUE SPACE.
000021 01 CHKTYPE PIC X VALUE SPACE.
000022
000023* DEFINE AND INITIALIZE LOOP CONTROL VARIABLE.
000024 01 STATE PIC X VALUE "1".
000025
000026* DEFINE LENGTHS FOR DIALOG VARIABLES.
000027 01 LEMP SER PIC 9(6) VALUE 6 COMP.
000028 01 LFNAME PIC 9(6) VALUE 16 COMP.
000029 01 LLNAME PIC 9(6) VALUE 16 COMP.
000030 01 LI PIC 9(6) VALUE 1 COMP.
000031 01 LADDR1 PIC 9(6) VALUE 40 COMP.
000032 01 LADDR2 PIC 9(6) VALUE 40 COMP.
000033 01 LADDR3 PIC 9(6) VALUE 40 COMP.
000034 01 LADDR4 PIC 9(6) VALUE 40 COMP.
000035 01 LPHA PIC 9(6) VALUE 3 COMP.
000036 01 LPHNUM PIC 9(6) VALUE 8 COMP.
000037 01 LTYPECH PIC 9(6) VALUE 1 COMP.
000038 01 LCHK TYP PIC 9(6) VALUE 1 COMP.
000039
000040* DEFINE DIALOG VARIABLE NAMES FOR DIALOG SERVICE CALLS.
000041 01 NEMP SER PIC X(8) VALUE "(EMP SER)".
000042 01 NFNAME PIC X(7) VALUE "(FNAME)".
000043 01 NLNAME PIC X(7) VALUE "(LNAME)".
000044 01 NI PIC X(3) VALUE "(I)".
000045 01 NADDR1 PIC X(7) VALUE "(ADDR1)".
000046 01 NADDR2 PIC X(7) VALUE "(ADDR2)".
000047 01 NADDR3 PIC X(7) VALUE "(ADDR3)".
000048 01 NADDR4 PIC X(7) VALUE "(ADDR4)".
000049 01 NPHA PIC X(5) VALUE "(PHA)".
000050 01 NPHNUM PIC X(7) VALUE "(PHNUM)".
000051 01 NTYPECH PIC X(9) VALUE "(TYPECHG)".
000052 01 NCHK TYP PIC X(9) VALUE "(CHKTYPE)".
```

```

000053
000054* DEFINE DATA ENTRY PANEL NAMES.
000055 01  EMPLA  PIC  X(8)  VALUE "EMPLA  ".
000056 01  EMPLB  PIC  X(8)  VALUE "EMPLB  ".
000057
000058* DEFINE EMPLOYEE TABLE NAME.
000059 01  EMPLTBL PIC  X(8)  VALUE "EMPLTBL ".
000060
000061* DEFINE KEY VARIABLES PARAMETER FOR TABLE CREATE.
000062 01  TABKEY  PIC  X(8)  VALUE "(EMP SER)".
000063
000064* DEFINE NAME VARIABLES PARAMETER FOR TABLE CREATE.
000065 01  TABVARS PIC  X(49) VALUE
000066      "(LNAME FNAME I ADDR1 ADDR2 ADDR3 ADDR4 PHA PHNUM)".
000067
000068      EJECT
000069* DEFINE DATA TYPE PARAMETER FOR VARIABLE DEFINE SERVICE CALL.
000070 01  CHAR    PIC  X(8)  VALUE "CHAR  ".
000071
000072* DEFINE DIALOG SERVICE TYPES FOR CALLS TO ISPF.
000073 01  DISPLAYE PIC  X(8)  VALUE "DISPLAY ".
000074 01  LOG      PIC  X(8)  VALUE "LOG     ".
000075 01  TBADD    PIC  X(8)  VALUE "TBADD   ".
000076 01  TBCLOSE  PIC  X(8)  VALUE "TBCLOSE ".
000077 01  TBCREATE PIC  X(8)  VALUE "TBCREATE".
000078 01  TBDELETE PIC  X(8)  VALUE "TBDELETE".
000079 01  TBGET    PIC  X(8)  VALUE "TBGET   ".
000080 01  TBOPEN   PIC  X(8)  VALUE "TBOPEN  ".
000081 01  TBPUT    PIC  X(8)  VALUE "TBPUT   ".
000082 01  VDEFINE  PIC  X(8)  VALUE "VDEFINE ".
000083 01  VRESET  PIC  X(8)  VALUE "VRESET  ".
000084
000085
000086
000087 PROCEDURE DIVISION.
000088* CALL VDEFINE TO DEFINE DIALOG FUNCTION VARIABLES TO ISPF.
000089      CALL "ISPLINK" USING VDEFINE
000090          NEMP SER EMP SER CHAR  LEMP SER
000091      CALL "ISPLINK" USING VDEFINE
000092          NFNAME FNAME CHAR  LFNAME
000093      CALL "ISPLINK" USING VDEFINE
000094          NLNAME LNAME CHAR  LLNAME
000095      CALL "ISPLINK" USING VDEFINE
000096          NI      I      CHAR  LI
000097      CALL "ISPLINK" USING VDEFINE
000098          NADDR1 ADDR1 CHAR  LADDR1
000099      CALL "ISPLINK" USING VDEFINE
000100          NADDR2 ADDR2 CHAR  LADDR2
000101      CALL "ISPLINK" USING VDEFINE
000102          NADDR3 ADDR3 CHAR  LADDR3
000103      CALL "ISPLINK" USING VDEFINE
000104          NADDR4 ADDR4 CHAR  LADDR4

```

```

000105     CALL "ISPLINK" USING VDEFINE
000106         NPHA     PHA     CHAR   LPHA
000107     CALL "ISPLINK" USING VDEFINE
000108         NPHNUM    PHNUM    CHAR   LPHNUM
000109     CALL "ISPLINK" USING VDEFINE
000110         NTYPECH   TYPECHG  CHAR   LTYPECH
000111     CALL "ISPLINK" USING VDEFINE
000112         NCHKTYP   CHKTYPE  CHAR   LCHKTYP
000113
000114*  INITIALIZE MESSAGE DIALOG VARIABLE TO BLANK.
000115         MOVE SPACES TO MSG
000116
000117*  OPEN EMPLOYEE TABLE.
000118         CALL "ISPLINK" USING TBOPEN  EMPLTBL
000119
000120*  IF EMPLOYEE TABLE DOES NOT EXIST CREATE IT.
000121         IF RETURN-CODE NOT = 0
000122             CALL "ISPLINK" USING TBCREATE  EMPLTBL  TABKEY  TABVARS
000123
000124*  GO PERFORM MAIN-LINE ROUTINE TO ADD, UPDATE, OR DELETE ROWS IN
000125*  THE EMPLOYEE TABLE UNTIL THE USER RESPONDS WITH END OR RETURN
000126*  INSTEAD OF AN EMPLOYEE NUMBER.
000127         PERFORM MAIN-LINE UNTIL STATE = 4
000128
000129*  WHEN THE USER HAS COMPLETED TABLE PROCESSING WRITE THE EMPLOYEE
000130*  TABLE TO DASD AND CLOSE IT.
000131         CALL "ISPLINK" USING TBCLOSE  EMPLTBL
000132
000133*  RESET (REMOVE ADDRESSABILITY) ALL FUNCTION DIALOG VARIABLES.
000134         CALL "ISPLINK" USING VRESET
000135
000136*  RETURN TO THE APPLICATION SELECTION MENU.
000137         MOVE 0 TO RETURN-CODE
000138         GOBACK.
000139         EJECT
000140
000141*  MAIN-LINE ROUTINE FOR OBTAIN EMPLOYEE DATA AND PROCESSING
000142*  ROWS IN THE EMPLOYEE TABLE.
000143  MAIN-LINE.
000144
000145*  DISPLAY THE DATA ENTRY PANEL EMPLA TO OBTAIN AN EMPLOYEE
000146*  NUMBER FOR PROCESSING
000147         CALL "ISPLINK" USING DISPLAY  EMPLA  MSG
000148
000149*  IF END OR RETURN RECEIVED FROM PANEL SET LOOP CONTROL TO
000150*  4 AND SKIP TABLE ROW PROCESSING.
000151         IF RETURN-CODE = 8
000152             MOVE 4 TO STATE
000153
000154*  ELSE RESET MESSAGE VARIABLE TO BLANKS, SET LOOP CONTROL TO
000155*  2, AND CONTINUE PROCESSING ROW.

```

```

000156     ELSE
000157         MOVE SPACES TO MSG
000158         MOVE 2 TO STATE
000159
000160* GO PERFORM ROUTINE TO GET SPECIFIED ROW FROM THE TABLE AND
000161*   SET TYPE OF CHANGE TO UPDATE IF IT EXIST OR TO NEW IF THE
000162*   ROW DOES NOT EXIST.
000163         PERFORM TBGET-EMPLTBL
000164
000165* SAVE THE TYPE OF CHANGE FOR VALIDATION ON EMPLOYEE INFORMATION
000166*   DATA ENTRY PANEL.
000167         MOVE TYPECHG TO CHKTYPE
000168
000169* DISPLAY DATA ENTRY PANEL EMPLB TO GET EMPLOYEE INFORMATION.
000170         CALL "ISPLINK" USING DISPLAY EMPLB
000171
000172* IF END OR RETURN NOT RECEIVED FROM DATA ENTRY PANEL, PROCESS
000173*   ROW IN TABLE WITH EMPLOYEE INFORMATION RECEIVED.
000174         IF RETURN-CODE NOT = 8
000175
000176* IF TYPE OF CHANGE REQUESTED WAS NEW, ADD NEW ROW TO THE TABLE
000177*   AND SET MESSAGE TO ROW ADDED.
000178         IF TYPECHG = "N"
000179             CALL "ISPLINK" USING TBADD EMPLTBL
000180             MOVE "EMPX217 " TO MSG
000181
000182* ELSE IF TYPE OF CHANGE REQUEST WAS UPDATE, UPDATE ROW WITH
000183*   NEW EMPLOYEE INFORMATION AND SET MESSAGE TO ROW UPDATED.
000184         ELSE IF TYPECHG = "U"
000185             CALL "ISPLINK" USING TBPUT EMPLTBL
000186             MOVE "EMPX218 " TO MSG
000187
000188* OTHERWISE, USER REQUESTED DELETE FUNCTION ON EMPLOYEE
000189*   INFORMATION DATA ENTRY PANEL. DELETE SPECIFIED ROW FROM
000190*   THE TABLE AND SET MESSAGE TO ROW DELETED.
000191         ELSE
000192             CALL "ISPLINK" USING TBDELETE EMPLTBL
000193             MOVE "EMPX219 " TO MSG.
000194
000195* IF MESSAGE VARIABLE IS NOT BLANK, WRITE THE MESSAGE TO THE LOG.
000196         IF MSG NOT = SPACES
000197             CALL "ISPLINK" USING LOG MSG.
000198
000199* UPON COMPLETION OF ROW PROCESSING, OR IF END OR RETURN ENTERED
000200*   RETURN TO CHECK IF ADDITIONAL EMPLOYEES ARE TO BE PROCESSED.
000201
000202
000203* ROUTINE TO GET EMPLOYEE ROW FROM TABLE AND SET TYPE OF
000204*   ROW PROCESSING (TYPE OF CHANGE) TO BE PERFORMED.
000205 TBGET-EMPLTBL.
000206
000207* GET ROW FOR SPECIFIED EMPLOYEE NUMBER FROM TABLE.
000208         CALL "ISPLINK" USING TBGET EMPLTBL

```

```

000209
000210* IF ROW FOR EMPLOYEE NUMBER FOUND SET TYPE OF CHANGE TO UPDATE.
000211     IF RETURN-CODE = 0
000212         MOVE "U" TO TYPECHG
000213
000214* ELSE, SET TYPE OF CHANGE TO NEW AND INITIALIZE EMPLOYEE
000215*     VARIABLES TO BLANKS.
000216     ELSE
000217         MOVE "N" TO TYPECHG
000218         MOVE SPACES TO LNAME
000219         MOVE SPACES TO FNAME
000220         MOVE SPACES TO I
000221         MOVE SPACES TO ADDR1
000222         MOVE SPACES TO ADDR2
000223         MOVE SPACES TO ADDR3
000224         MOVE SPACES TO ADDR4
000230         MOVE SPACES TO PHA
000300         MOVE SPACES TO PHNUM.

```

FORTRAN IMPLEMENTATION

The components listed under "Dialog Components Applicable to All Implementations," and the following are required for the FORTRAN implementation of the dialog:

- Primary option menu definition EMPL
- Function EMPLFN

FORTRAN - PRIMARY OPTION MENU EMPL

Definition and display of primary option menu EMPL are shown below.

Panel Definition

```

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

```

Panel Display

```
----- EMPLOYEE RECORDS -----  
  
SELECT OPTION ==> 1_  
  
1  MODIFY      - ADD, UPDATE, OR DELETE EMPLOYEE RECORDS  
2  (FUTURE)   - FUNCTION NOT YET AVAILABLE  
3  (FUTURE)   - FUNCTION NOT YET AVAILABLE  
4  (FUTURE)   - FUNCTION NOT YET AVAILABLE  
5  (FUTURE)   - FUNCTION NOT YET AVAILABLE  
  
ENTER END COMMAND TO TERMINATE.
```

FORTRAN - FUNCTION EMPLFN

C EMPLOYEE UPDATE FUNCTION

C THE CALL TO ISPLINK IS DONE VIA AN ASSEMBLER PROCEDURE CALLED ISPLINK
C WHICH IS A FUNCTION OF THE FORTRAN PROGRAM. IT IS INVOKED BY THE
C STATEMENT LASTRC = ISPLNK(ARGLIST). THE RETURN CODE FROM ISPLINK
C IS RETURNED IN THE INTEGER LASTRC WHEN ISPLINK IS CALLED.

C VARIABLES FOR FORTRAN PROGRAM EMPLFN
C DEFAULT ALL VARIABLES TO INTEGERS OF LENGTH FOUR
C ALL CHARACTER STRINGS ARE STORED IN ARRAYS WHICH ARE DEFINED
C IN THE DIMENSION STATEMENTS.

```
IMPLICIT INTEGER (A-Z)  
DIMENSION EMPSER(2)  
DIMENSION FNAME(4),LNAME(4)  
DIMENSION ADDR1(10),ADDR2(10),ADDR3(10),ADDR4(10)  
DIMENSION PHNUM(2),MSG(2),MSG1(2),MSG2(2),MSG3(2)  
DATA N/'N'/  
DATA U/'U'/  
DATA BLNK4 /'       '/  
DATA MSG1 /'EMPX','217 '/  
DATA MSG2 /'EMPX','218 '/  
DATA MSG3 /'EMPX','219 '/
```

C VARIABLES DEFINED FOR DIALOG SERVICE USE

```
LASTRC = ISPLNK('VDEFINE','(EMPSER)',EMPSER,'CHAR',8)  
LASTRC = ISPLNK('VDEFINE','(FNAME)',FNAME,'CHAR',16)  
LASTRC = ISPLNK('VDEFINE','(LNAME)',LNAME,'CHAR',16)  
LASTRC = ISPLNK('VDEFINE','(I)',I,'CHAR',4)
```

```

LASTRC = ISPLNK('VDEFINE', '(ADDR1)', ADDR1, 'CHAR', 40)
LASTRC = ISPLNK('VDEFINE', '(ADDR2)', ADDR2, 'CHAR', 40)
LASTRC = ISPLNK('VDEFINE', '(ADDR3)', ADDR3, 'CHAR', 40)
LASTRC = ISPLNK('VDEFINE', '(ADDR4)', ADDR4, 'CHAR', 40)
LASTRC = ISPLNK('VDEFINE', '(PHA)', PHA, 'CHAR', 4)
LASTRC = ISPLNK('VDEFINE', '(PHNUM)', PHNUM, 'CHAR', 8)
LASTRC = ISPLNK('VDEFINE', '(CHKTYPE)', CHKTYP, 'CHAR', 4)
LASTRC = ISPLNK('VDEFINE', '(TYPECHG)', TYPCHG, 'CHAR', 4)

```

```

C START OF PROCEDURE
C OPEN TABLE IF IT EXISTS, ELSE CREATE THE TABLE
C TABLE NAME MUST BE SIX CHARACTERS AS IN FORTRAN

```

```

DO 1 J = 1,2
  MSG(J) = BLNK4
  EMPSER(J) = BLNK4
1  CONTINUE
  LASTRC = ISPLNK('TBOPEN', 'EMPLTBL ')
  IF(LASTRC.EQ.0) GO TO 10

```

```

C TABLE DOES NOT EXIST SO CREATE IT

```

```

LASTRC = ISPLNK('TBCREATE', 'EMPLTBL ', '(EMPSER)',
& '(LNAME FNAME I ADDR1 ADDR2 ADDR3 ADDR4 PHA PHNUM)')

```

```

C SELECT EMPLOYEE

```

```

10  CONTINUE
    LASTRC = ISPLNK('DISPLAY', 'EMPLA ', MSG)

```

```

C TERMINATE IF END KEY PRESSED

```

```

IF(LASTRC.EQ.8) GO TO 999

```

```

C ELSE THE ENTER KEY WAS PRESSED

```

```

DO 2 J = 1,2
  MSG(J) = BLNK4
2  CONTINUE
  LASTRC = ISPLNK('TBGET', 'EMPLTBL ')

```

```

C IF RECORD EXISTS THEN SET UPDATE FLAG, ELSE TYPE IS NEW

```

```

IF(LASTRC.EQ.0) GO TO 25
  GO TO 30
25  CONTINUE
    TYPCHG = U
    GO TO 32

```

```

C INITIALIZE PANEL VARIABLES TO BLANKS VIA LOOPS

```

```

30  CONTINUE
    TYPCHG = N

```

```

        DO 3 J = 1,4
            LNAME(J) = BLNK4
            FNAME(J) = BLNK4
3        CONTINUE
        PHA = BLNK4
        I = BLNK4
        DO 5 J = 1,10
            ADDR1(J) = BLNK4
            ADDR2(J) = BLNK4
            ADDR3(J) = BLNK4
            ADDR4(J) = BLNK4
5        CONTINUE
        DO 6 J = 1,2
            PHNUM(J)= BLNK4
6        CONTINUE

C SAVE TYPE OF CHANGE

32     CONTINUE
        CHKTYP = TYPCHG

C DISPLAY EMPLOYEE DATA

        LASTRC = ISPLNK('DISPLAY','EMPLB ')
        IF(LASTRC.EQ.8)          GO TO 70

C ENTER NEW EMPLOYEE

        IF (TYPCHG.EQ.N)          GO TO 40
        GO TO 50

C THEN
C NEW

40     CONTINUE
        LASTRC = ISPLNK('TBADD','EMPLTBL ')
        DO 8 J = 1,2
            MSG(J) = MSG1(J)
8        CONTINUE
        GO TO 70

C ELSE
C CHECK IF UPDATE OR DELETE

50     CONTINUE
        IF (TYPCHG.EQ.U)          GO TO 55
        GO TO 60

C THEN
C UPDATE

55     CONTINUE
        LASTRC = ISPLNK('TBPUR','EMPLTBL ')
        DO 9 J = 1,2

```



```

        MSG(J) = MSG2(J)
9    CONTINUE
    GO TO 70

C ELSE
C DELETE

60   CONTINUE
    LASTRC = ISPLNK('TBDELETE', 'EMPLTBL ')
    DO 11 J = 1,2
        MSG(J) = MSG3(J)
11   CONTINUE

C GO BACK TO START AND CHECK FOR ANOTHER ENTRY OR END KEY PRESSED

70   CONTINUE
    IF(MSG(1).NE.BLNK4) LASTRC = ISPLNK('LOG', MSG)
    GO TO 10

C DELETE VARIABLE DEFINITONS FOR EMPFPGM AND CLOSE EMLPTBL

999  CONTINUE
    LASTRC = ISPLNK('TBCLOSE', 'EMPLTBL ')
    LASTRC = ISPLNK('VDELETE', '(EMP SER)')
    LASTRC = ISPLNK('VDELETE', '(FNAME)')
    LASTRC = ISPLNK('VDELETE', '(LNAME)')
    LASTRC = ISPLNK('VDELETE', '(I)')
    LASTRC = ISPLNK('VDELETE', '(ADDR1)')
    LASTRC = ISPLNK('VDELETE', '(ADDR2)')
    LASTRC = ISPLNK('VDELETE', '(ADDR3)')
    LASTRC = ISPLNK('VDELETE', '(ADDR4)')
    LASTRC = ISPLNK('VDELETE', '(PHA)')
    LASTRC = ISPLNK('VDELETE', '(PHNUM)')
    LASTRC = ISPLNK('VDELETE', '(TYPCHG)')
    LASTRC = ISPLNK('VDELETE', '(CHKTYP)')
    STOP
    END

```

PL/I IMPLEMENTATION

The components listed under "Dialog Components Applicable to All Implementations," and the following are required for the PL/I implementation of the dialog:

- Primary option menu definition EMPL
- Function EMPLFN

PL/I - PRIMARY OPTION MENU DEFINITION EMPL

Definition and display of primary option menu EMPL are shown below.

Panel Definition

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

Panel Display

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

SELECT OPTION ==> 1_

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

ENTER END COMMAND TO TERMINATE.
```

PL/I - FUNCTION EMPLFN

```

EMPLFN: PROC OPTIONS(MAIN);                                /*EMPLOYEE UPDATE FUNCTION */
%INCLUDE EMPLDCL;                                        /*DCL & DEFINE VARIABLES*/
MSG = ' ';                                             /*INITIALIZE MESSAGE */
CALL ISPLINK('TBOPEN', 'EMPLTBL ');                    /*OPEN TABLE */
IF PLIRETV() ^= 0 THEN                                  /*IF TABLE DOESN'T EXIST*/
  CALL ISPLINK('TBCREATE', 'EMPLTBL ', '(EMP SER)', /*CREATE IT */
              '(LNAME FNAME I ADDR1 ADDR2 ADDR3 ADDR4 PHA PHNUM)');
DO WHILE (STATE ^= '4');                                /*LOOP UNTIL TERM SET */
  CALL ISPLINK('DISPLAY', 'EMPLA ', MSG);              /*SELECT EMPLOYEE */
  IF PLIRETV() = 8 THEN                                  /*IF END KEY PRESSED */
    STATE = '4';                                       /* TERMINATE */
  ELSE DO;                                              /*ENTER KEY PRESSED */
    MSG = ' ';                                         /*RESET MESSAGE */
    STATE = '2';                                       /*PROCESS EMPLOYEE PANEL*/
    CALL ISPLINK('TBGET', 'EMPLTBL ');                 /*OBTAIN EMPLOYEE DATA */
    IF PLIRETV() = 0 THEN                               /*IF RECORD EXISTS */
      TYPECHG = 'U';                                   /* SET UPDATE FLAG */
    ELSE DO;                                           /*RECORD DOES NOT EXIST */
      TYPECHG = 'N';                                   /* SET TYPE = NEW */
      LNAME = ' ';                                     /* */
      FNAME = ' ';                                     /* INITIALIZE */
      I = ' ';                                         /* PANEL */
      ADDR1 = ' ';                                     /* VARIABLES */
      ADDR2 = ' ';                                     /* TO NULLS */
      ADDR3 = ' ';                                     /* */
      ADDR4 = ' ';                                     /* */
      PHA = ' ';                                       /* */
      PHNUM = ' ';                                     /* */
    END;                                               /* */
    CHKTYPE = TYPECHG;                                 /*SAVE TYPE OF CHANGE */
    CALL ISPLINK('DISPLAY', 'EMPLB ');                 /*DISPLAY EMPLOYEE DATA */
    IF PLIRETV() ^= 8 THEN DO;                          /*END KEY NOT PRESSED */
      IF TYPECHG = 'N' THEN DO;                         /*IF NEW EMPLOYEE */
        CALL ISPLINK('TBADD', 'EMPLTBL ');             /* ADD TO TABLE */
        MSG = 'EMPX217 ';                               /*EMPLOYEE ADDED MESSAGE*/
      END;                                             /* */
    ELSE DO;                                           /* */
      IF TYPECHG = 'U' THEN DO;                         /*IF UPDATE REQUESTED */
        CALL ISPLINK('TBPUT', 'EMPLTBL ');             /* UPDATE TABLE */
        MSG = 'EMPX218 ';                               /* UPDATE MESSAGE */
      END;                                             /* */
    ELSE DO;                                           /*ELSE ASSUME DELETE */
      CALL ISPLINK('TBDELETE', 'EMPLTBL ');            /* */
      MSG = 'EMPX219 ';                               /*EMPLOYEE DELETED MSG */
    END;                                             /* */
  END;                                               /*END TABLE MODS */
END;                                               /*END 2ND PANEL PROCESS */
END;                                               /*END 1ST PANEL PROCESS */
IF MSG ^= ' ' THEN CALL ISPLINK('LOG', MSG);          /*LOG MSG */
END;                                               /*END DO LOOP */

```

```

CALL ISPLINK('TBCLOSE', 'EMPLTBL ');      /*CLOSE TABLE      */
%INCLUDE EMPLDEL;                          /*DELETE DEFINED VARS*/
RETURN(0);
END EMPLFN;

```

Included Segment EMPLDCL

```

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

```

```

LPHNUM = LENGTH(PHNUM) ;
LTYPECH = LENGTH(TYPECHG) ;
LCHKTYP = LENGTH(CHKTYPE) ;
/*
/*DEFINE VARIABLES FOR DIALOG SERVICE USE
/*
CALL ISPLINK('VDEFINE', '(EMP SER)', EMP SER, 'CHAR', LEMP SER) ;
CALL ISPLINK('VDEFINE', '(FNAME)', FNAME, 'CHAR', LFNAME) ;
CALL ISPLINK('VDEFINE', '(LNAME)', LNAME, 'CHAR', LLNAME) ;
CALL ISPLINK('VDEFINE', '(I)', I, 'CHAR', LI) ;
CALL ISPLINK('VDEFINE', '(ADDR1)', ADDR1, 'CHAR', LADDR1) ;
CALL ISPLINK('VDEFINE', '(ADDR2)', ADDR2, 'CHAR', LADDR2) ;
CALL ISPLINK('VDEFINE', '(ADDR3)', ADDR3, 'CHAR', LADDR3) ;
CALL ISPLINK('VDEFINE', '(ADDR4)', ADDR4, 'CHAR', LADDR4) ;
CALL ISPLINK('VDEFINE', '(PHA)', PHA, 'CHAR', LPHA) ;
CALL ISPLINK('VDEFINE', '(PHNUM)', PHNUM, 'CHAR', LPHNUM) ;
CALL ISPLINK('VDEFINE', '(TYPECHG)', TYPECHG, 'CHAR', LTYPECH) ;
CALL ISPLINK('VDEFINE', '(CHKTYPE)', CHKTYPE, 'CHAR', LCHKTYP) ;
/*

```

Included Segment EMPLDEL

```

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

```

EXAMPLE 2. TELEPHONE BOOK APPLICATION

This section describes implementations of a "telephone book" application dialog. Provided implementations include dialog functions coded in:

- CLIST
- EXEC2
- COBOL
- FORTRAN
- PL/I

OVERVIEW

The telephone book application dialog creates, maintains, displays, and reports information about employees to be included in a telephone book. The information is saved in an ISPF table. Each row of the table contains the following information for each employee:

- First name
- Last name
- Location
- Phone number
- Color of phone
- Type of phone

The employee last name is the KEY field for the table.

The telephone book application dialog performs the following functions:

- Displays the table using the table display facility. The user may scroll up and down in the table. If a row of the table is selected on the table display panel, the table will be positioned at that row for the next interaction.
- Adds entries to the table.
- Changes existing table entries.
- Deletes table entries.

- Gets and displays the last entry in the table.
- Modifies an entry in the table; if the entry does not exist, adds it to the table.
- Sets the current table values displayed on the screen to blanks.
- Uses file tailoring to produce a report on the contents of the table and stores the report as a member of the file tailoring output library.
- Invokes the ISPF/PDF browse function to display the contents of the telephone book table report.
- Invokes the ISPF/PDF edit function to allow display and edit of the contents of the telephone book table report.
- Erases the telephone book table report from the file tailoring output library.
- Reads a new copy of the telephone book table from the table input library.
- Erases the telephone book table from the table output library.
- Queries the table and displays the current number of rows in the table, the name(s) of the table key variable(s), and the names of the non-key variables in the table.
- Scans the table for a row that matches any combination of table variable values specified.
- Rescans the table for another row that matches the variable values from the previous scan.
- Searches the table for a row with a specified key variable value.
- Saves the table on disk.

DIALOG COMPONENTS

The dialog components that implement the telephone book application are:

- A main function routine (TELEBOOK), which receives control when the telephone book dialog is invoked.
- A combined selection and data entry panel, which is SELECTed by the main function routine. This panel is named:
 - CPANEL1, for CLIST and EXEC2 implementations
 - TPANEL1, for COBOL and PL/I implementations)
 - FPANEL1, for the FORTRAN implementation

- A subfunction routine (TELESRVC), which is invoked from the selection panel by the SELECT service to perform all of the table maintenance, display, and report functions.
- A table display panel (TPANEL2), which is used by the subfunction routine to display the contents of a telephone book table.
- A display panel (TPANEL3), which is used by the subfunction to display the number of rows and variable names contained in the telephone book table.
- A set of message definitions (TELBO1, TELBO2), which are used by the panels and functions to display information and error conditions on the panel message lines.
- A file skeleton (TELBOOK), which is used by the subfunction routine to build a report on the contents of the telephone book table.

The organization of the dialog is shown in Figure 2.

INVOCATION

The main function routine (TELEBOOK) receives control when the telephone book dialog is invoked. The dialog may be invoked by any of the following mechanisms:

- When invoking ISPF, the telephone book dialog may be specified on the ISPSTART command:
 - For CLIST and EXEC2 implementations:


```
ISPSTART CMD(TELEBOOK)
```
 - For COBOL, FORTRAN, and PL/I implementations:


```
ISPSTART PGM(TELEBOOK)
```
- When ISPF is already active, the dialog may be invoked from any selection panel on which it appears as an option. The &ZSEL variable in the processing section for the selection panel would have to include the telephone book application option as follows:

```
&ZSEL = TRAN (TRUNC (&ZCMD, '.'))
      :
      :
      'x', 'CMD(TELEBOOK) 'NEWAPPL'
           OR
      'x', 'PGM(TELEBOOK) 'NEWAPPL'
```

where 'x' is the option defined in the displayable portion of the selection panel for the telephone book dialog. 'CMD(TELEBOOK)' is used if the dialog function is implemented in CLIST or EXEC2. 'PGM(TELEBOOK)' is used if the dialog function is implemented in COBOL, FORTRAN, or PL/1.

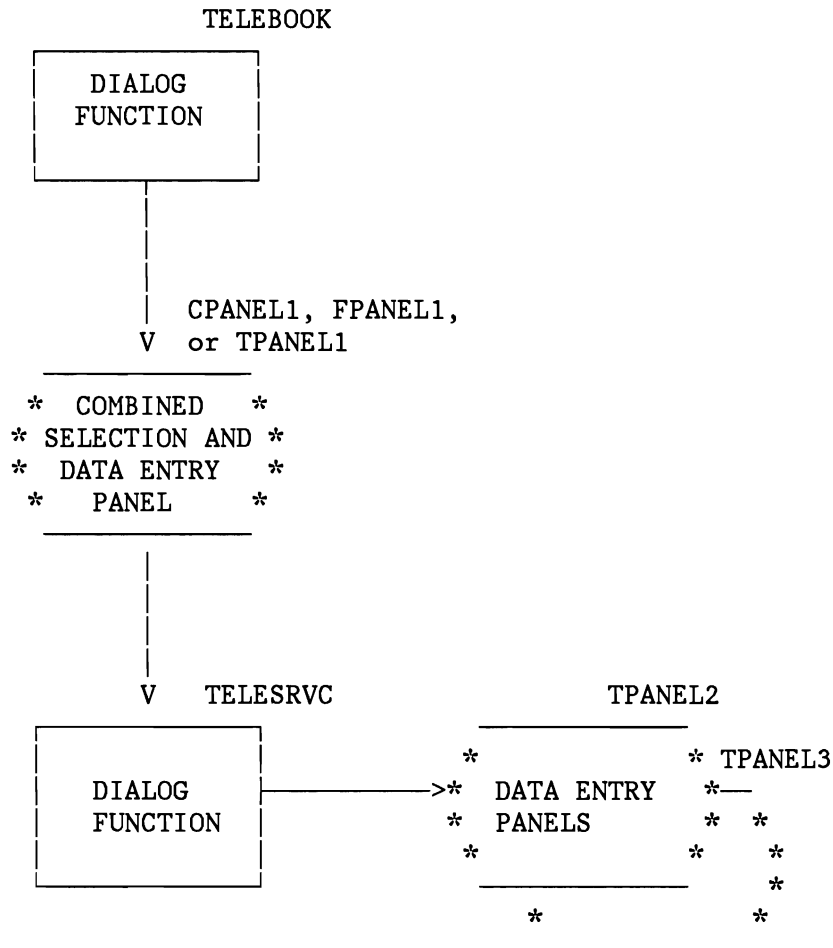


Figure 2. Organization of Telephone Book Application

- When ISPF is already active, the dialog may also be invoked from the command line of any selection panel or display panel if the active application's command table or the system command table contains the following entry:

```
'ZCTVERB = TELEBOOK'
```

```
'ZCTTRUNC = 3' or any value 2 or greater that makes the command verb unique in the user's system.
```

```
'ZCTACT = SELECT CMD(TELEBOOK) NEWAPPL' if the function routines are implemented as TSO CLIST or VM EXEC2 routines.
```

```
'ZCTACT = SELECT PGM(TELEBOOK) NEWAPPL' if the function routines are implemented as COBOL, FORTRAN, or PL/1 programs.
```

The application identifier is not specified with the 'NEWAPPL' keyword because the telephone book application dialog does not require its own profile variable and or command table. The 'NEWAPPL' keyword is optional and may be omitted if shared variable pool variables defined by the telephone book application dialog do not conflict with variables defined by other applications implemented under ISPF in the system.

DESCRIPTION OF PROCESSING

Application processing, performed by functions and the)PROC section of panels, is described below.

MAIN FUNCTION ROUTINE TELEBOOK

The main function routine performs the following processing:

- Opens the telephone book table (TELEBOOK) in write mode or creates a new table if it does not exist.
- Initializes the table variable in the shared variable pool to the values contained in the first row of the table.
- Calls ISPF SELECT to display the combined selection/data entry panel.
- Closes the table, sets a termination message via the SETMSG service, and terminates when the user enters END or RETURN in response to the selection/data entry panel.

COMBINED SELECTION AND DATA ENTRY PANEL CPANEL1, FPANEL1, OR TPANEL1

Because a different interface is required to invoke functions written as TSO CLISTS and VM EXEC2 routines (i.e., SELECT CMD ...) and functions written in a programming language (i.e., SELECT PGM ...), multiple versions of this panel are shown.

The combined selection and data entry panel provides the following:

- Displays a list of all of the table maintenance, display, and report functions.
- Provides data entry fields for all of the variables in a row of the table. When the panel is displayed, these fields contain the values of the current table row except following a 'blank' row request.
- The user may modify the values in the data entry fields in conjunction with the option specified on the command line to:
 - Add a row of information to the table.
 - Change a row of information in the table.

- Delete a row of information from the table.
- Add/change a row of information in the table.
- Scan or search for a specified row of information in the table.
- Invokes the subfunction routine (TELESRVC) to perform an action requested by the user (add, change, etc.). A special version of the panel (FPANEL1) is included for the FORTRAN implementation to illustrate yet another mechanism for passing parameters to the subfunction routine (TELESRVC).

SUBFUNCTION ROUTINE TELESRVC

This function routine receives control from the selection/data entry panel (CPANEL1, FPANEL1, or TPANEL1) via the ISPF SELECT service. The option requested by the user is passed to TELESRVC from the selection panel as a CLIST or EXEC2 parameter, in the PARM field for COBOL and PL/1, and via the shared variable pool for FORTRAN. The subfunction routine (TELESRVC) does the following:

- Gets the table variable values specified in the selection data entry menu from the shared variable pool.
- Executes the appropriate ISPF service based on the service requested by the user on the command line.
- For the table display (LIST) function, invokes ISPF table display panel TPANEL2.
- For the table query function, displays the results using panel TPANEL3.
- For the report function, invokes the ISPF file tailoring function using the TELBOOK file skeleton.
- Invokes the ISPF/PDF (PDF) browse and edit functions to display and edit the output of the report function. If PDF is not installed, this function will fail.

DIALOG COMPONENTS APPLICABLE TO ALL IMPLEMENTATIONS

Dialogs applicable to all of the command procedure and programming language implementations of the application are:

- Table display panel definition TPANEL2
- Table information panel definition TPANEL3
- Message definitions TELB01 and TELB02
- File tailoring skeleton TELBOOK

Detailed listings of these components follow.

TABLE DISPLAY PANEL TPANEL2

Definition of table display panel TPANEL2 is shown below, followed by the panel as displayed.

Panel Defintion

```

)ATTR
  @ TYPE(OUTPUT) INTENS(LOW)
)BODY
%----- TELEPHONE DIRECTORY -----
%COMMAND INPUT ==> _ZCMD                                %SCROLL ==> _AMT

+S   FIRST      LAST      PHONE  LOCATION  PHONE  PHONE
+   NAME        NAME        NUMBER          TYPE      COLOR
+
)MODEL
_Z @Z          @Z          @Z          @Z          @Z          @Z
)INIT
.CURSOR=ZCMD
&AMT='HALF'
&TSEL=' '
&ZCMD=' '
.ZVARS='(TSEL TFNAM TLNAM TNUM TLOC TTYPE TCOLOR)'
)END

```

Panel Display

-----		TELEPHONE DIRECTORY			-----	
COMMAND INPUT ==> _					SCROLL ==> HALF	
S	FIRST NAME	LAST NAME	PHONE NUMBER	LOCATION	PHONE TYPE	PHONE COLOR
	LARRY	MILNS	385-2222	STUDY	DESK DIAL	WHITE
	ARTHUR	JONES	892-3232	DINING ROOM	WALL DIAL	BLUE
	KATHERINE	GREEN	762-8012	LIVING ROOM	PRINCESS	WHITE
	LINDA	MILLER	462-4441	KITCHEN	WALL DIAL	RED
	RICHARD	THOMPSON	332-4812	KITCHEN	WALL TOUCH	BLACK
***** BOTTOM OF DATA *****						

MESSAGE DEFINITIONS TELB01

TELBO10 'ENTRY ALREADY EXISTS' .ALARM=YES
'CANNOT ADD AN EXISTING ENTRY'

TELBO11 'ENTRY DOES NOT EXIST' .ALARM=YES
'NO ACTION TAKEN FOR ATTEMPT TO DELETE NON-EXISTENT ENTRY'

TELBO12 'ENTRY DELETED' .ALARM=YES
'THE ENTRY HAS BEEN DELETED FROM THE TABLE'

TELBO13 'ENTRY ADDED TO TABLE' .ALARM=YES
'THE ENTRY SHOWN BELOW HAS BEEN ADDED TO THE TABLE'

TELBO14 'ENTRY MODIFIED' .ALARM=YES
'THE ENTRY SHOWN BELOW HAS BEEN MODIFIED IN THE TABLE'

TELBO15 'ENTRY CHANGED' .ALARM=YES
'THE ENTRY SHOWN BELOW HAS BEEN CHANGED IN THE TABLE'

TELBO16 'LAST LINE FOUND' .ALARM=YES
'THE ENTRY SHOWN BELOW IS THE LAST LINE IN THE TABLE'

TELBO17 'LAST LINE NOT FOUND' .ALARM=YES
'THE TABLE IS EMPTY OR AN ERROR CONDITION EXISTS'

TELBO18 'ENTRY DOES NOT EXIST' .ALARM=YES
'LAST NAME NOT IN TABLE OR AN ERROR CONDITION EXISTS'

TELBO19 'TABLE SAVED' .ALARM=YES
'THE TABLE HAS BEEN SAVED ON DISK'

MESSAGE DEFINITIONS TELB02

TELBO20 'TABLE REMOVED' .ALARM=YES
'THE TABLE HAS BEEN REMOVED FROM THE DISK'

TELBO21 'TABLE REFRESHED' .ALARM=YES
'THE TABLE HAS BEEN REFRESHED FROM THE DISK'

TELBO22 'REPORT PREPARED' .ALARM=YES
'THE TABLE HAS BEEN FORMATED INTO A REPORT'

TELBO23 'REPORT DESTROYED' .ALARM=YES
'THE FORMATTED TABLE REPORT HAS BEEN DELETED'

TELBO24
'THE TELEPHONE BOOK APPLICATION HAS BEEN TERMINATED'

TELBO25
' '

FILE TAILORING SKELETON TELBOOK

)TB 15 28 37 51 65

TELEPHONE BOOK FILE TAILORING OUTPUT

LAST NAME!FIRST NAME!TEL NUM!LOCATION!TYPE!COLOR

-----!-----!-----!-----!-----!-----

)DOT TELBOOK

&TLNAM !&TFNAM !&TNUM !&TLOC !&TTYPE !&TCOLOR

)ENDDOT

CLIST IMPLEMENTATION

In addition to the components listed under "Dialog Components Applicable to All Implementations," the following components are required for the CLIST implementation of the dialog:

- Combined selection and data entry panel CPANEL1
- Function TELEBOOK
- Function TELESRVC

CLIST - COMBINED SELECTION AND DATA ENTRY PANEL CPANEL1

Definition of combined selection and data entry panel CPANEL1 is shown below, followed by the panel as displayed.

Panel Definition

)ATTR

TYPE(INPUT)	INTENS(LOW)	JUST(LEFT)	PAD(' ')	CAPS(ON)
@ TYPE(INPUT)	INTENS(HIGH)	JUST(LEFT)	PAD(' ')	CAPS(ON)

)BODY

```
%----- TELEPHONE BOOK APPLICATION -----  
+SELECT AN OPTION:%==>@ZCMD  
%
```

```

% A+- Add new entry
% C+- Change existing entry
% D+- Delete old entry
% G+- Get last entry
% M+- Add/Change entry
% Z+- Blank screen entry
% P+- Prepare report
% B+- Browse report
% E+- Edit report
% Y+- Destroy report
%
%F+- Refresh the table, from disk
%K+- Remove the table, from disk
%L+- List the table
%Q+- Query the table
%S+- Scan the table, NON UNIQUE
%R+- Rescan the table, continue scan
%T+- Search on last name, unique key
%V+- Save the table, on disk

```

```

+Current values / enter search values:

```

```

%
+ FIRST NAME ==>_TFNAM      +
+ LAST NAME  ==>_TLNAM      +
+ LOCATION   ==>_TLOC       +
+ PHONE NUM  ==>_TNUM       +
+ PHONE COLOR==>_TCOLOR     +
+ PHONE TYPE ==>_TTYPE      +

```

```

)INIT

```

```

.CURSOR = ZCMD
&ZCMD = &OTELOPT
IF(&ZERRMSG ^= ' ')
  .MSG = &ZERRMSG
  &ZERRMSG = ' '

```

```

)PROC

```

```

&ZSEL = TRANS( &ZCMD
              L,'CMD(TELESRVC LIST      )'
              A,'CMD(TELESRVC ADD       )'
              C,'CMD(TELESRVC CHANGE   )'
              D,'CMD(TELESRVC DELETE   )'
              G,'CMD(TELESRVC GETLAST  )'
              M,'CMD(TELESRVC MODIFY   )'
              Z,'CMD(TELESRVC BLANK     )'
              S,'CMD(TELESRVC SCAN      )'
              R,'CMD(TELESRVC RESCAN   )'
              T,'CMD(TELESRVC SEARCH   )'
              Q,'CMD(TELESRVC QUERY     )'
              P,'CMD(TELESRVC PREPREP  )'
              B,'CMD(TELESRVC BROWSE   )'
              E,'CMD(TELESRVC EDIT     )'
              Y,'CMD(TELESRVC DESTROY  )'
              F,'CMD(TELESRVC REFRESH  )'
              K,'CMD(TELESRVC REMOVE   )'
              V,'CMD(TELESRVC SAVE     )'
              ',',' '
              *,'? ' )

```

```

&OSEL = &ZSEL

```



```

IF(&ZCMD = S)
  &ZCMD = R
&OTELOPT = &ZCMD

IF(&ZCMD = R)
  VER(&TNUM,PICT,'NNN-NNNN')

)END

```

Panel Display

```

----- TELEPHONE BOOK APPLICATION -----
SELECT AN OPTION: ==> A_

A - Add new entry
C - Change existing entry
D - Delete old entry
G - Get last entry
M - Add/Change entry
Z - Blank screen entry
P - Prepare report
B - Browse report
E - Edit report
Y - Destroy report

F - Refresh the table, from disk
K - Remove the table, from disk
L - List the table
Q - Query the table
S - Scan the table, NON UNIQUE
R - Rescan the table, continue scan
T - Search on last name, unique key
V - Save the table, on disk

Current values / enter search values:

FIRST NAME ==> RICHARD
LAST NAME ==> THOMPSON
LOCATION ==> KITCHEN
PHONE NUM ==> 332-4812
PHONE COLOR==> BLACK
PHONE TYPE ==> WALL TOUCH

```

CLIST - FUNCTION TELEBOOK

```

/* THE OBJECTIVE OF THE CODING TECHNIQUE OF THIS APPLICATION IS
/* TO ILLUSTRATE ISPF SERVICES
/*
/* THE APPLICATION CONSISTS OF
/* THIS MAIN PROGRAM - TELEBOOK
/* 1 SUB PROGRAM - TELESRVC
/* 3 PANELS - CPANEL1 TPANEL2 TPANEL3
/* 15 MESSAGES - TELBO10 THRU TELBO24
/* 1 FILE TAILORING SKELETON - TELBOOK
/*
/* THE APPLICATION PROVIDES THE FOLLOWING FUNCTIONS

```

```

/* FROM PANEL CPANEL1 TO ACCESS A TELEPHONE BOOK
/*
/* A ADD NEW ENTRY
/* C CHANGE EXISTING ENTRY F REFRESH THE TABLE, FROM DISK
/* D DELETE OLD ENTRY K REMOVE THE TABLE, FROM DISK
/* G GET LAST ENTRY L LIST THE TABLE
/* M ADD/CHANGE ENTRY Q QUERY THE TABLE
/* Z BLANK SCREEN ENTRY S SCAN THE TABLE, NON-UNIQUE
/* P PREPARE REPORT R RESCAN THE TABLE, CONTINUE SCAN
/* B BROWSE REPORT T SEARCH ON LAST NAME, UNIQUE KEY
/* E EDIT REPORT V SAVE THE TABLE, ON DISK
/* Y DESTROY REPORT
    SET TERMMESSAGE = TELB024
    SET PANEL1 = CPANEL1
    SET TABKEY = TLNAM
    SET TABVRS = &STR(TFNAM TLOC TNUM TCOLOR TTYPE)
    SET TELBOOK = TELBOOK
    SET VPUTLIST = &STR(TFNAM TLNAM TLOC TNUM TCOLOR TTYPE)

```

```

/*****
/* CALL THE CONTROL SERVICE TO SET THE ERROR PROCESSING *
/* MODE FOR THIS PROGRAM TO TERMINATE ONLY FOR ANY ERROR *
/* THAT HAS A RETURN CODE OF 12 OR HIGHER *
/*****

```

ISPEXEC CONTROL ERRORS CANCEL

```

/*****
/* CALL THE TBOPEN SERVICE TO *
/* OPEN THE TABLE, TELBOOK IN WRITE MODE *
/*****

```

ISPEXEC TBOPEN &TELBOOK WRITE

```

/*****
/* CALL THE TBCREATE SERVICE TO *
/* CREATE THE TABLE TELBOOK IF IT DOES NOT EXIST. THE *
/* TABLE KEY IS THE LAST NAME VARIABLE SPECIFIED IN TABKEY *
/* AND THE NON-KEY VARIABLES ARE SPECIFIED IN TABVRS *
/*****

```

IF &LASTCC = 8 THEN +
ISPEXEC TBCREATE &TELBOOK KEYS(&TABKEY) NAMES(&TABVRS)

```

/*****
/* CALL THE TBSKIP SERVICE TO MOVE THE CURRENT ROW POINTER *
/* (CRP) OF THE TABLE TELBOOK FORWARD ONE ROW (DEFAULT). *
/* THE VARIABLES DEFINED FOR THE TABLE ARE SET TO THE VALUES *
/* CONTAINED IN THE FIRST ROW OF THE TABLE. IF THE TABLE *
/* HAS JUST BEEN CREATED, THE VARIABLE VALUES REMAIN NULL. *
/*****

```

ISPEXEC TBSKIP &TELBOOK

```

/*****
/*   SET THE TABLE VARIABLES IN THE SHARED VARIABLE POOL TO THE *
/*   VALUES IN THE 1ST TABLE ROW.  THESE VALUES WILL BE USED TO *
/*   INITIALIZE THOSE VARIABLES IN THE SELECT PANEL CPANEL1.      *
/*****

```

ISPEXEC VPUT (&VPUTLIST) SHARED

```

/*****
/*   CALL THE SELECT SERVICE TO CONTINUE TO DISPLAY CPANEL1      *
/*   AND CONTROL THE INVOCATION OF SUBPROGRAM TELESRVC           *
/*****

```

ISPEXEC SELECT PANEL(&PANEL1)

```

/*****
/*   CALL THE TBCLOSE SERVICE TO CLOSE THE TABLE, TELBOOK     *
/*****

```

ISPEXEC TBCLOSE &TELBOOK

```

/*****
/*   CALL THE LOG SERVICE TO                                     *
/*   PUT THE MESSAGE TELB024 ON THE ISPF LOG FILE               *
/*****

```

ISPEXEC LOG MSG(&TERMMESSAGE)

```

/*****
/*   CALL THE SETMSG SERVICE TO                                  *
/*   PUT THE MESSAGE TELB024 ON THE NEXT SCREEN DISPLAYED      *
/*****

```

ISPEXEC SETMSG MSG(&TERMMESSAGE)

CLIST - FUNCTION TELESRVC

```

PROC 1 SERVN1
/*   THIS CLIST IS INVOKED FROM PANEL CPANEL1 DISPLAYED
/*   BY THE MAIN CLIST TELEBOOK USING THE SELECT SERVICE
    SET SERVN   = &SERVN1
    SET VARLIST = &STR(TNUM TFNAM TLNAM TLOC TTYPE TCOLOR)
    SET PANEL2  = TPANEL2
    SET PANEL3  = TPANEL3
    SET TELBOOK = TELBOOK
    SET TELOUT  = TELOUT
    SET TABKEY  = TLNAM
    SET TABVARS = &STR(TFNAM TLOC TNUM TCOLOR TTYPE)
    SET ZERRMSG =
    SET ZMSGNAME = ZERRMSG

```

```

/*****
/*      FTOOUTPUT SHOULD BE SET TO THE USER'S FILE TAILORING      *
/*      OUTPUT PARTITIONED DATASET WITH THE MEMBER NAME SPECIFIED *
/*      AS THE VARIABLE '&TELOUT'.                                *
/*****

```

```

      SET FTOOUTPUT = 'ISPFPROJ.FTOOUTPUT(&TELOUT)'
```

```

/*****
/*      CALL THE VGET SERVICE TO                                  *
/*      COPY THE VALUES LISTED IN VARLIST FROM THE SHARED      *
/*      VARIABLE POOL TO THIS CLIST'S SET OF FUNCTION           *
/*      VARIABLES.  THE VALUES IN THE SHARED POOL ARE THOSE    *
/*      RETURNED FROM THE SELECT PANEL CPANEL1.                 *
/*****

```

```

      ISPEXEC VGET (&VARLIST) SHARED
```

```

/*****
/*      &SERVN CONTAINS THE FUNCTION REQUESTED BY THE USER ON  *
/*      THE SELECT PANEL CPANEL1.                                *
/*****

```

```

      IF      &SERVN = LIST      THEN DO
```

```

/*****
/*      CALL THE TBTOP SERVICE TO                                *
/*      SET THE CURRENT ROW POINTER (CRP) AHEAD OF THE FIRST    *
/*      ROW OF THE TABLE TELBOOK                               *
/*****

```

```

      ISPEXEC TBTOP &TELBOOK
```

```

/*****
/*      CALL THE TBDISPL SERVICE TO                              *
/*      DISPLAY THE TABLE TELBOOK USING THE PANEL TPANEL2     *
/*****

```

```

      ISPEXEC TBDISPL &TELBOOK  PANEL(&PANEL2)
```

```

      END
```

```

      ELSE IF &SERVN = ADD      THEN DO
```

```

/*****
/*      CALL THE TBADD SERVICE TO TRY TO                        *
/*      ADD A NEW ROW OF VARIABLES TO THE TABLE TELBOOK       *
/*****

```

```

      ISPEXEC TBADD &TELBOOK
```

```

      IF LASTCC = 0 THEN SET ZERRMSG = TELB013
```

```

      ELSE SET ZERRMSG = TELB010
```

```

      END
```

```

      ELSE IF &SERVN = CHANGE  THEN DO
```

```

/*****
/*  CALL THE TBPOT SERVICE TO TRY TO  *
/*  UPDATE A ROW OF VARIABLES IN THE TABLE TELBOOK  *
/*****

        ISPEXEC TBPOT &TELBOOK
        IF LASTCC = 0 THEN SET ZERRMSG = TELB015
                ELSE SET ZERRMSG = TELB018
        END

ELSE IF &SERVN = DELETE  THEN DO

/*****
/*  CALL THE TBDELETE SERVICE TO TRY TO  *
/*  DELETE A ROW OF VARIABLES IN THE TABLE TELBOOK  *
/*****

        ISPEXEC TBDELETE &TELBOOK
        IF LASTCC = 8 THEN SET ZERRMSG = TELB011
                ELSE SET ZERRMSG = TELB012
        ISPEXEC TBSKIP &TELBOOK NUMBER(0)
        IF &LASTCC = 8 THEN ISPEXEC TBSKIP &TELBOOK NUMBER(1)
        END
ELSE IF &SERVN = GETLAST  THEN DO

/*****
/*  CALL THE TBBOTTOM SERVICE TO  *
/*  SET THE CURRENT ROW POINTER (CRP) TO THE LAST ROW OF  *
/*  THE TABLE TELBOOK AND TO RETRIEVE THAT ROW  *
/*****

        ISPEXEC TBBOTTOM &TELBOOK
        IF &LASTCC = 0 THEN SET ZERRMSG = TELB016
                ELSE SET ZERRMSG = TELB017
        END
ELSE IF &SERVN = MODIFY  THEN DO

/*****
/*  CALL THE TBMOD SERVICE TO  *
/*  UPDATE OR ADD A ROW OF VARIABLES TO THE TABLE TELBOOK  *
/*****

        ISPEXEC TBMOD &TELBOOK
        IF LASTCC = 0 THEN SET ZERRMSG = TELB014
                ELSE SET ZERRMSG = TELB013
        END

```

```

/*****
/*  SET THE CURRENT VALUES OF THE TABLE VARIABLES TO NULL.      *
/*  WHEN CPANEL1 IS REDISPLAYED, THE TABLE VARIABLES WILL      *
/*  CONTAIN NULL VALUES ON THE SCREEN.                          *
/*****

```

```

        ELSE IF &SERVN = BLANK      THEN ISPEXEC TBVCLEAR &TELBOOK
        ELSE IF &SERVN = SEARCH    THEN DO

```

```

/*****
/*  CALL THE TBEXIST SERVICE TO                                  *
/*  TEST FOR THE EXISTENCE OF A ROW IN THE TABLE WITH THE      *
/*  SPECIFIED LAST NAME (KEY VARIABLE).                          *
/*****

```

```

        ISPEXEC TBEXIST &TELBOOK

```

```

/*****
/*  IF THE ROW WAS FOUND,                                       *
/*  CALL THE TBGET SERVICE TO                                    *
/*  FETCH THE VALUES FOR THE ROW FROM THE TABLE. OTHERWISE,   *
/*  THE VALUES FOR THE TABLE VARIABLES WILL BE UNCHANGED.    *
/*****

```

```

        IF &LASTCC = 0 THEN ISPEXEC TBGET &TELBOOK
        END
        ELSE IF &SERVN = QUERY      THEN DO

```

```

/*****
/*  CALL THE TBQUERY SERVICE TO                                  *
/*  OBTAIN FROM THE TABLE TELBOOK THE KEY VARIABLE NAME(S),    *
/*  THE NON-KEY VARIABLE NAMES, AND THE NUMBER OF ROWS          *
/*****

```

```

        ISPEXEC TBQUERY &TELBOOK KEYS(QKEYS) NAMES(QNAMES) ROWNUM(QROWS)

```

```

/*****
/*  CALL THE DISPLAY SERVICE TO DISPLAY THE PANEL TPANEL3      *
/*****

```

```

        ISPEXEC DISPLAY PANEL(&PANEL3)
        END
        ELSE IF &SERVN = PREPREP    THEN DO

```

```

/*****
/*  CALL THE FTOPEN SERVICE TO BEGIN FILE TAILORING            *
/*****

```

```

        ISPEXEC FTOPEN

```

```
/*
*****
/* CALL THE FTINCL SERVICE TO
/* DO FILE TAILORING USING THE FILE TAILORING SKELETON TELBOOK
*****
```

ISPEXEC FTINCL &TELBOOK

```
/*
*****
/* CALL THE FTCLOSE SERVICE TO TERMINATE FILE TAILORING
/* AND STORE THE OUTPUT AS MEMBER TELOUT
*****
```

ISPEXEC FTCLOSE NAME(&TELOUT)
SET ZERRMSG = TELB022
END

ELSE IF &SERVN = BROWSE THEN DO

```
/*
*****
/* CALL THE BROWSE SERVICE TO BROWSE TELOUT IN
/* FTOUTPUT. THE ISPF/PROGRAM DEVELOPMENT FACILITY
/* PROGRAM PRODUCT MUST BE INSTALLED TO USE THIS FUNCTION.
*****
```

ISPEXEC BROWSE DATASET(&FTOUTPUT)
END

ELSE IF &SERVN = EDIT THEN DO

```
/*
*****
/* CALL THE EDIT SERVICE TO EDIT TELOUT IN THE
/* FTOUTPUT. THE ISPF/PROGRAM DEVELOPMENT FACILITY
/* PROGRAM PRODUCT MUST BE INSTALLED TO USE THIS FUNCTION.
*****
```

ISPEXEC EDIT DATASET(&FTOUTPUT)
END

ELSE IF &SERVN = DESTROY THEN DO

```
/*
*****
/* CALL THE FTERASE SERVICE TO
/* DELETE THE FILE TAILORING OUTPUT TELOUT
*****
```

ISPEXEC FTERASE &TELOUT
SET ZERRMSG = TELB023
END

ELSE IF &SERVN = REFRESH THEN DO

```
/*
*****
/* CALL THE TBEND SERVICE TO
/* CLOSE THE TABLE TELBOOK WITHOUT SAVING IT ON DISK
*****
```

ISPEXEC TBEND &TELBOOK

```

ISPEXEC TBOPEN &TELBOOK
ISPEXEC TBSKIP &TELBOOK
SET ZERRMSG = TELB021
END
ELSE IF &SERVN = REMOVE THEN DO
ISPEXEC TBCLOSE &TELBOOK

```

```

/*****
/* CALL THE TBERASE SERVICE TO *
/* DELETE THE TABLE TELBOOK FROM THE TABLE OUTPUT LIBRARY *
/*****

```

```

ISPEXEC TBERASE &TELBOOK
ISPEXEC TBCREATE &TELBOOK KEYS(&TABKEY) NAMES(&TABVARS)
SET ZERRMSG = TELB020
END
ELSE IF &SERVN = SAVE THEN DO

```

```

/*****
/* CALL THE TBSAVE SERVICE TO WRITE THE TABLE TELBOOK *
/* FROM VIRTUAL STORAGE TO THE TABLE OUTPUT LIBRARY *
/*****

```

```

ISPEXEC TBSAVE &TELBOOK
SET ZERRMSG = TELB019
END
ELSE IF &SERVN = SCAN THEN DO

```

```

/*****
/* CALL THE TBSARG SERVICE TO ESTABLISH A SEARCH *
/* ARGUMENT FOR SCANNING THE TABLE TELBOOK *
/*****

```

```

ISPEXEC TBSARG &TELBOOK

```

```

/*****
/* CALL THE TBSCAN SERVICE TO SEARCH THE TABLE TELBOOK *
/* FOR A ROW WITH VALUES THAT MATCH THE ARGUMENT LIST *
/*****

```

```

ISPEXEC TBSCAN &TELBOOK
END
ELSE IF &SERVN = RESCAN THEN DO
ISPEXEC TBSCAN &TELBOOK

```

```

/*****
/* CALL THE TBVCLEAR SERVICE TO CLEAR FROM THE SCREEN *
/* THE VARIABLES DISPLAYED FROM THE TABLE TELBOOK IF *
/* ANOTHER ROW WITH THE SAME SEARCH VALUES IS NOT FOUND. *
/*****

```

```

IF &LASTCC = 8 THEN ISPEXEC TBVCLEAR &TELBOOK
END

```



```

/*****
/*   CALL THE VPUT SERVICE TO                               *
/*   COPY THE VALUES LISTED IN VARLIST FROM THIS PROGRAMS SET *
/*   OF FUNCTION VARIABLES TO THE SHARED VARIABLE POOL       *
/*****

```

```

ISPEXEC VPUT (&VARLIST) SHARED
ISPEXEC VPUT (&ZMSGNAME) SHARED

```

EXEC2 IMPLEMENTATION

The components listed under "Dialog Components Applicable to All Implementations," and the following are required for the EXEC2 implementation of the dialog:

- Combined selection and data entry panel CPANEL1
- Function TELEBOOK
- Function TELESRVC

EXEC2 - COMBINED SELECTION AND DATA ENTRY PANEL CPANEL1

Definition of combined selection and data entry panel CPANEL1 is shown below, followed by the panel as displayed.

Panel Definition

```

)ATTR
  @ TYPE(INPUT) INTENS(LOW) JUST(LEFT) PAD(' ') CAPS(ON)
  @ TYPE(INPUT) INTENS(HIGH) JUST(LEFT) PAD(' ') CAPS(ON)

)BODY
%----- TELEPHONE BOOK APPLICATION -----
+SELECT AN OPTION:%==>@ZCMD
%
% A+- Add new entry
% C+- Change existing entry %F+- Refresh the table, from disk
% D+- Delete old entry %K+- Remove the table, from disk
% G+- Get last entry %L+- List the table
% M+- Add/Change entry %Q+- Query the table
% Z+- Blank screen entry %S+- Scan the table, NON UNIQUE
% P+- Prepare report %R+- Rescan the table, continue scan
% B+- Browse report %T+- Search on last name, unique key
% E+- Edit report %V+- Save the table, on disk
% Y+- Destroy report
%
+Current values / enter search values:
%
+ FIRST NAME ==>_TFNAM +

```

```

+ LAST NAME ==> TLNAM      +
+ LOCATION  ==> TLOC       +
+ PHONE NUM  ==> TNUM      +
+ PHONE COLOR==> TCOLOR    +
+ PHONE TYPE ==> TTYPE     +

```

```
)INIT
```

```

.CURSOR = ZCMD
&ZCMD = &OTELOPT
IF(&ZERRMSG ^= ' ')
  .MSG = &ZERRMSG
  &ZERRMSG = ' '

```

```
)PROC
```

```

&ZSEL = TRANS( &ZCMD
               L, 'CMD(TELESRVC LIST )'
               A, 'CMD(TELESRVC ADD )'
               C, 'CMD(TELESRVC CHANGE )'
               D, 'CMD(TELESRVC DELETE )'
               G, 'CMD(TELESRVC GETLAST )'
               M, 'CMD(TELESRVC MODIFY )'
               Z, 'CMD(TELESRVC BLANK )'
               S, 'CMD(TELESRVC SCAN )'
               R, 'CMD(TELESRVC RESCAN )'
               T, 'CMD(TELESRVC SEARCH )'
               Q, 'CMD(TELESRVC QUERY )'
               P, 'CMD(TELESRVC PREPREP )'
               B, 'CMD(TELESRVC BROWSE )'
               E, 'CMD(TELESRVC EDIT )'
               Y, 'CMD(TELESRVC DESTROY )'
               F, 'CMD(TELESRVC REFRESH )'
               K, 'CMD(TELESRVC REMOVE )'
               V, 'CMD(TELESRVC SAVE )'
               , ' ' )
* , '?' )

```

```

&OSEL = &ZSEL
IF(&ZCMD = S)
  &ZCMD = R
&OTELOPT = &ZCMD

```

```

IF(&ZCMD ^= R)
  VER(&TNUM, PICT, 'NNN-NNNN')

```

```
)END
```

Panel Display

```
----- TELEPHONE BOOK APPLICATION -----
SELECT AN OPTION: ==> A_

A - Add new entry
C - Change existing entry
D - Delete old entry
G - Get last entry
M - Add/Change entry
Z - Blank screen entry
P - Prepare report
B - Browse report
E - Edit report
Y - Destroy report

F - Refresh the table, from disk
K - Remove the table, from disk
L - List the table
Q - Query the table
S - Scan the table, NON UNIQUE
R - Rescan the table, continue scan
T - Search on last name, unique key
V - Save the table, on disk

Current values / enter search values:

FIRST NAME ==> RICHARD
LAST NAME ==> THOMPSON
LOCATION ==> KITCHEN
PHONE NUM ==> 332-4812
PHONE COLOR==> BLACK
PHONE TYPE ==> WALL TOUCH
```

EXEC2 - FUNCTION TELEBOOK

&TRACE

```
*****
*
* THE OBJECTIVE OF THE CODING TECHNIQUE OF THIS APPLICATION IS
* TO ILLUSTRATE ISPF SERVICES
*
* THE APPLICATION CONSISTS OF
* THIS MAIN PROGRAM - TELEBOOK
* 1 SUB PROGRAM - TELESRVC
* 3 PANELS - CPANEL1 TPANEL2 TPANEL3
* 15 MESSAGES - TELB010 THRU TELB024
* 1 FILE TAILORING SKELETON - TELBOOK
*
* THE APPLICATION PROVIDES THE FOLLOWING FUNCTIONS
* FROM PANEL CPANEL1 TO ACCESS A TELEPHONE BOOK
*
* A ADD NEW ENTRY
* C CHANGE EXISTING ENTRY F REFRESH THE TABLE, FROM DISK
* D DELETE OLD ENTRY K REMOVE THE TABLE, FROM DISK
* G GET LAST ENTRY L LIST THE TABLE
* M ADD/CHANGE ENTRY Q QUERY THE TABLE
*
```

```

* Z BLANK SCREEN ENTRY          S SCAN THE TABLE, NON UNIQUE      *
* P PREPARE REPORT              R RESCAN THE TABLE, CONTINUE SCAN  *
* B BROWSE REPORT               T SEARCH ON LAST NAME, UNIQUE KEY  *
* E EDIT REPORT                 V SAVE THE TABLE, ON DISK        *
* Y DESTROY REPORT              *

```

```

*****

```

```

&PRESUME &SUBCOMMAND ISPEXEC2
* &TERMSG IS THE TERMINATION MESSAGE
&TERMSG = &CONCAT OF MSG ( TELB024 )
&TABKEY = &CONCAT OF ( TLNAM )
&TABVAR1 = &CONCAT OF ( TFNAM &BLANK TLOC &BLANK TNUM
&TABVAR2 = &CONCAT OF &BLANK TCOLOR &BLANK TTYPE )
&TABVARS = &CONCAT OF &TABVAR1 &TABVAR2
&TELBOOK = TELBOOK
&VPUTLI1 = &CONCAT OF ( TFNAM &BLANK TLNAM &BLANK TLOC
&VPUTLI2 = &CONCAT OF &BLANK TNUM &BLANK TCOLOR &BLANK TTYPE )
&VPUTLIS = &CONCAT OF &VPUTLI1 &VPUTLI2

```

```

*****
* CALL THE CONTROL SERVICE TO *
* SET THE ERROR PROCESSING MODE FOR THIS PROGRAM TO *
* TERMINATE ONLY FOR ANY ERROR THAT HAS A RETURN CODE OF *
* 12 OR HIGHER *
*****

```

ISPEXEC2 CONTROL ERRORS CANCEL

```

*****
* CALL THE TBOPEN SERVICE TO *
* OPEN THE TABLE, TELBOOK IN WRITE MODE *
*****

```

ISPEXEC2 TBOPEN &TELBOOK WRITE
&IF &RC NE 8 &GOTO -NONOTTO

```

*****
* CALL THE TBCREATE SERVICE TO CREATE TELBOOK IF IT DID NOT *
* EXIST. THE TABLE KEY VARIABLE IS SPECIFIED IN TABKEY *
* AND THE NON-KEY VARIABLES ARE SPECIFIED IN TABVARS. *
*****

```

ISPEXEC2 TBCREATE &TELBOOK KEYS &TABKEY NAMES &TABVARS
-NONOTTO

```

*****
* CALL THE TBSKIP SERVICE TO MOVE THE CURRENT ROW POINTER *
* (CRP) OF THE TABLE TO THE 1ST ROW. THE TABLE VARIABLES *
* ARE SET TO THE VALUES CONTAINED IN THE 1ST ROW. *
*****

```

ISPEXEC2 TBSKIP &TELBOOK

```
*****
*   SET THE TABLE VARIABLES IN THE SHARED VARIABLE POOL TO THE *
*   VALUES CONTAINED IN THE 1ST ROW OF THE TABLE.  THESE VALUES *
*   WILL BE USED TO INITIALIZE THE VALUES DISPLAYED IN CPANEL1. *
*****
```

ISPEXEC2 VPUT &VPUTLIS SHARED

```
*****
*   CALL THE SELECT SERVICE TO                               *
*   CONTINUE TO DISPLAY PANEL CPANEL1 AND CONTROL THE       *
*   INVOCATION OF THE SUB PROGRAM TELESRVC                   *
*****
```

ISPEXEC2 SELECT PANEL(CPANEL1)

```
*****
*   CALL THE TBCLOSE SERVICE TO CLOSE THE TABLE, TELBOOK *
*****
```

ISPEXEC2 TBCLOSE &TELBOOK

```
*****
*   CALL THE LOG SERVICE TO                                 *
*   PUT THE MESSAGE TELB024 ON THE ISPF LOG FILE           *
*****
```

ISPEXEC2 LOG &TERMSG

```
*****
*   CALL THE SETMSG SERVICE TO                             *
*   PUT THE MESSAGE TELB024 ON THE NEXT SCREEN DISPLAYED *
*****
```

ISPEXEC2 SETMSG &TERMSG
&EXIT

EXEC2 - FUNCTION TELESRVC

&TRACE

```
*****
*   THIS SUBPROGRAM IS INVOKED FROM PANEL TPANEL1 DISPLAYED *
*   BY THE MAIN PROGRAM TELEBOOK USING THE SELECT SERVICE *
*****
```

&PRESUME &SUBCOMMAND ISPEXEC
&TPANEL2 = &CONCAT OF PANEL (TPANEL2)
&TPANEL3 = &CONCAT OF PANEL (TPANEL3)
&TELBOOK = TELBOOK
&TELOUT = TELOUT

```

*****
*   &FTOMAC SHOULD BE SET TO THE FILEID OF THE USER'S           *
*   FILE TAILORING OUTPUT MACLIB.                               *
*****

```

```

&FTOMAC = &CONCAT OF ( &BLANK FTOUTPUT &BLANK MACLIB &BLANK )
&TABKEY = &CONCAT OF ( TLNAM )
&TABVAR1 = &CONCAT OF ( TFNAM &BLANK TLOC &BLANK TNUM
&TABVAR2 = &CONCAT OF &BLANK TCOLOR &BLANK TTYPE )
&TABVARS = &CONCAT OF &TABVAR1 &TABVAR2
&ZERRMSG = &BLANK
&ZMSGNA = &CONCAT OF ( ZERRMSG )
&QROWSD = 0
&QROWSDZ = 0
&L8 = 8

```

```

*****
*   SET &SERVN TO THE FUNCTION REQUESTED BY THE                 *
*   USER ON PANEL CPANEL1.                                     *
*****

```

```

&SERVN = &1
&VARLIS1 = &CONCAT OF ( TFNAM &BLANK TLNAM &BLANK TLOC &BLANK TNUM
&VARLIS2 = &CONCAT OF &BLANK TCOLOR &BLANK TTYPE )
&VARLIST = &CONCAT OF &VARLIS1 &VARLIS2

```

```

*****
*   CALL THE VGET SERVICE TO                                     *
*   COPY THE VALUES LISTED IN VARLIST FROM THE SHARED         *
*   VARIABLE POOL TO THIS EXEC'S SET OF FUNCTION VARIABLES.   *
*   THE VALUES IN THE SHARED POOL ARE THOSE RETURNED FROM    *
*   THE SELECT PANEL CPANEL1.                                  *
*****

```

```
ISPEXEC VGET &VARLIST SHARED
```

```

*****
*   CHOOSE APPROPRIATE SERVICE ROUTINE                         *
*****

```

```

&IF &SERVN = LIST      &GOTO -LISTAB
&IF &SERVN = ADD       &GOTO -ADDLINE
&IF &SERVN = CHANGE    &GOTO -CHGLINE
&IF &SERVN = DELETE    &GOTO -DELINE
&IF &SERVN = GETLAST   &GOTO -GLLINE
&IF &SERVN = MODIFY    &GOTO -MODLINE
&IF &SERVN = BLANK     &GOTO -BLSCR
&IF &SERVN = SCAN      &GOTO -SCANTAB
&IF &SERVN = RESCAN    &GOTO -RESCANT
&IF &SERVN = SEARCH    &GOTO -SEARCHT
&IF &SERVN = QUERY     &GOTO -QTAB
&IF &SERVN = PREPREP   &GOTO -PREPREP
&IF &SERVN = BROWSE    &GOTO -BRREP

```

```
&IF &SERVN = EDIT      &GOTO -EDREP
&IF &SERVN = DESTROY  &GOTO -DESREP
&IF &SERVN = REFRESH  &GOTO -REFREP
&IF &SERVN = REMOVE   &GOTO -REMTAB
&IF &SERVN = SAVE     &GOTO -SAVTAB
```

-RITORNO

```
*****
*   CALL THE VPUT SERVICE TO           *
*   COPY THE VALUES LISTED IN VARLIST *
*   FROM THIS PROGRAMS SET OF FUNCTION *
*   VARIABLES                           *
*   TO THE SHARED VARIABLE POOL        *
*****
```

```
ISPEXEC VPUT &VARLIST SHARED
ISPEXEC VPUT &ZMSGNA SHARED
&EXIT
```

```
*****
*   SERVICE ROUTINES                   *
*****
```

-LISTAB

```
*****
*   CALL THE TBTOP SERVICE TO           *
*   SET THE CURRENT ROW POINTER (CRP)   *
*   AHEAD OF THE FIRST ROW OF THE TABLE *
*   TELBOOK                             *
*****
```

```
ISPEXEC TBTOP &TELBOOK
```

```
*****
*   CALL THE TBDISPL SERVICE TO         *
*   DISPLAY THE TABLE TELBOOK USING THE *
*   TABLE TELBOOK.                     *
*****
```

```
ISPEXEC TBDISPL &TELBOOK &TPANEL2
&GOTO -RITORNO
```

-ADDLINE

```
*****
*   CALL THE TBADD SERVICE TO TRY TO    *
*   ADD A NEW ROW OF VARIABLES TO THE TABLE *
*   TELBOOK                             *
*****
```

```
ISPEXEC TBADD &TELBOOK
&IF &RC NE 0 &GOTO -ALO
&ZERRMSG = TELB013
&GOTO -RITORNO
-ALO
&ZERRMSG = TELB010
&GOTO -RITORNO
```

-CHGLINE

```
*****
*   CALL THE TBPOT SERVICE TO TRY TO   *
*   UPDATE A ROW OF VARIABLES IN THE TABLE TELBOOK   *
*****
```

```
ISPEXEC TBPOT &TELBOOK
&IF &RC NE 0 &GOTO -CLO
&ZERRMSG = TELB015
&GOTO -RITORNO
-CLO
&ZERRMSG = TELB018
&GOTO -RITORNO
```

-DELIN

```
*****
*   CALL THE TBDELETE SERVICE TO TRY TO   *
*   DELETE A ROW OF VARIABLES IN THE TABLE TELBOOK   *
*****
```

```
ISPEXEC TBDELETE &TELBOOK
&IF &RC EQ 8 &GOTO -DLO
&ZERRMSG = TELB012
ISPEXEC TBSKIP &TELBOOK NUMBER ( 0 )
&IF &RC NE 8 &GOTO -RITORNO
ISPEXEC TBSKIP &TELBOOK NUMBER ( 1 )
&GOTO -RITORNO
-DLO
&ZERRMSG = TELB011
&GOTO -RITORNO
```

-GLLINE

```
*****
*   CALL THE TBBOTTOM SERVICE TO SET THE CURRENT ROW   *
*   POINTER (CRP) TO THE LAST ROW OF THE TABLE TELBOOK   *
*   AND TO RETRIEVE THAT ROW   *
*****
```

```
ISPEXEC TBBOTTOM &TELBOOK
&IF &RC NE 0 &GOTO -GLO
&ZERRMSG = TELB016
&GOTO -RITORNO
-GLO
&ZERRMSG = TELB017
&GOTO -RITORNO
```


-MODLINE

```
*****  
*   CALL THE TBMOD SERVICE TO                               *  
*   UPDATE OR ADD A ROW OF VARIABLES TO THE TABLE TELBOOK *  
*****
```

```
ISPEXEC TBMOD &TELBOOK  
&IF &RC NE 0 &GOTO -MLO  
&ZERRMSG = TELB014  
&GOTO -RITORNO  
-MLO  
&ZERRMSG = TELB013  
&GOTO -RITORNO
```

-BLSCR

```
*****  
*   SET THE CURRENT VALUES OF THE TABLE VARIABLES TO NULL. *  
*   WHEN THE SELECT PANEL CPANEL1 IS REDISPLAYED, THE TABLE *  
*   VARIABLES WILL BE DISPLAYED AS NULLS ON THE SCREEN.      *  
*****
```

```
ISPEXEC TBVCLEAR &TELBOOK  
&GOTO -RITORNO
```

-SCANTAB

```
*****  
*   CALL THE TBSARG SERVICE TO                               *  
*   ESTABLISH A SEARCH ARGUMENT FOR SCANNING THE TABLE TELBOOK *  
*****
```

```
ISPEXEC TBTOP &TELBOOK  
ISPEXEC TBSARG &TELBOOK
```

```
*****  
*   CALL THE TBSCAN SERVICE TO SEARCH THE TABLE TELBOOK    *  
*   FOR A ROW WITH VALUES THAT MATCH THE ARGUMENT LIST      *  
*****
```

```
ISPEXEC TBSCAN &TELBOOK  
&GOTO -RITORNO
```

-RESCANT

```
ISPEXEC TBSCAN &TELBOOK  
&IF &RC NE 8 &GOTO -RS0
```

```
*****
*   CALL THE TBVCLEAR SERVICE TO CLEAR FROM THE SCREEN THE   *
*   VARIABLES DISPLAYED FROM THE TABLE TELBOOK IF ANOTHER  *
*   ROW IS NOT FOUND USING THE SAME SEARCH ARGUMENTS.       *
*****
```

```
ISPEXEC TBVCLEAR &TELBOOK
-RSO
&GOTO -RITORNO
```

```
-SEARCHT
```

```
*****
*   CALL THE TBEXIST SERVICE TO                               *
*   TEST FOR THE EXISTENCE OF A ROW WITH THE KEY VARIABLE    *
*   (LAST NAME) SPECIFIED ON THE SELECT PANEL CPANEL1.      *
*****
```

```
ISPEXEC TBEXIST &TELBOOK
&IF &RC NE 0 &GOTO -RITORNO
```

```
*****
*   IF THE ROW IS FOUND, CALL THE TBGET SERVICE TO FETCH     *
*   THE ROW FROM THE TABLE. OTHERWISE, THE VALUE OF THE    *
*   TABLE VARIABLES REMAINS UNCHANGED.                     *
*****
```

```
ISPEXEC TBGET &TELBOOK
&GOTO -RITORNO
-QTAB
```

```
*****
*   CALL THE TBQUERY SERVICE TO OBTAIN FROM THE TABLE TELBOOK *
*   KEY VARIABLE NAME(S), NON-KEY VARIABLE NAMES, NUMBER OF ROWS*
*****
```

```
ISPEXEC TBQUERY &TELBOOK KEYS (QKEYS) NAMES (QNAMES) ROWNUM (QROWS)
```

```
*****
*   CALL THE DISPLAY SERVICE TO DISPLAY THE PANEL TPANEL3   *
*****
```

```
ISPEXEC DISPLAY &TPANEL3
&GOTO -RITORNO
```

```
-PREPREP
```

```
*****
*   CALL THE FTOPEN SERVICE TO BEGIN FILE TAILORING          *
*****
```

```
ISPEXEC FTOPEN
```

```
*****
*   CALL THE FTINCL SERVICE TO                               *
*   DO FILE TAILORING USING THE FILE TAILORING SKELETON TELBOOK *
*****
```

ISPEXEC FTINCL &TELBOOK

```
*****
*   CALL THE FTCLOSE SERVICE TO END FILE TAILORING AND       *
*   STORE THE OUTPUT AS MEMBER TELOUT                         *
*****
```

ISPEXEC FTCLOSE NAME (&TELOUT)
&ZERRMSG = TELB022
&GOTO -RITORNO

-BRREP

```
*****
*   CALL THE BROWSE SERVICE TO BROWSE TELOUT IN THE USER'S   *
*   FILE TAILORING OUTPUT MACLIB. THIS FUNCTION IS VALID     *
*   ONLY IF THE ISPF/PROGRAM DEVELOPMENT FACILITY PROGRAM    *
*   PRODUCT IS INSTALLED ON THE USER'S SYSTEM.              *
*****
```

ISPEXEC BROWSE FILE &FTOMAC MEMBER (&TELOUT)
&GOTO -RITORNO

-EDREP

```
*****
*   CALL THE EDIT SERVICE TO EDIT TELOUT IN THE USER'S     *
*   FILE TAILORING OUTPUT MACLIB. THIS FUNCTION IS VALID     *
*   ONLY IF THE ISPF/PROGRAM DEVELOPMENT FACILITY PROGRAM    *
*   PRODUCT IS INSTALLED ON THE USER'S SYSTEM.              *
*****
```

ISPEXEC EDIT FILE &FTOMAC MEMBER (&TELOUT)
&GOTO -RITORNO

-DESREP

```
*****
*   CALL THE FTERASE SERVICE TO                               *
*   DELETE THE FILE TAILORING OUTPUT TELOUT                   *
*****
```

ISPEXEC FTERASE &TELOUT
&ZERRMSG = TELB023
&GOTO -RITORNO

-REFREP

```
*****  
*   CALL THE TBEND SERVICE TO                               *  
*   CLOSE THE TABLE TELBOOK WITHOUT SAVING IT ON DISK    *  
*****
```

```
ISPEXEC TBEND &TELBOOK  
ISPEXEC TBOPEN &TELBOOK  
ISPEXEC TBSKIP &TELBOOK  
&ZERRMSG = TELB021  
&GOTO -RITORNO
```

-REMTAB

```
ISPEXEC TBCLOSE &TELBOOK
```

```
*****  
*   CALL THE TBERASE SERVICE TO                             *  
*   DELETE THE TABLE TELBOOK FROM THE TABLE OUTPUT LIBRARY *  
*****
```

```
ISPEXEC TBERASE &TELBOOK  
ISPEXEC TBCREATE &TELBOOK KEYS &TABKEY NAMES &TABVARS REPLACE  
&ZERRMSG = TELB020  
&GOTO -RITORNO
```

-SAVTAB

```
*****  
*   CALL THE TBSAVE SERVICE TO WRITE THE TABLE TELBOOK FROM *  
*   VIRTUAL STORAGE TO THE TABLE OUTPUT LIBRARY            *  
*****
```

```
ISPEXEC TBSAVE &TELBOOK  
&ZERRMSG = TELB019  
&GOTO -RITORNO  
*
```

COBOL IMPLEMENTATION

In addition to the components listed under "Dialog Components Applicable to All Implementations," the following components are required for the COBOL implementation of the dialog:

- Combined selection and data entry panel TPANEL1
- Function TELEBOOK
- Function TELESRVC

COBOL - COMBINED SELECTION AND DATA ENTRY PANEL TPANEL1

Defintion and display of combined selection and data entry panel TPANEL1, are shown below.

Panel Definition

```
)ATTR
  _ TYPE(INPUT) INTENS(LOW) JUST(LEFT) PAD(' ') CAPS(ON)
  @ TYPE(INPUT) INTENS(HIGH) JUST(LEFT) PAD(' ') CAPS(ON)

)BODY
%----- Telephone Book Application -----
+SELECT AN OPTION:%==>@ZCMD
%
% A+- Add new entry
% C+- Change existing entry %F+- Refresh the table, from disk
% D+- Delete old entry %K+- Remove the table, from disk
% G+- Get last entry %L+- List the table
% M+- Add/Change entry %Q+- Query the table
% Z+- Blank screen entry %S+- Scan the table, non unique
% P+- Prepare report %R+- Rescan the table, continue scan
% B+- Browse report %T+- Search on last name, unique key
% E+- Edit report %V+- Save the table, on disk
% Y+- Destroy report
%
+Current values / enter search values:
%
+ FIRST NAME ==> _TFNAM +
+ LAST NAME ==> _TLNAM +
+ LOCATION ==> _TLOC +
+ PHONE NUM ==> _TNUM +
+ PHONE COLOR==> _TCOLOR +
+ PHONE TYPE ==> _TTYPE +

)INIT

.CURSOR = ZCMD
&ZCMD = &OTELOPT
```

```

&VARL = '(TNUM TLNAM TFNAM TLOC TTYPE TCOLOR)'
IF(&ZERRMSG ^= ' ')
  .MSG = &ZERRMSG
  &ZERRMSG = ' '
)PROC

&ZSEL = TRANS( &ZCMD
  L, 'PGM(TELESRVC) PARM(/LIST &VARL)'
  A, 'PGM(TELESRVC) PARM(/ADD &VARL)'
  C, 'PGM(TELESRVC) PARM(/CHANGE &VARL)'
  D, 'PGM(TELESRVC) PARM(/DELETE &VARL)'
  G, 'PGM(TELESRVC) PARM(/GETLAST &VARL)'
  M, 'PGM(TELESRVC) PARM(/MODIFY &VARL)'
  Z, 'PGM(TELESRVC) PARM(/BLANK &VARL)'
  S, 'PGM(TELESRVC) PARM(/SCAN &VARL)'
  R, 'PGM(TELESRVC) PARM(/RESCAN &VARL)'
  T, 'PGM(TELESRVC) PARM(/SEARCH &VARL)'
  Q, 'PGM(TELESRVC) PARM(/QUERY &VARL)'
  P, 'PGM(TELESRVC) PARM(/PREPREP &VARL)'
  B, 'PGM(TELESRVC) PARM(/BROWSE &VARL)'
  E, 'PGM(TELESRVC) PARM(/EDIT &VARL)'
  Y, 'PGM(TELESRVC) PARM(/DESTROY &VARL)'
  F, 'PGM(TELESRVC) PARM(/REFRESH &VARL)'
  K, 'PGM(TELESRVC) PARM(/REMOVE &VARL)'
  V, 'PGM(TELESRVC) PARM(/SAVE &VARL)'
  , ' '
  *, '? ' )

&OSEL = &ZSEL
IF(&ZCMD = S)
  &ZCMD = R
&OTELOPT = &ZCMD

IF(&ZCMD ^= R)
  VER(&TNUM, PICT, 'NNN-NNNN')
)END

```

Panel Display

```
----- TELEPHONE BOOK APPLICATION -----
SELECT AN OPTION: ==> C_

  A - Add new entry
  C - Change existing entry
  D - Delete old entry
  G - Get last entry
  M - Add/Change entry
  Z - Blank screen entry
  P - Prepare report
  B - Browse report
  E - Edit report
  Y - Destroy report

  F - Refresh the table, from disk
  K - Remove the table, from disk
  L - List the table
  Q - Query the table
  S - Scan the table, non unique
  R - Rescan the table, continue scan
  T - Search on last name, unique key
  V - Save the table, on disk

Current values / enter search values:

FIRST NAME ==> LINDA
LAST NAME  ==> MILLER
LOCATION     ==> KITCHEN
PHONE NUM  ==> 462-4441
PHONE COLOR==> RED
PHONE TYPE ==> WALL DIAL
```

COBOL - FUNCTION TELEBOOK

ID DIVISION.

PROGRAM-ID. TELEBOOK.

* THE OBJECTIVE OF THE CODING TECHNIQUE OF THIS APPLICATION IS
* TO ILLUSTRATE ISPF SERVICES

* THE APPLICATION CONSISTS OF
* THIS MAIN PROGRAM - TELEBOOK
* 1 SUB PROGRAM - TELESRVC
* 3 PANELS - TPANEL1 TPANEL2 TPANEL3
* 15 MESSAGES - TELB010 THRU TELB024
* 1 FILE TAILORING SKELETON - TELBOOK

* THE APPLICATION PROVIDES THE FOLLOWING FUNCTIONS
* FROM PANEL TPANEL1 TO ACCESS A TELEPHONE BOOK

* A ADD NEW ENTRY
* C CHANGE EXISTING ENTRY F REFRESH THE TABLE, FROM DISK
* D DELETE OLD ENTRY K REMOVE THE TABLE, FROM DISK
* G GET LAST ENTRY L LIST THE TABLE
* M ADD/CHANGE ENTRY Q QUERY THE TABLE
* Z BLANK SCREEN ENTRY S SCAN THE TABLE, NON UNIQUE

```

* P PREPARE REPORT          R RESCAN THE TABLE, CONTINUE SCAN
* B BROWSE REPORT          T SEARCH ON LAST NAME, UNIQUE KEY
* E EDIT REPORT            V SAVE THE TABLE, ON DISK
* Y DESTROY REPORT

```

```

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.

```

```

01 CONTROLE      PIC X(8) VALUE 'CONTROL ' .
01      ERRORS   PIC X(8) VALUE 'ERRORS ' .
01      CANCEL   PIC X(8) VALUE 'CANCEL ' .
01 LOG          PIC X(8) VALUE 'LOG ' .
01      TERMINATION-MESSAGE PIC X(8) VALUE 'TELBO24' .
01 SELEKT       PIC X(8) VALUE 'SELECT ' .
01      L15     PIC 9(9) VALUE 15 COMP.
01      COMMAND-STREAM     PIC X(15) VALUE 'PANEL(TPANEL1)' .
01 SETMSG       PIC X(8) VALUE 'SETMSG ' .
01 TBCLOSE      PIC X(8) VALUE 'TBCLOSE ' .
01 TBCREATE     PIC X(8) VALUE 'TBCREATE' .
01      TABKEY   PIC X(7) VALUE '(TLNAM)' .
01      TABVARS  PIC X(30) VALUE
      '(TFNAM TLOC TNUM TCOLOR TTYPE)' .
01 TBOPEN       PIC X(8) VALUE 'TBOPEN ' .
01      TELBOOK  PIC X(8) VALUE 'TELBOOK ' .
01      WRYTE    PIC X(8) VALUE 'WRITE ' .
01 TBSKIP       PIC X(8) VALUE 'TBSKIP ' .
01 VPUT         PIC X(8) VALUE 'VPUT ' .
01      VPUTLIST PIC X(36) VALUE
      '(TFNAM TLNAM TLOC TNUM TCOLOR TTYPE)' .
01      SHARED   PIC X(8) VALUE 'SHARED ' .
01 VRESET       PIC X(8) VALUE 'VRESET ' .

```

```

PROCEDURE DIVISION.

```

```

*****
* CALL THE CONTROL SERVICE TO *
* SET THE ERROR PROCESSING MODE FOR THIS PROGRAM TO *
* TERMINATE ONLY FOR ANY ERROR THAT HAS A RETURN CODE OF *
* 12 OR HIGHER *
*****

```

```

CALL 'ISPLINK' USING CONTROLE ERRORS CANCEL

```

```

*****
* CALL THE TBOPEN SERVICE TO *
* OPEN THE TABLE, TELBOOK IN WRITE MODE *
*****

```

```

CALL 'ISPLINK' USING TBOPEN TELBOOK WRYTE
IF RETURN-CODE = 8

```



```

*****
*   CALL THE TBCREATE SERVICE TO                               *
*   CREATE THE TABLE TELBOOK IF IT DID NOT EXIST.  THE       *
*   KEY VARIABLE IS SPECIFIED IN TABKEY AND THE NON-KEY       *
*   VARIABLES ARE SPECIFIED IN TABVARS.                        *
*****

```

CALL 'ISPLINK' USING TBCREATE TELBOOK TABKEY TABVARS.

```

*****
*   CALL THE TBSKIP SERVICE TO                                 *
*   MOVE THE CURRENT ROW POINTER (CRP) OF THE TABLE TELBOOK *
*   FORWARD ONE POSITION (DEFAULT).  THE TABLE VARIABLES     *
*   WILL BE SET TO THE VALUES IN THE FIRST ROW OF THE TABLE *
*   IF THE TABLE WAS JUST CREATED, THE TABLE VARIABLES WILL *
*   CONTAIN NULL VALUES.                                     *
*****

```

CALL 'ISPLINK' USING TBSKIP TELBOOK

```

*****
*   SET THE TABLE VARIABLES IN THE SHARED VARIABLE POOL.  THESE *
*   VALUES WILL BE USED TO INITIALIZE THE VALUES OF THE     *
*   VARIABLES IN THE SELECT PANEL TPANEL1.                    *
*****

```

CALL 'ISPLINK' USING VPUT VPUTLIST SHARED

```

*****
*   CALL THE SELECT SERVICE TO CONTINUE TO                     *
*   DISPLAY PANEL TPANEL1 AND CONTROL THE INVOCATION OF       *
*   THE SUB PROGRAM TELESRVC                                   *
*****

```

CALL 'ISPLINK' USING SELECT L15 COMMAND-STREAM

```

*****
*   CALL THE TBCLOSE SERVICE TO CLOSE THE TABLE, TELBOOK   *
*****

```

CALL 'ISPLINK' USING TBCLOSE TELBOOK

```

*****
*   CALL THE LOG SERVICE TO                                    *
*   PUT THE MESSAGE TELB024 ON THE ISPF LOG FILE              *
*****

```

CALL 'ISPLINK' USING LOG TERMINATION-MESSAGE

```

*****
*   CALL THE SETMSG SERVICE TO                               *
*   PUT THE MESSAGE TELB024 ON THE NEXT SCREEN DISPLAYED    *
*****

```

CALL 'ISPLINK' USING SETMSG TERMINATION-MESSAGE

```

*****
*   CALL THE VRESET SERVICE TO                               *
*   REMOVE THE ISPF ADDRESSABILITY OF ANY VDEFINED VARIABLES *
*****

```

CALL 'ISPLINK' USING VRESET

GOBACK.

COBOL - FUNCTION TELESRVC

ID DIVISION.
PROGRAM-ID. TELESRVC.

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.

```

01 TPANEL2          PIC X(8) VALUE 'TPANEL2 ':
01 TPANEL3          PIC X(8) VALUE 'TPANEL3 ':
01 TELBOOK          PIC X(8) VALUE 'TELBOOK ':
01 TELOUT           PIC X(8) VALUE 'TELOUT ':

```

```

*****
*   FTOUTPUT SHOULD CONTAIN THE NAME OF THE USER'S FILE    *
*   TAILORING OUTPUT PARTITIONED DATASET WITH THE MEMBER   *
*   SPECIFIED AS TELOUT.                                    *
*                                                           *
*   FOR VM, FTOUTPUT SHOULD SPECIFY THE FILEID OF THE FILE *
*   TAILORING OUTPUT MACLIB WITHOUT A MEMBER NAME.  EXAMPLE: *
*                                                           *
*01 FTOUTPUT        PIC X(18) VALUE '(FTOUTPUT MACLIB )'. *
*                                                           *
*****

```

```

01 FTOUTPUT          PIC X(30) VALUE
                    ' ' ISPFPROJ.FTOUTPUT(TELOUT) ' ' '.
01 TABKEY            PIC X(7) VALUE '(TLNAM)'.
01 TABVARS           PIC X(30) VALUE
                    '(TFNAM TLOC TNUM TCOLOR TTYPE)'.
01 BROWSE            PIC X(8) VALUE 'BROWSE '.
01 DISPLAYE         PIC X(8) VALUE 'DISPLAY '.

```

```

01 EDIT          PIC X(8)  VALUE 'EDIT      ' .
01 FTCLOSE      PIC X(8)  VALUE 'FTCLOSE  ' .
01 FTERASE      PIC X(8)  VALUE 'FTERASE  ' .
01 FTINCL       PIC X(8)  VALUE 'FTINCL   ' .
01 FTOPEN       PIC X(8)  VALUE 'FTOPEN   ' .
01 TBADD        PIC X(8)  VALUE 'TBADD    ' .
01 TBBOTTOM     PIC X(8)  VALUE 'TBBOTTOM ' .
01 TBCLOSE      PIC X(8)  VALUE 'TBCLOSE  ' .
01 TBCREATE     PIC X(8)  VALUE 'TBCREATE ' .
01 TBDELETE     PIC X(8)  VALUE 'TBDELETE ' .
01 TBDISPL      PIC X(8)  VALUE 'TBDISPL ' .
01 TBEND        PIC X(8)  VALUE 'TBEND    ' .
01 TBERASE      PIC X(8)  VALUE 'TBERASE  ' .
01 TBEXIST      PIC X(8)  VALUE 'TBEXIST  ' .
01 TBGET        PIC X(8)  VALUE 'TBGET    ' .
01 TBMOD        PIC X(8)  VALUE 'TBMOD    ' .
01 TBOPEN       PIC X(8)  VALUE 'TBOPEN   ' .
01 TBPUT        PIC X(8)  VALUE 'TBPUT    ' .
01 TBQUERY      PIC X(8)  VALUE 'TBQUERY  ' .
01 TBSARG       PIC X(8)  VALUE 'TBSARG   ' .
01 TBSAVE       PIC X(8)  VALUE 'TBSAVE   ' .
01 TBSCAN       PIC X(8)  VALUE 'TBSCAN   ' .
01 TBSKIP       PIC X(8)  VALUE 'TBSKIP   ' .
01 TBTOP        PIC X(8)  VALUE 'TBTOP    ' .
01 TBVCLEAR     PIC X(8)  VALUE 'TBVCLEAR ' .
01 VCOPY        PIC X(8)  VALUE 'VCOPY    ' .
01 VDEFINE      PIC X(8)  VALUE 'VDEFINE  ' .
01 VDELETE      PIC X(8)  VALUE 'VDELETE  ' .
01 VGET         PIC X(8)  VALUE 'VGET     ' .
01 VPUT         PIC X(8)  VALUE 'VPUT     ' .
01 VREPLACE     PIC X(8)  VALUE 'VREPLACE ' .
01 ZERRMSG      PIC X(8)  VALUE SPACE .
01 ZMSGNAME     PIC X(9)  VALUE '(ZERRMSG)' .
01 QKEYS        PIC X(8)  VALUE 'QKEYS   ' .
01 QNAMES       PIC X(8)  VALUE 'QNAMES  ' .
01 QROWS        PIC X(8)  VALUE 'QROWS   ' .
01 QROWSDATA    PIC 9(8)  VALUE ZERO .
01 QROWSDATAZ   PIC Z(8) .
01 CHAR         PIC X(8)  VALUE 'CHAR    ' .
01 MOOV         PIC X(8)  VALUE 'MOVE    ' .
01 SHARED       PIC X(8)  VALUE 'SHARED  ' .
01 ZEERO        PIC 9(6)  VALUE 0  COMP .
01 ONE          PIC 9(6)  VALUE 1  COMP .
01 L8           PIC 9(6)  VALUE 8  COMP .

```

```
01 PARM-WORK.
```

```

04 PARM-LENGTH  PIC S9(4)  COMP .
04 FILLER-SLASH PIC X .
04 SERVN        PIC X(7) .
04 FILLER       PIC X .
04 VARLIST      PIC X(36) .

```

LINKAGE SECTION.

01 PARM PIC X(47).

PROCEDURE DIVISION USING PARM.

```
*****
*   MOVE THE PARAMETER LIST CONTAINING THE FUNCTION REQUESTED   *
*   BY THE USER (SERVN) AND THE LIST OF TABLE VARIABLE NAMES  *
*   PASSED FROM THE SELECT REQUEST IN THE SELECT PANEL TPANEL1 *
*   TO THE PARAMETER WORK AREA.                                  *
*****
```

MOVE PARM TO PARM-WORK

```
*****
*   CALL THE VDEFINE SERVICE TO GIVE ISPF ADDRESSABILITY TO     *
*   ZERRMSG AS DEFINED WITHIN THIS PROGRAM AS A CHARACTER FIELD *
*   WITH A LENGTH OF 8                                         *
*****
```

CALL 'ISPLINK' USING VDEFINE ZMSGNAME ZERRMSG CHAR L8

```
*****
*   CALL THE VGET SERVICE TO                                     *
*   COPY THE VALUES FOR THE TABLE VARIABLES LISTED IN VARLIST *
*   FROM THE SHARED VARIABLE POOL                              *
*   TO THIS PROGRAMS SET OF FUNCTION VARIABLES                  *
*****
```

CALL 'ISPLINK' USING VGET VARLIST SHARED

PERFORM SERVICE-ROUTINES.

```
*****
*   CALL THE VPUT SERVICE TO                                     *
*   COPY THE VALUES LISTED IN VARLIST AND THE MESSAGE NAME    *
*   VALUE FROM THIS PROGRAMS SET OF FUNCTION VARIABLES         *
*   TO THE SHARED VARIABLE POOL                                *
*****
```

CALL 'ISPLINK' USING VPUT VARLIST SHARED
CALL 'ISPLINK' USING VPUT ZMSGNAME SHARED

```
*****
*   CALL THE VDELETE SERVICE TO                                 *
*   REMOVE ISPF ADDRESSABILITY TO ZERRMSG                       *
*   AS PREVIOUSLY VDEFINED WITHIN THIS PROGRAM                 *
*****
```

CALL 'ISPLINK' USING VDELETE ZMSGNAME
GOBACK.

SERVICE-ROUTINES.

```
IF      SERVN = 'LIST  ' PERFORM LIST-TABLE
ELSE IF SERVN = 'ADD   ' PERFORM ADD-LINE
ELSE IF SERVN = 'CHANGE ' PERFORM CHANGE-LINE
ELSE IF SERVN = 'DELETE ' PERFORM DELETE-LINE
ELSE IF SERVN = 'GETLAST' PERFORM GET-LAST-LINE
ELSE IF SERVN = 'MODIFY ' PERFORM MODIFY-LINE
ELSE IF SERVN = 'BLANK  ' PERFORM BLANK-SCREEN
ELSE IF SERVN = 'SCAN   ' PERFORM SCAN-TABLE
ELSE IF SERVN = 'RESCAN ' PERFORM RESCAN-TABLE
ELSE IF SERVN = 'SEARCH ' PERFORM SEARCH-TABLE
ELSE IF SERVN = 'QUERY  ' PERFORM QUERY-TABLE
ELSE IF SERVN = 'PREPREP' PERFORM PREPARE-REPORT
ELSE IF SERVN = 'BROWSE ' PERFORM BROWSE-REPORT
ELSE IF SERVN = 'EDIT   ' PERFORM EDIT-REPORT
ELSE IF SERVN = 'DESTROY' PERFORM DESTROY-REPORT
ELSE IF SERVN = 'REFRESH' PERFORM REFRESH-TABLE
ELSE IF SERVN = 'REMOVE ' PERFORM REMOVE-TABLE
ELSE IF SERVN = 'SAVE   ' PERFORM SAVE-TABLE.
```

LIST-TABLE.

```
*****
*   CALL THE TBTOP SERVICE TO                               *
*   SET THE CURRENT ROW POINTER (CRP)                       *
*   AHEAD OF THE FIRST ROW OF THE TABLE TELBOOK          *
*****
```

```
CALL 'ISPLINK' USING TBTOP TELBOOK
```

```
*****
*   CALL THE TBDISPL SERVICE TO                             *
*   DISPLAY THE TABLE TELBOOK USING THE PANEL TPANEL2    *
*****
```

```
CALL 'ISPLINK' USING TBDISPL TELBOOK TPANEL2.
```

ADD-LINE.

```
*****
*   CALL THE TBADD SERVICE TO TRY TO                       *
*   ADD A NEW ROW OF VARIABLES TO THE TABLE TELBOOK      *
*****
```

```
CALL 'ISPLINK' USING TBADD TELBOOK
IF RETURN-CODE = 0
  MOVE 'TELBO13' TO ZERRMSG
ELSE
  MOVE 'TELBO10' TO ZERRMSG.
```

CHANGE-LINE.

```
*****
* CALL THE TBPOT SERVICE TO TRY TO *
* UPDATE A ROW OF VARIABLES IN THE TABLE TELBOOK *
*****
```

```
CALL 'ISPLINK' USING TBPOT TELBOOK
IF RETURN-CODE = 0
    MOVE 'TELBO15' TO ZERRMSG
ELSE
    MOVE 'TELBO18' TO ZERRMSG.
```

DELETE-LINE.

```
*****
* CALL THE TBDELETE SERVICE TO TRY TO *
* DELETE A ROW OF VARIABLES IN THE TABLE TELBOOK *
*****
```

```
CALL 'ISPLINK' USING TBDELETE TELBOOK
IF RETURN-CODE = 8
    MOVE 'TELBO11' TO ZERRMSG
ELSE
    MOVE 'TELBO12' TO ZERRMSG
    CALL 'ISPLINK' USING TBSKIP TELBOOK ZEERO
    IF RETURN-CODE = 8
        CALL 'ISPLINK' USING TBSKIP TELBOOK ONE.
```

GET-LAST-LINE.

```
*****
* CALL THE TBBOTTOM SERVICE TO *
* SET THE CURRENT ROW POINTER (CRP) *
* TO THE LAST ROW OF THE TABLE TELBOOK *
* AND TO RETRIEVE THAT ROW *
*****
```

```
CALL 'ISPLINK' USING TBBOTTOM TELBOOK
IF RETURN-CODE = 0
    MOVE 'TELBO16' TO ZERRMSG
ELSE
    MOVE 'TELBO17' TO ZERRMSG.
```

MODIFY-LINE.

```
*****
* CALL THE TBMOD SERVICE TO *
* UPDATE OR ADD A ROW OF VARIABLES TO THE TABLE TELBOOK *
*****
```

```
CALL 'ISPLINK' USING TBMOD TELBOOK
IF RETURN-CODE = 0
    MOVE 'TELBO14' TO ZERRMSG
ELSE
    MOVE 'TELBO13' TO ZERRMSG.
```

BLANK-SCREEN.

```
*****
*   CALL THE TBVCLEAR SERVICE TO                               *
*   SET THE CURRENT VALUES OF THE TABLE VARIABLES TO NULL.  *
*   WHEN TPANEL1 IS REDISPLAYED THE TABLE VARIABLE VALUES  *
*   DISPLAYED ON THE SCREEN WILL BE NULL.                      *
*****
```

CALL 'ISPLINK' USING TBVCLEAR TELBOOK.

SCAN-TABLE.

CALL 'ISPLINK' USING TBTOP TELBOOK

```
*****
*   CALL THE TBSARG SERVICE TO                                 *
*   ESTABLISH A SEARCH ARGUMENT FOR SCANNING THE TABLE TELBOOK *
*****
```

CALL 'ISPLINK' USING TBSARG TELBOOK

```
*****
*   CALL THE TBSCAN SERVICE TO                                 *
*   SEARCH THE TABLE TELBOOK FOR A ROW WITH VALUES THAT    *
*   MATCH THE ARGUMENT LIST                                    *
*****
```

CALL 'ISPLINK' USING TBSCAN TELBOOK.

RESCAN-TABLE.

CALL 'ISPLINK' USING TBSCAN TELBOOK
IF RETURN-CODE = 8

```
*****
*   CALL THE TBVCLEAR SERVICE TO                               *
*   CLEAR FROM THE SCREEN THE VARIABLES DISPLAYED FROM THE    *
*   TABLE TELBOOK IF ANOTHER ROW WITH THE SAME SEARCH      *
*   ARGUMENT IS NOT FOUND.                                    *
*****
```

CALL 'ISPLINK' USING TBVCLEAR TELBOOK.

SEARCH-TABLE.

```
*****
*   CALL THE TBEXIST SERVICE TO TEST FOR THE EXISTENCE OF A ROW *
*   IN THE TABLE WITH THE KEY VARIABLE SPECIFIED ON THE SCREEN. *
*****
```

CALL 'ISPLINK' USING TBEXIST TELBOOK

```

IF RETURN-CODE = 0
*****
*   IF THE ROW WAS FOUND, CALL                               *
*   THE TBGET SERVICE TO FETCH THE ROW FROM THE TABLE.     *
*   OTHERWISE THE VALUES OF THE VARIABLES WILL BE UNCHANGED. *
*****

```

```

CALL 'ISPLINK' USING TBGET TELBOOK.

```

```

QUERY-TABLE.

```

```

*****
*   CALL THE TBQUERY SERVICE TO OBTAIN FROM THE TABLE TELBOOK *
*   KEY VARIABLE NAME (S)                                       *
*   NON-KEY VARIABLE NAMES                                     *
*   NUMBER OF ROWS                                           *
*****

```

```

CALL 'ISPLINK' USING TBQUERY TELBOOK QKEYS QNAMES QROWS

```

```

*****
*   CALL THE VCOPY SERVICE TO                                  *
*   OBTAIN A COPY OF THE DIALOG VARIABLE QROWS                *
*****

```

```

CALL 'ISPLINK' USING VCOPY QROWS L8 QROWSDATA MOOV
MOVE QROWSDATA TO QROWSDATAZ

```

```

*****
*   CALL THE VREPLACE SERVICE TO                              *
*   UPDATE THE DIALOG VARIABLE QROWS                          *
*****

```

```

CALL 'ISPLINK' USING VREPLACE QROWS L8 QROWSDATAZ

```

```

*****
*   CALL THE DISPLAY SERVICE TO DISPLAY THE PANEL TPANEL3    *
*****

```

```

CALL 'ISPLINK' USING DISPLAYE TPANEL3.

```

```

PREPARE-REPORT.

```

```

*****
*   CALL THE FTOPEN SERVICE TO BEGIN FILE TAILORING          *
*****

```

```

CALL 'ISPLINK' USING FTOPEN

```



```
*****
*   CALL THE FTINCL SERVICE TO                               *
*   DO FILE TAILORING USING THE FILE TAILORING SKELETON TELBOOK *
*****
```

CALL 'ISPLINK' USING FTINCL TELBOOK

```
*****
*   CALL THE FTCLOSE SERVICE TO                               *
*   TERMINATE FILE TAILORING AND STORE THE OUTPUT AS MEMBER    *
*   TELOUT                                                       *
*****
```

CALL 'ISPLINK' USING FTCLOSE TELOUT
MOVE 'TELBO22' TO ZERRMSG.

BROWSE-REPORT.

```
*****
*   CALL THE BROWSE SERVICE TO                               *
*   BROWSE TELOUT IN THE FILE TAILORING OUTPUT DATASET.      *
*                                                                *
*   FOR VM, FTOUTPUT MUST CONTAIN THE FILEID OF THE FILE     *
*   TAILORING OUTPUT MACLIB. THE MEMBER NAME IS SPECIFIED    *
*   AS A SEPARATE PARAMETER. EXAMPLE: 'CALL 'ISPLINK' USING  *
*   BROWSE FTOUTPUT TELOUT.'. THIS FUNCTION IS AVAILABLE     *
*   ONLY IF THE ISPF/PROGRAM DEVELOPMENT FACILITY PROGRAM    *
*   PRODUCT IS INSTALLED.                                     *
*****
```

CALL 'ISPLINK' USING BROWSE FTOUTPUT.

EDIT-REPORT.

```
*****
*   CALL THE EDIT SERVICE TO                                 *
*   EDIT TELOUT IN THE FILE TAILORING OUTPUT DATASET.      *
*   FOR VM, FTOUTPUT MUST CONTAIN THE FILEID OF THE FILE     *
*   TAILORING OUTPUT MACLIB. THE MEMBER NAME IS SPECIFIED    *
*   AS A SEPARATE PARAMETER. EXAMPLE: 'CALL 'ISPLINK' USING  *
*   EDIT FTOUTPUT TELOUT.'. THIS FUNCTION IS AVAILABLE ONLY  *
*   IF THE ISPF/PROGRAM DEVELOPMENT FACILITY PROGRAM PRODUCT *
*   IS INSTALLED.                                           *
*****
```

CALL 'ISPLINK' USING EDIT FTOUTPUT.

DESTROY-REPORT.

```
*****
*   CALL THE FTERASE SERVICE TO                               *
*   DELETE THE FILE TAILORING OUTPUT TELOUT                  *
*****
```

CALL 'ISPLINK' USING FTERASE TELOUT
MOVE 'TELBO23' TO ZERRMSG.

REFRESH-TABLE.

```
*****
*   CALL THE TBEND SERVICE TO                               *
*   CLOSE THE TABLE TELBOOK WITHOUT SAVING IT ON DISK     *
*****
```

```
CALL 'ISPLINK' USING TBEND TELBOOK
CALL 'ISPLINK' USING TBOPEN TELBOOK
CALL 'ISPLINK' USING TBSKIP TELBOOK.
MOVE 'TELBO21' TO ZERRMSG.
```

REMOVE-TABLE.

```
CALL 'ISPLINK' USING TBCLOSE TELBOOK
```

```
*****
*   CALL THE TBERASE SERVICE TO                             *
*   DELETE THE TABLE TELBOOK FROM THE TABLE OUTPUT LIBRARY *
*****
```

```
CALL 'ISPLINK' USING TBERASE TELBOOK
CALL 'ISPLINK' USING TBCREATE TELBOOK TABKEY TABVARS.
MOVE 'TELBO20' TO ZERRMSG.
```

SAVE-TABLE.

```
*****
*   CALL THE TBSAVE SERVICE TO                               *
*   WRITE THE TABLE TELBOOK FROM VIRTUAL STORAGE          *
*   TO THE TABLE OUTPUT LIBRARY                           *
*****
```

```
CALL 'ISPLINK' USING TBSAVE TELBOOK
MOVE 'TELBO19' TO ZERRMSG.
```

FORTRAN IMPLEMENTATION

In addition to the components listed under "Dialog Components Applicable to All Implementations," the following components are required for the FORTRAN implementation of the dialog:

- Combined selection and data entry panel FPANEL1
- Function TELEBOOK
- Function TELESRVC

FORTTRAN - COMBINED SELECTION AND DATA ENTRY PANEL FANEL1

Definition and display of combined selection and data entry panel FANEL1, are shown below.

Panel Definition

```
)ATTR
  TYPE(INPUT)  INTENS(LOW)    JUST(LEFT)  PAD(' ')  CAPS(ON)
  @ TYPE(INPUT)  INTENS(HIGH)  JUST(LEFT)  PAD(' ')  CAPS(ON)

)BODY
%----- Telephone Book Application -----
+SELECT AN OPTION:%====>@ZCMD
%
% A+- Add new entry
% C+- Change existing entry      %F+- Refresh the table, from disk
% D+- Delete old entry          %K+- Remove the table, from disk
% G+- Get last entry            %L+- List the table
% M+- Add/Change entry          %Q+- Query the table
% Z+- Blank screen entry        %S+- Scan the table, non unique
% P+- Prepare report            %R+- Rescan the table, continue scan
% B+- Browse report             %T+- Search on last name, unique key
% E+- Edit report               %V+- Save the table, on disk
% Y+- Destroy report
%
+Current values / enter search values:
%
+ FIRST NAME ==>_TFNAM          +
+ LAST NAME  ==>_TLNAM          +
+ LOCATION   ==>_TLOC           +
+ PHONE NUM  ==>_TNUM           +
+ PHONE COLOR==>_TCOLOR         +
+ PHONE TYPE ==>_TTYPE         +

)INIT

  .CURSOR = ZCMD
  &ZCMD = &OTELOPT
  &VARL = '(TNUM TLNAM TFNAM TLOC TTYPE TCOLOR)'
  IF(&ZERRMSG -=' ')
    .MSG = &ZERRMSG
    &ZERRMSG = ' '

)PROC

  &OPTSV = TRANS( &ZCMD
                  L,'LIST'
                  A,'ADD '
                  C,'CHAN'
                  D,'DELE'
                  G,'GETL'
```

```

M, 'MODI'
Z, 'BLAN'
S, 'SCAN'
R, 'RESC'
T, 'SEAR'
Q, 'QUER'
P, 'PREP'
B, 'BROW'
E, 'EDIT'
Y, 'DEST'
F, 'REFR'
K, 'REMO'
V, 'SAVE' )

```

```

&ZSEL = TRANS( &ZCMD
L, 'PGM(TELESRVC)'
A, 'PGM(TELESRVC)'
C, 'PGM(TELESRVC)'
D, 'PGM(TELESRVC)'
G, 'PGM(TELESRVC)'
M, 'PGM(TELESRVC)'
Z, 'PGM(TELESRVC)'
S, 'PGM(TELESRVC)'
R, 'PGM(TELESRVC)'
T, 'PGM(TELESRVC)'
Q, 'PGM(TELESRVC)'
P, 'PGM(TELESRVC)'
B, 'PGM(TELESRVC)'
E, 'PGM(TELESRVC)'
Y, 'PGM(TELESRVC)'
F, 'PGM(TELESRVC)'
K, 'PGM(TELESRVC)'
V, 'PGM(TELESRVC)'
, ' ' )
*, '?' )

```

```

&OSEL = &ZSEL
IF(&ZCMD = S)
  &ZCMD = R
&OTELOPT = &ZCMD

```

```

IF(&ZCMD = R)
  VER(&TNUM, PICT, 'NNN-NNNN')

```

```

)END

```

Panel Display

```
----- TELEPHONE BOOK APPLICATION -----
SELECT AN OPTION: ==> M_

A - Add new entry
C - Change existing entry
D - Delete old entry
G - Get last entry
M - Add/Change entry
Z - Blank screen entry
P - Prepare report
B - Browse report
E - Edit report
Y - Destroy report

F - Refresh the table, from disk
K - Remove the table, from disk
L - List the table
Q - Query the table
S - Scan the table, non unique
R - Rescan the table, continue scan
T - Search on last name, unique key
V - Save the table, on disk

Current values / enter search values:

FIRST NAME ==> LARRY
LAST NAME ==> MILNS
LOCATION ==> STUDY
PHONE NUM ==> 385-2222
PHONE COLOR==> WHITE
PHONE TYPE ==> DESK DIAL
```

FORTRAN - FUNCTION TELEBOOK

C THE OBJECTIVE OF THE CODING TECHNIQUE OF THIS APPLICATION IS
C TO ILLUSTRATE ISPF SERVICES

C THE APPLICATION CONSISTS OF
C THIS MAIN PROGRAM (FORTRAN) - TELEBOOK
C 1 SUB PROGRAM (FORTRAN) - TELESRVC
C 3 PANELS - FANEL1 TPANEL2 TPANEL3
C 16 MESSAGES - TELB010 THRU TELB025
C 1 FILE TAILORING SKELETON - TELBOOK
C

C THE APPLICATION PROVIDES THE FOLLOWING FUNCTIONS
C FROM PANEL FANEL1 TO ACCESS A TELEPHONE BOOK

C A	ADD NEW ENTRY		
C C	CHANGE EXISTING ENTRY	F	REFRESH THE TABLE, FROM DISK
C D	DELETE OLD ENTRY	K	REMOVE THE TABLE, FROM DISK
C G	GET LAST ENTRY	L	LIST THE TABLE
C M	ADD/CHANGE ENTRY	Q	QUERY THE TABLE
C Z	BLANK SCREEN ENTRY	S	SCAN THE TABLE, NON UNIQUE
C P	PREPARE REPORT	R	RESCAN THE TABLE, CONTINUE SCAN
C B	BROWSE REPORT	T	SEARCH ON LAST NAME, UNIQUE KEY

C E EDIT REPORT
C Y DESTROY REPORT

V SAVE THE TABLE, ON DISK

```
INTEGER  TABKEY(2)
DATA     TABKEY /'(TLN', 'AM) '/

INTEGER  TABVRS(8)
DATA     TABVRS /'(TFN', 'AM T', 'LOC ', 'TNUM',
&        ' TCO', 'LOR ', 'TTYP', 'E) '/

INTEGER  VPLIST(9)
DATA     VPLIST /'(TFN', 'AM T', 'LNAM', ' TLO', 'C TN',
&        'UM T', 'COLO', 'R TT', 'YPE)'/
```

C CALL THE CONTROL SERVICE TO
C SET THE ERROR PROCESSING MODE FOR THIS PROGRAM TO
C TERMINATE ONLY FOR ANY ERROR THAT HAS A RETURN CODE OF
C 12 OR HIGHER

```
LASTCC = ISPLNK ('CONTROL ', 'ERRORS ', 'CANCEL ')
```

C CALL THE TBOPEN SERVICE TO
C OPEN THE TABLE, TELBOOK IN WRITE MODE

```
LASTCC = ISPLNK ('TBOPEN ', 'TELBOOK ', 'WRITE ')
IF (LASTCC .EQ. 8) GO TO 1001
GO TO 1002
```

C CALL THE TBCREATE SERVICE TO CREATE THE TABLE TELBOOK IF
C IT DID NOT EXIST. THE KEY VARIABLE IS SPECIFIED IN TABKEY
C AND THE NON-KEY VARIABLES ARE SPECIFIED IN TABVARS.

```
1001 LASTCC = ISPLNK ('TBCREATE', 'TELBOOK ', TABKEY, TABVRS)
1002 CONTINUE
```

C CALL THE TBSKIP SERVICE TO MOVE THE CURRENT ROW POINTER
C (CRP) FORWARD ONE POSITION (DEFAULT). THE VARIABLE VALUES
C IN THE FIRST ROW WILL BE READ. IF THE TABLE WAS JUST CREATED,
C THE VARIABLE VALUES WILL REMAIN NULL.

```
LASTCC = ISPLNK ('TBSKIP ', 'TELBOOK ')
```

C SET THE TABLE VARIABLES IN THE SHARED VARIABLE POOL TO THE
C VALUES IN THE 1ST TABLE ROW. THESE VALUES WILL BE USED
C TO INITIALIZE THE VARIABLES IN SELECT PANEL FANEL1 ON
C THE SCREEN.

```
LASTCC = ISPLNK ('VPUT ', VPLIST, 'SHARED ')
```

```

C CALL THE SELECT SERVICE TO CONTINUE TO
C DISPLAY PANEL FPANEL1 AND CONTROL THE INVOCATION OF
C THE SUB PROGRAM TELESV UNTIL FPANEL1 IS TERMINATED

```

```

      LASTCC = ISPLNK ('SELECT ', 15, 'PANEL(FPANEL1) ')

```

```

C CALL THE TBCLOSE SERVICE TO
C CLOSE THE TABLE, TELBOOK

```

```

      LASTCC = ISPLNK ('TBCLOSE ', 'TELBOOK ')

```

```

C CALL THE LOG SERVICE TO
C PUT THE MESSAGE TELB024 ON THE ISPF LOG FILE

```

```

      LASTCC = ISPLNK ('LOG ', 'TELB024 ')

```

```

C CALL THE SETMSG SERVICE TO *
C PUT THE MESSAGE TELB024 ON THE NEXT SCREEN DISPLAYED *

```

```

      LASTCC = ISPLNK ('SETMSG ', 'TELB024 ')

```

```

C CALL THE VRESET SERVICE TO *
C REMOVE THE ISPF ADDRESSABILITY OF ANY VDEFINED VARIABLES *

```

```

      LASTCC = ISPLNK ('VRESET ')
      STOP
      END

```

FORTTRAN - FUNCTION TELESRVC

```

C THIS SUBPROGRAM IS INVOKED FROM PANEL FPANEL1 AND DISPLAYED
C BY THE MAIN PROGRAM TELEBOOK USING THE SELECT SERVICE

```

```

      INTEGER TABKEY(2)
      DATA TABKEY /'(TLN', 'AM) '/

      INTEGER TABVRS(8)
      DATA TABVRS /'(TFN', 'AM T', 'LOC ', 'TNUM', ' TCO',
&                  'LOR ', 'TTYP', 'E) '/

      INTEGER ZERMSG(2)
      DATA ZERMSG /'TELB', '025 '/

      INTEGER LIST
      DATA LIST /'LIST'/
      INTEGER ADD
      DATA ADD /'ADD '/
      INTEGER CHANGE
      DATA CHANGE /'CHAN'/
      INTEGER DELETE

```

DATA	DELETE	/'DELE'/
INTEGER	GETLST	
DATA	GETLST	/'GETL'/
INTEGER	MODIFY	
DATA	MODIFY	/'MODI'/
INTEGER	BLANK	
DATA	BLANK	/'BLAN'/
INTEGER	SCAN	
DATA	SCAN	/'SCAN'/
INTEGER	RESCAN	
DATA	RESCAN	/'RESC'/
INTEGER	SEARCH	
DATA	SEARCH	/'SEAR'/
INTEGER	QUERY	
DATA	QUERY	/'QUER'/
INTEGER	PREPRP	
DATA	PREPRP	/'PREP'/
INTEGER	BROWSE	
DATA	BROWSE	/'BROW'/
INTEGER	EDIT	
DATA	EDIT	/'EDIT'/
INTEGER	DESTRY	
DATA	DESTRY	/'DEST'/
INTEGER	REFRSH	
DATA	REFRSH	/'REFR'/
INTEGER	REMOVE	
DATA	REMOVE	/'REMO'/
INTEGER	SAVE	
DATA	SAVE	/'SAVE'/

INTEGER	M010	
DATA	M010	/'010 '/
INTEGER	M011	
DATA	M011	/'011 '/
INTEGER	M012	
DATA	M012	/'012 '/
INTEGER	M013	
DATA	M013	/'013 '/
INTEGER	M014	
DATA	M014	/'014 '/
INTEGER	M015	
DATA	M015	/'015 '/
INTEGER	M016	
DATA	M016	/'016 '/
INTEGER	M017	
DATA	M017	/'017 '/
INTEGER	M018	
DATA	M018	/'018 '/
INTEGER	M019	
DATA	M019	/'019 '/
INTEGER	M020	
DATA	M020	/'020 '/
INTEGER	M021	


```
DATA      M021  /'021 '/
INTEGER   M022
DATA      M022  /'022 '/
INTEGER   M023
DATA      M023  /'023 '/
```

```
INTEGER   QDATA(2)
```

```
INTEGER   SERVN
```

```
INTEGER   VRLIST(9)
```

```
C COPY THE VARIABLES OPTSV (CONTAINING THE FUNCTION REQUESTED
C BY THE USER ON PANEL FANEL1) AND VARL (CONTAINING THE
C LIST OF THE NAMES OF THE TABLE VARIABLES) TO THE PROGRAM
C VARIABLES SERVN AND VRLIST DEFINED IN THIS PROGRAM.
```

```
LASTCC = ISPLNK ('VCOPY ', 'OPTSV ', 4, SERVN, 'MOVE ')
LASTCC = ISPLNK ('VCOPY ', 'VARL ', 36, VRLIST, 'MOVE ')
```

```
C CALL THE VDEFINE SERVICE TO
C GIVE ISPF ADDRESSABILITY TO ZERMSG
C AS DEFINED WITHIN THIS PROGRAM
C AS A CHARACTER FIELD WITH A LENGTH OF 8
```

```
LASTCC = ISPLNK ('VDEFINE ', 'ZERRMSG ', ZERMSG, 'CHAR ', 8)
```

```
C CALL THE VGET SERVICE TO
C COPY THE VALUES LISTED IN VRLIST
C FROM THE SHARED VARIABLE POOL
C TO THIS PROGRAMS SET OF FUNCTION VARIABLES
```

```
LASTCC = ISPLNK ('VGET ', VRLIST, 'SHARED ')
```

```
C PERFORM SERVICE-ROUTINES AT 1002
GOTO 1002
C AND ALWAYS RETURN TO 1001
1001 CONTINUE
```

```
C CALL THE VPUT SERVICE TO
C COPY THE VALUES LISTED IN VRLIST
C FROM THIS PROGRAMS SET OF FUNCTION VARIABLES
C TO THE SHARED VARIABLE POOL
```

```
LASTCC = ISPLNK ('VPUT ', VRLIST, 'SHARED ')
LASTCC = ISPLNK ('VPUT ', 'ZERRMSG ', 'SHARED ')
```

```
C CALL THE VDELETE SERVICE TO
C REMOVE ISPF ADDRESSABILITY TO ZERMSG
C AS PREVIOUSLY VDEFINED WITHIN THIS PROGRAM
```

```
LASTCC = ISPLNK ('VDELETE ', 'ZERRMSG ')
STOP
```

C SERVICE-ROUTINES. (ALWAYS RETURN TO 1001)

1002 CONTINUE

IF (SERVN .EQ. LIST) GO TO 1010
IF (SERVN .EQ. ADD) GO TO 1020
IF (SERVN .EQ. CHANGE) GO TO 1030
IF (SERVN .EQ. DELETE) GO TO 1040
IF (SERVN .EQ. GETLST) GO TO 1050
IF (SERVN .EQ. MODIFY) GO TO 1060
IF (SERVN .EQ. BLANK) GO TO 1070
IF (SERVN .EQ. SCAN) GO TO 1080
IF (SERVN .EQ. RESCAN) GO TO 1090
IF (SERVN .EQ. SEARCH) GO TO 1100
IF (SERVN .EQ. QUERY) GO TO 1110
IF (SERVN .EQ. PREPRP) GO TO 1120
IF (SERVN .EQ. BROWSE) GO TO 1130
IF (SERVN .EQ. EDIT) GO TO 1140
IF (SERVN .EQ. DESTRY) GO TO 1150
IF (SERVN .EQ. REFRSH) GO TO 1160
IF (SERVN .EQ. REMOVE) GO TO 1170
IF (SERVN .EQ. SAVE) GO TO 1180

1003 GO TO 1001

C LIST-TABLE.

1010 CONTINUE

C CALL THE TBTOP SERVICE TO
C SET THE CURRENT ROW POINTER (CRP)
C AHEAD OF THE FIRST ROW OF THE TABLE TELBOOK

LASTCC = ISPLNK ('TBTOP ', 'TELBOOK ')

C CALL THE TBDISPL SERVICE TO
C DISPLAY THE TABLE TELBOOK
C USING THE PANEL TPANEL2

LASTCC = ISPLNK ('TBDISPL ', 'TELBOOK ', 'TPANEL2 ')
GO TO 1003

C ADD-LINE.

1020 CONTINUE

C CALL THE TBADD SERVICE TO TRY TO
C ADD A NEW ROW OF VARIABLES TO THE TABLE TELBOOK

LASTCC = ISPLNK ('TBADD ', 'TELBOOK ')
ZERMSG(2) = M013
IF (LASTCC .NE. 0) ZERMSG(2) = M010

1022 GO TO 1003

C CHANGE-LINE.

1030 CONTINUE

C CALL THE TBPOT SERVICE TO TRY TO
C UPDATE A ROW OF VARIABLES IN THE TABLE TELBOOK

```
        LASTCC = ISPLNK ('TBPOT ', 'TELBOOK ')
        ZERMSG(2) = M015
        IF (LASTCC .NE. 0) ZERMSG(2) = M018
1032    GO TO 1003
```

C DELETE-LINE.

1040 CONTINUE

C CALL THE TBDELETE SERVICE TO TRY TO
C DELETE A ROW OF VARIABLES IN THE TABLE TELBOOK

```
        LASTCC = ISPLNK ('TBDELETE', 'TELBOOK ')

        IF (LASTCC .EQ. 8) GO TO 1042
        ZERMSG(2) = M012
        LASTCC = ISPLNK ('TBSKIP ', 'TELBOOK ', 0)
        IF (LASTCC .EQ. 8) LASTCC = ISPLNK ('TBSKIP ',
&          'TELBOOK ', 1)
        GO TO 1044
1042    ZERMSG(2) = M011
1044    GO TO 1003
```

C GET-LAST-LINE.

1050 CONTINUE

C CALL THE TBBOTTOM SERVICE TO
C SET THE CURRENT ROW POINTER (CRP)
C TO THE LAST ROW OF THE TABLE TELBOOK
C AND TO RETRIEVE THAT ROW

```
        LASTCC = ISPLNK ('TBBOTTOM', 'TELBOOK ')
        ZERMSG(2) = M016
        IF (LASTCC .NE. 0) ZERMSG(2) = M017
1052    GO TO 1003
```

C MODIFY-LINE.

1060 CONTINUE

C CALL THE TBMOD SERVICE TO
C UPDATE OR ADD A ROW OF VARIABLES TO THE TABLE TELBOOK

```
        LASTCC = ISPLNK ('TBMOD ', 'TELBOOK ')
        ZERMSG(2) = M014
        IF (LASTCC .NE. 0) ZERMSG(2) = M013
```

1062 GO TO 1003

C BLANK-SCREEN.

C SET THE CURRENT VALUES OF THE TABLE VARIABLES TO NULL.
C WHEN FANEL1 IS REDISPLAYED, THE TABLE VARIABLES WILL BE
C INITIALIZED TO NULL ON THE SCREEN.

1070 CONTINUE

LASTCC = ISPLNK ('TBVCLEAR', 'TELBOOK')

GO TO 1003

C SCAN-TABLE.

1080 CONTINUE

LASTCC = ISPLNK ('TBTOP ', 'TELBOOK')

C CALL THE TBSARG SERVICE TO ESTABLISH
C A SEARCH ARGUMENT FOR SCANNING THE TABLE TELBOOK

LASTCC = ISPLNK ('TBSARG ', 'TELBOOK')

C CALL THE TBSCAN SERVICE TO
C SEARCH THE TABLE TELBOOK FOR A ROW WITH VALUES THAT
C MATCH THE ARGUMENT LIST

LASTCC = ISPLNK ('TBSCAN ', 'TELBOOK')
GO TO 1003

C RESCAN-TABLE.

1090 CONTINUE

LASTCC = ISPLNK ('TBSCAN ', 'TELBOOK')

C CALL THE TBVCLEAR SERVICE TO CLEAR FROM THE SCREEN
C THE VARIABLES DISPLAYED FROM THE TABLE TELBOOK IF ANOTHER
C TABLE ROW IS NOT FOUND WITH THE SAME SEARCH ARGUMENT.

IF (LASTCC .EQ. 8) LASTCC = ISPLNK ('TBVCLEAR', 'TELBOOK')
GO TO 1003

C SEARCH-TABLE.

1100 CONTINUE

C CALL THE TBEXIST SERVICE TO TEST FOR THE EXISTENCE OF
C A ROW IN THE TABLE WITH THE KEY VARIABLE (LAST NAME)
C SPECIFIED ON PANEL FANEL1.

LASTCC = ISPLNK ('TBEXIST ', 'TELBOOK')

C IF THE ROW WAS FOUND,
C CALL THE TBGET SERVICE TO FETCH THE ROW FROM THE TABLE.
C OTHERWISE, THE TABLE VARIABLES WILL REMAIN UNCHANGED.

IF (LASTCC .EQ. 0) LASTCC = ISPLNK ('TBGET ', 'TELBOOK ')
GO TO 1003

C QUERY-TABLE.

1110 CONTINUE

C CALL THE TBQUERY SERVICE TO
C OBTAIN FROM THE TABLE TELBOOK
C THE KEY VARIABLE NAME (S)
C THE NON-KEY VARIABLE NAMES
C THE NUMBER OF ROWS

LASTCC = ISPLNK('TBQUERY ', 'TELBOOK ',
& 'QKEYS ', 'QNames ', 'QROWS ')

C CALL THE VCOPY SERVICE TO
C OBTAIN A COPY OF THE DIALOG VARIABLE QROWS

LASTCC = ISPLNK ('VCOPY ', 'QROWS ', 8, QDATA, 'MOVE ')

C CALL THE VREPLACE SERVICE TO
C UPDATE THE DIALOG VARIABLE QROWS

LASTCC = ISPLNK ('VREPLACE', 'QROWS ', 8, QDATA)

C CALL THE DISPLAY SERVICE TO
C DISPLAY THE PANEL TPANEL3

LASTCC = ISPLNK ('DISPLAY ', 'TPANEL3 ')
GO TO 1003

C PREPARE-REPORT.

1120 CONTINUE

C CALL THE FTOPEN SERVICE TO
C BEGIN FILE TAILORING

LASTCC = ISPLNK ('FTOPEN ')

C CALL THE FTINCL SERVICE TO
C DO FILE TAILORING USING THE FILE TAILORING SKELETON TELBOOK

LASTCC = ISPLNK ('FTINCL ', 'TELBOOK ')

C CALL THE FTCLOSE SERVICE TO
C TERMINATE FILE TAILORING AND
C STORE THE OUTPUT AS MEMBER TELOUT

```
LASTCC = ISPLNK ('FTCLOSE ', 'TELOUT ')
ZERMSG(2) = M022
GO TO 1003
```

C BROWSE-REPORT.

1130 CONTINUE

```
C CALL THE BROWSE SERVICE TO BROWSE TELOUT IN THE FILE
C TAILORING OUTPUT DATASET. THE USER SHOULD SPECIFY THE
C DATASET NAME WITH THE MEMBER. FOR VM THE FILEID OF THE FILE
C TAILORING OUTPUT MACLIB IS SPECIFIED WITH THE MEMBER NAME
C AS A SEPARATE PARAMETER. EXAMPLE:
C LASTCC = ISPLNK ('BROWSE', '(FTOUTPUT MACLIB )', 'TELOUT ')
C THIS FUNCTION IS NOT AVAILABLE UNLESS THE ISPF/PROGRAM
C DEVELOPMENT FACILITY PROGRAM PRODUCT IS INSTALLED.
```

```
LASTCC = ISPLNK ('BROWSE', '' ISPFPROJ.FTOUTPUT(TELOUT)'' ')
GO TO 1003
```

C EDIT-REPORT.

1140 CONTINUE

```
C CALL THE EDIT SERVICE TO EDIT TELOUT IN THE FILE
C TAILORING OUTPUT DATASET. THE USER SHOULD SPECIFY THE
C DATASET NAME WITH THE MEMBER. FOR VM THE FILEID OF THE FILE
C TAILORING OUTPUT MACLIB IS SPECIFIED WITH THE MEMBER NAME
C AS A SEPARATE PARAMETER. EXAMPLE:
C LASTCC = ISPLNK ('EDIT', '(FTOUTPUT MACLIB )', 'TELOUT ')
C THIS FUNCTION IS NOT AVAILABLE UNLESS THE ISPF/PROGRAM
C DEVELOPMENT FACILITY PROGRAM PRODUCT IS INSTALLED.
```

```
LASTCC = ISPLNK ('EDIT', '' ISPFPROJ.FTOUTPUT(TELOUT)'' ')
GO TO 1003
```

C DESTROY-REPORT.

1150 CONTINUE

```
C CALL THE FTERASE SERVICE TO
C DELETE THE FILE TAILORING OUTPUT TELOUT
```

```
LASTCC = ISPLNK ('FTERASE ', 'TELOUT ')
ZERMSG(2) = M023
GO TO 1003
```

C REFRESH-TABLE.

1160 CONTINUE

```
C CALL THE TBEND SERVICE TO
C CLOSE THE TABLE TELBOOK WITHOUT SAVING IT ON DISK
```

```
LASTCC = ISPLNK ('TBEND ', 'TELBOOK ')
LASTCC = ISPLNK ('TBOPEN ', 'TELBOOK ')
LASTCC = ISPLNK ('TBSKIP ', 'TELBOOK ')
ZERMSG(2) = M021
GO TO 1003
```

```
C REMOVE-TABLE.
```

```
1170 CONTINUE
```

```
LASTCC = ISPLNK ('TBCLOSE ', 'TELBOOK ')
```

```
C CALL THE TBERASE SERVICE TO
C DELETE THE TABLE TELBOOK FROM THE TABLE OUTPUT LIBRARY
```

```
LASTCC = ISPLNK ('TBERASE ', 'TELBOOK ')
LASTCC = ISPLNK ('TBCREATE', 'TELBOOK ', TABKEY, TABVRS)
ZERMSG(2) = M020
GO TO 1003
```

```
C SAVE-TABLE.
```

```
1180 CONTINUE
```

```
C CALL THE TBSAVE SERVICE TO
C WRITE THE TABLE TELBOOK FROM VIRTUAL STORAGE
C TO THE TABLE OUTPUT LIBRARY
```

```
LASTCC = ISPLNK ('TBSAVE ', 'TELBOOK ')
ZERMSG(2) = M019
GO TO 1003
END
```

PL/I IMPLEMENTATION

In addition to the components listed under "Dialog Components Applicable to All Implementations," the following components are required for the PL/I implementation of the dialog:

- Combined selection and data entry panel TPANEL1
- Function TELEBOOK
- Function TELESRVC

PL/I - COMBINED SELECTION AND DATA ENTRY PANEL TPANEL1

Definition and display of panel TPANEL1 is shown below.

Panel Definition

```
)ATTR
  _ TYPE(INPUT) INTENS(LOW) JUST(LEFT) PAD(' ') CAPS(ON)
  @ TYPE(INPUT) INTENS(HIGH) JUST(LEFT) PAD(' ') CAPS(ON)

)BODY
%----- Telephone Book Application -----
+SELECT AN OPTION:%==>@ZCMD
%
% A+- Add new entry
% C+- Change existing entry %F+- Refresh the table, from disk
% D+- Delete old entry %K+- Remove the table, from disk
% G+- Get last entry %L+- List the table
% M+- Add/Change entry %Q+- Query the table
% Z+- Blank screen entry %S+- Scan the table, non unique
% P+- Prepare report %R+- Rescan the table, continue scan
% B+- Browse report %T+- Search on last name, unique key
% E+- Edit report %V+- Save the table, on disk
% Y+- Destroy report
%
+Current values / enter search values:
%
+ FIRST NAME ==> _TFNAM +
+ LAST NAME ==> _TLNAM +
+ LOCATION ==> _TLOC +
+ PHONE NUM ==> _TNUM +
+ PHONE COLOR==> _TCOLOR +
+ PHONE TYPE ==> _TTYPE +

)INIT

.CURSOR = ZCMD
&ZCMD = &OTELOPT
&VARL = '(TNUM TLNAM TFNAM TLOC TTYPE TCOLOR)'
IF(&ZERRMSG ^= ' ')
.MSG = &ZERRMSG
&ZERRMSG = ' '

)PROC

&ZSEL = TRANS( &ZCMD
L, 'PGM(TELESRVC) PARM(/LIST &VARL)'
A, 'PGM(TELESRVC) PARM(/ADD &VARL)'
C, 'PGM(TELESRVC) PARM(/CHANGE &VARL)'
D, 'PGM(TELESRVC) PARM(/DELETE &VARL)'
G, 'PGM(TELESRVC) PARM(/GETLAST &VARL)'
M, 'PGM(TELESRVC) PARM(/MODIFY &VARL)'
```



```

Z, 'PGM(TELESRVC) PARM(/BLANK &VARL)'
S, 'PGM(TELESRVC) PARM(/SCAN &VARL)'
R, 'PGM(TELESRVC) PARM(/RESCAN &VARL)'
T, 'PGM(TELESRVC) PARM(/SEARCH &VARL)'
Q, 'PGM(TELESRVC) PARM(/QUERY &VARL)'
P, 'PGM(TELESRVC) PARM(/PREPREP &VARL)'
B, 'PGM(TELESRVC) PARM(/BROWSE &VARL)'
E, 'PGM(TELESRVC) PARM(/EDIT &VARL)'
Y, 'PGM(TELESRVC) PARM(/DESTROY &VARL)'
F, 'PGM(TELESRVC) PARM(/REFRESH &VARL)'
K, 'PGM(TELESRVC) PARM(/REMOVE &VARL)'
V, 'PGM(TELESRVC) PARM(/SAVE &VARL)'
, ' '
*, '?' )

```

```

&OSEL = &ZSEL
IF(&ZCMD = S)
  &ZCMD = R
&OTEOPT = &ZCMD

```

```

IF(&ZCMD = R)
  VER(&TNUM, PICT, 'NNN-NNNN')

```

)END

Panel Display

```

----- TELEPHONE BOOK APPLICATION -----
SELECT AN OPTION: ==> C_

A - Add new entry
C - Change existing entry
D - Delete old entry
G - Get last entry
M - Add/Change entry
Z - Blank screen entry
P - Prepare report
B - Browse report
E - Edit report
Y - Destroy report

F - Refresh the table, from disk
K - Remove the table, from disk
L - List the table
Q - Query the table
S - Scan the table, non unique
R - Rescan the table, continue scan
T - Search on last name, unique key
V - Save the table, on disk

Current values / enter search values:

FIRST NAME ==> LINDA
LAST NAME ==> MILLER
LOCATION ==> KITCHEN
PHONE NUM ==> 462-4441
PHONE COLOR==> RED
PHONE TYPE ==> WALL DIAL

```

PL/I - FUNCTION TELEBOOK

TELEBK: PROC OPTIONS(MAIN);

```
/*
* THE OBJECTIVE OF THE CODING TECHNIQUE OF THIS APPLICATION IS
* TO ILLUSTRATE ISPF SERVICES
*
* THE APPLICATION CONSISTS OF
* THIS MAIN PROGRAM - TELEBOOK
* 1 SUB PROGRAM - TELESRVC
* 3 PANELS - TPANEL1 TPANEL2 TPANEL3
* 15 MESSAGES - TELB010 THRU TELB024
* 1 FILE TAILORING SKELETON - TELBOOK
*
* THE APPLICATION PROVIDES THE FOLLOWING FUNCTIONS
* FROM PANEL TPANEL1 TO ACCESS A TELEPHONE BOOK
*
* A ADD NEW ENTRY
* C CHANGE EXISTING ENTRY F REFRESH THE TABLE, FROM DISK
* D DELETE OLD ENTRY K REMOVE THE TABLE, FROM DISK
* G GET LAST ENTRY L LIST THE TABLE
* M ADD/CHANGE ENTRY Q QUERY THE TABLE
* Z BLANK SCREEN ENTRY S SCAN THE TABLE, NON UNIQUE
* P PREPARE REPORT R RESCAN THE TABLE, CONTINUE SCAN
* B BROWSE REPORT T SEARCH ON LAST NAME, UNIQUE KEY
* E EDIT REPORT V SAVE THE TABLE; ON DISK
* Y DESTROY REPORT
*/
```

```
DCL ISPLINK EXTERNAL ENTRY OPTIONS(ASM RETCODE INTER);
DCL PLIRETV BUILTIN;
DCL TERMINATION_MESSAGE CHAR(8) INIT('TELBO24 ');
DCL L15 FIXED BIN(31,0) INIT(15);
DCL COMMAND_STREAM CHAR(15) INIT('PANEL(TPANEL1)');
```

```
/*
* CALL THE CONTROL SERVICE TO
* SET THE ERROR PROCESSING MODE FOR THIS PROGRAM TO
* TERMINATE ONLY FOR ANY ERROR THAT HAS A RETURN CODE OF
* 12 OR HIGHER
*/
```

```
CALL ISPLINK('CONTROL ', 'ERRORS ', 'CANCEL ');
```

```
/*
* CALL THE TBOPEN SERVICE TO
* OPEN THE TABLE, TELBOOK IN WRITE MODE
*/
```

```
CALL ISPLINK('TBOPEN', 'TELBOOK ', 'WRITE');
IF PLIRETV() = 8 THEN
```

```

/*****
*   CALL THE TBCREATE SERVICE TO                               *
*   CREATE THE TELBOOK IF IT DOES NOT EXIST.  THE KEY        *
*   VARIABLE IS SPECIFIED BY TABKEY AND THE NON-KEY          *
*   VARIABLES ARE SPECIFIED IN TABVARS.                       *
*****/

```

```

      CALL ISPLINK('TBCREATE ', 'TELBOOK ', '(TLNAM) ',
                  '(TFNAM TLOC TNUM TCOLOR TTYPE)');

```

```

/*****
*   CALL THE TBSKIP SERVICE TO                               *
*   MOVE THE CURRENT ROW POINTER (CRP) OF THE TABLE FORWARD *
*   ONE POSITION (DEFAULT).  THE VARIABLES WILL BE SET TO     *
*   VALUES CONTAINED IN THE FIRST ROW OF THE TABLE OR REMAIN *
*   SET TO NULLS IF THE TABLE WAS JUST CREATED.            *
*****/

```

```

      CALL ISPLINK('TBSKIP ', 'TELBOOK ');

```

```

/*****
*   SET THE TABLE VARIABLES IN THE SHARED VARIABLE POOL TO THE *
*   VALUES CONTAINED IN THE 1ST ROW OF THE TABLE.  THESE VALUES *
*   WILL BE USED TO INITIALIZE THE VALUES ON THE SCREEN FOR   *
*   PANEL TPANEL1.                                             *
*****/

```

```

      CALL ISPLINK('VPUT ', '(TFNAM TLNAM TLOC TNUM TCOLOR TTYPE)',

```

```

/*****
*   CALL THE SELECT SERVICE TO                               *
*   CONTINUE TO DISPLAY PANEL TPANEL1 AND CONTROL THE        *
*   INVOCATION OF THE SUB PROGRAM TELESRVC                   *
*****/

```

```

      CALL ISPLINK('SELECT ', L15, COMMAND_STREAM);

```

```

/*****
*   CALL THE TBCLOSE SERVICE TO CLOSE THE TABLE TELBOOK    *
*****/

```

```

      CALL ISPLINK('TBCLOSE ', 'TELBOOK ');

```

```

/*****
*   CALL THE LOG SERVICE TO                                  *
*   PUT THE MESSAGE TELB024 ON THE ISPF LOG FILE             *
*****/

```

```

      CALL ISPLINK('LOG ', TERMINATION_MESSAGE);

```

```

/*****
*   CALL THE SETMSG SERVICE TO                               *
*   PUT THE MESSAGE TELB024 ON THE NEXT SCREEN DISPLAYED    *
*****/

```

```

    CALL ISPLINK('SETMSG ', TERMINATION_MESSAGE);

```

```

/*****
*   CALL THE VRESET SERVICE TO                               *
*   REMOVE THE ISPF ADDRESSABILITY OF ANY VDEFINED VARIABLES *
*****/

```

```

    CALL ISPLINK('VRESET ');
END TELEBK;

```

PL/I - FUNCTION TELESRVC

```

TELESRV: PROCEDURE (PRM) OPTIONS(MAIN);

```

```

/*****
*
*   THIS SUBPROGRAM IS INVOKED FROM PANEL TPANEL1 DISPLAYED *
*   BY THE MAIN PROGRAM TELEBOOK USING THE SELECT SERVICE *
*
*****/

```

```

DCL PRM          CHAR(45) VARYING;
DCL ISPLINK EXTERNAL ENTRY OPTIONS(ASM RETCODE INTER);
DCL PLIRETV BUILTIN;
DCL SUBSTR BUILTIN;
DCL PANEL2      CHAR(8) INIT('TPANEL2 ');
DCL PANEL3      CHAR(8) INIT('TPANEL3 ');
DCL TELBOOK     CHAR(8) INIT('TELBOOK ');

```

```

/*****
*   FTOUTPUT SHOULD BE INITIALIZED TO THE USER'S FILE TAILORING *
*   OUTPUT PARTITIONED DATASET NAME WITH THE MEMBER DEFINED     *
*   AS TELOUT.  FOR VM FTOUTPUT SHOULD CONTAIN THE FILE        *
*   TAILORING OUTPUT MACLIB NAME WITHOUT A MEMBER NAME, AS:    *
*
*   DCL FTOUTPUT CHAR(20) INIT('(FTOUTPUT MACLIB A)');
*****/

```

```

DCL FTOUTPUT CHAR(30) INIT(''ISPFPROJ.FTOUTPUT(TELOUT)''');
DCL TABKEY   CHAR(8)  INIT('(TLNAM) ');
DCL TABVRS   CHAR(30) INIT('(TFNAM TLOC TNUM TCOLOR TTYPE)');
DCL ZERMSG   CHAR(8)  INIT('(8) ');
DCL ZMSGNAME CHAR(9)  INIT('(ZERRMSG)');
DCL QKEYS    CHAR(8)  INIT('QKEYS ');
DCL QNAMES   CHAR(8)  INIT('QNAMES ');

```

```

DCL QROWS      CHAR(8) INIT('QROWS ');
DCL QROWSDATA  CHAR(8);
DCL QROWSDATAZ CHAR(8);
DCL ZEERO      FIXED BIN(31,0) INIT(0);
DCL ONE        FIXED BIN(31,0) INIT(1);
DCL L8         FIXED BIN(31,0) INIT(8);
DCL VARLIST    CHAR(36);
DCL SERVN      CHAR(7);

```

```

/*****
*   VARLIST IS SET TO THE LIST OF THE NAMES OF THE TABLE      *
*   VARIABLES AND SERVN IS SET TO THE FUNCTION REQUESTED BY    *
*   THE USER ON TPANEL1.  THESE VALUES WERE PASSED AS A      *
*   PARAMETER FROM TPANEL1 WHEN THE PROGRAM WAS INVOKED.      *
*****/

```

```

VARLIST = SUBSTR(PRM,9,36);
SERVN   = SUBSTR(PRM,1,7);

```

```

/*****
*   CALL THE VDEFINE SERVICE TO                                *
*   GIVE ISPF ADDRESSABILITY TO ZERMSG AS DEFINED WITHIN      *
*   THIS PROGRAM AS A CHARACTER FIELD WITH A LENGTH OF 8     *
*****/

```

```

CALL ISPLINK('VDEFINE ', ZMSGNAME, ZERMSG, 'CHAR ', L8);

```

```

/*****
*   CALL THE VGET SERVICE TO                                  *
*   COPY THE VALUES LISTED IN VARLIST FROM THE SHARED VARIABLE *
*   POOL TO THIS PROGRAM'S SET OF FUNCTION VARIABLES         *
*****/

```

```

CALL ISPLINK('VGET ', VARLIST, 'SHARED ');
CALL SERVRTN;

```

```

/*****
*   CALL THE VPUT SERVICE TO                                  *
*   COPY THE VALUES LISTED IN VARLIST FROM THIS PROGRAM'S SET *
*   OF FUNCTION VARIABLES TO THE SHARED VARIABLE POOL        *
*****/

```

```

CALL ISPLINK('VPUT ', VARLIST, 'SHARED ');
CALL ISPLINK('VPUT ', ZMSGNAME, 'SHARED ');

```

```

/*****
*   CALL THE VDELETE SERVICE TO REMOVE ISPF ADDRESSABILITY    *
*   TO ZERMSG AS PREVIOUSLY VDEFINED WITHIN THIS PROGRAM     *
*****/

```

```

CALL ISPLINK('VDELETE ', ZMSGNAME);

```

```

SERVRTN: PROCEDURE;
  IF      SERVN = 'LIST  ' THEN CALL LISTTAB;
  ELSE IF SERVN = 'ADD   ' THEN CALL ADDLINE;
  ELSE IF SERVN = 'CHANGE' THEN CALL CHNGLIN;
  ELSE IF SERVN = 'DELETE' THEN CALL DELETED;
  ELSE IF SERVN = 'GETLAST' THEN CALL GETLSTL;
  ELSE IF SERVN = 'MODIFY ' THEN CALL MODIFYL;
  ELSE IF SERVN = 'BLANK ' THEN CALL BLNKSCR;
  ELSE IF SERVN = 'SCAN  ' THEN CALL SCANTAB;
  ELSE IF SERVN = 'RESCAN' THEN CALL RESCNTB;
  ELSE IF SERVN = 'SEARCH' THEN CALL SRCHTAB;
  ELSE IF SERVN = 'QUERY  ' THEN CALL QRYTABL;
  ELSE IF SERVN = 'PREPREP' THEN CALL PREPRRP;
  ELSE IF SERVN = 'BROWSE ' THEN CALL BRSEREP;
  ELSE IF SERVN = 'EDIT  ' THEN CALL EDITREP;
  ELSE IF SERVN = 'DESTROY' THEN CALL DSTRYRP;
  ELSE IF SERVN = 'REFRESH' THEN CALL RFRSHTB;
  ELSE IF SERVN = 'REMOVE ' THEN CALL RMVTABL;
  ELSE IF SERVN = 'SAVE  ' THEN CALL SAVETAB;
END SERVRTN;

```

```

LISTTAB: PROCEDURE;

```

```

/*****
*   CALL THE TBTOP SERVICE TO                               *
*   SET THE CURRENT ROW POINTER (CRP)                       *
*   AHEAD OF THE FIRST ROW OF THE TABLE TELBOOK          *
*****/

```

```

      CALL ISPLINK('TBTOP ', TELBOOK);

```

```

/*****
*   CALL THE TBDISPL SERVICE TO                             *
*   DISPLAY THE TABLE TELBOOK USING THE PANEL TPANEL2    *
*****/

```

```

      CALL ISPLINK('TBDISPL ', TELBOOK, PANEL2);
END LISTTAB;

```

```

ADDLINE : PROCEDURE;

```

```

/*****
*   CALL THE TBADD SERVICE TO TRY TO                       *
*   ADD A NEW ROW OF VARIABLES TO THE TABLE TELBOOK      *
*****/

```

```

      CALL ISPLINK('TBADD ', TELBOOK);
      IF PLIRETV() = 0 THEN
        ZERMSG = 'TELBO13 ';
      ELSE
        ZERMSG = 'TELBO10 ';
      END ADDLINE;

```

CHNGLIN: PROCEDURE;

```
/*  
* CALL THE TBPOT SERVICE TO TRY TO *  
* UPDATE A ROW OF VARIABLES IN THE TABLE TELBOOK *  
*/
```

```
CALL ISPLINK('TBPOT ', TELBOOK);  
IF PLIRETV() = 0 THEN  
    ZERMSG = 'TEL015 ';  
ELSE  
    ZERMSG = 'TEL018 ';  
END CHNGLIN;
```

DELETEL: PROCEDURE;

```
/*  
* CALL THE TBDELETE SERVICE TO TRY TO *  
* DELETE A ROW OF VARIABLES IN THE TABLE TELBOOK *  
*/
```

```
CALL ISPLINK('TBDELETE ', TELBOOK);  
IF PLIRETV() = 8 THEN  
    ZERMSG = 'TEL011 ';  
ELSE  
    ZERMSG = 'TEL012 ';  
    CALL ISPLINK('TBSKIP ', TELBOOK, ZEERO);  
    IF PLIRETV() = 8 THEN  
        CALL ISPLINK('TBSKIP ', TELBOOK, ONE);  
END DELETEL;
```

GETLSTL: PROCEDURE;

```
/*  
* CALL THE TBBOTTOM SERVICE TO *  
* SET THE CURRENT ROW POINTER (CRP) TO THE LAST ROW OF THE *  
* LAST ROW OF THE TABLE TELBOOK AND TO RETRIEVE THAT ROW *  
*/
```

```
CALL ISPLINK('TBBOTTOM', TELBOOK);  
IF PLIRETV() = 0 THEN  
    ZERMSG = 'TEL016 ';  
ELSE  
    ZERMSG = 'TEL017 ';  
END GETLSTL;
```

MODIFYL: PROCEDURE;

```
/*  
* CALL THE TBMOD SERVICE TO *  
* UPDATE OR ADD A ROW OF VARIABLES TO THE TABLE TELBOOK *  
*/
```

```
CALL ISPLINK('TBMOD ', TELBOOK);  
IF PLIRETV() = 0 THEN  
    ZERMSG = 'TELBO14 ' ;  
ELSE  
    ZERMSG = 'TELBO13 ' ;  
END MODIFYL;
```

BLNKSCR: PROCEDURE;

```
/*  
* SET THE CURRENT VALUES OF THE TABLE VARIABLES TO NULL. *  
* WHEN TPANEL1 IS REDISPLAYED, THESE VARIABLES WILL BE *  
* DISPLAYED AS NULL VALUES ON THE SCREEN. *  
*/
```

```
CALL ISPLINK('TBVCLEAR', TELBOOK);  
END BLNKSCR;
```

SCANTAB: PROCEDURE;

```
CALL ISPLINK('TBTOP ', TELBOOK);
```

```
/*  
* CALL THE TBSARG SERVICE TO ESTABLISH *  
* A SEARCH ARGUMENT FOR SCANNING THE TABLE TELBOOK *  
*/
```

```
CALL ISPLINK('TBSARG ', TELBOOK);
```

```
/*  
* CALL THE TBSCAN SERVICE TO *  
* SEARCH THE TABLE TELBOOK FOR A ROW WITH VALUES THAT *  
* MATCH THE ARGUMENT LIST *  
*/
```

```
CALL ISPLINK('TBSCAN ', TELBOOK);  
END SCANTAB;
```

RESCNTB: PROCEDURE;

```
CALL ISPLINK('TBSCAN ', TELBOOK);  
IF PLIRETV() = 8 THEN
```



```

/*****
*   CALL THE TBVCLEAR SERVICE TO                               *
*   CLEAR FROM THE SCREEN THE VARIABLES DISPLAYED FROM THE   *
*   TABLE TELBOOK IF ANOTHER ROW WAS NOT FOUND WITH THE    *
*   SAME SEARCH ARGUMENTS.                                   *
*****/

        CALL ISPLINK('TBVCLEAR', TELBOOK);
END RESCNTB;

SRCHTAB: PROCEDURE;

/*****
*   CALL THE TBEXIST SERVICE TO                               *
*   TEST FOR THE EXISTENCE OF A ROW WITH KEY VARIABLE VALUE  *
*   (LAST NAME) SPECIFIED ON TPANEL1.                       *
*****/

        CALL ISPLINK('TBEXIST ', TELBOOK);

        IF PLIRETV() = 0 THEN

/*****
*   IF THE ROW WAS FOUND CALL THE TBGET SERVICE TO          *
*   FETCH THE ROW OF VARIABLES FROM THE TABLE. OTHERWISE  *
*   THE VALUE OF THE TABLE VARIABLES WILL REMAIN UNCHANGED. *
*****/

        CALL ISPLINK('TBGET ', TELBOOK);
END SRCHTAB;

QRYTABL: PROCEDURE;

/*****
*   CALL THE TBQUERY SERVICE TO OBTAIN FROM THE TABLE TELBOOK: *
*   KEY VARIABLE NAME (S)                                       *
*   NON-KEY VARIABLE NAMES                                       *
*   NUMBER OF ROWS                                               *
*****/

        CALL ISPLINK('TBQUERY ', TELBOOK, QKEYS, QNAMES, QROWS);

/*****
*   CALL THE VCOPY SERVICE TO                                 *
*   OBTAIN A COPY OF THE DIALOG VARIABLE QROWS                *
*****/

        CALL ISPLINK('VCOPY ', 'QROWS ', L8, QROWSDATA, 'MOVE ');
        QROWSDATAZ = QROWSDATA;

```

```

/*****
*   CALL THE VREPLACE SERVICE TO                               *
*   UPDATE THE DIALOG VARIABLE QROWS                           *
*****/

      CALL ISPLINK('VREPLACE', 'QROWS ', L8, QROWSDATAZ);

/*****
*   CALL THE DISPLAY SERVICE TO DISPLAY THE PANEL TPANEL3     *
*****/

      CALL ISPLINK('DISPLAY ', PANEL3);
      END QRYTABL;

      PREPRRP: PROCEDURE;

/*****
*   CALL THE FTOPEN SERVICE TO BEGIN FILE TAILORING           *
*****/

      CALL ISPLINK('FTOPEN ');

/*****
*   CALL THE FTINCL SERVICE TO                                 *
*   DO FILE TAILORING USING THE FILE TAILORING SKELETON TELBOOK *
*****/

      CALL ISPLINK('FTINCL ', TELBOOK);

/*****
*   CALL THE FTCLOSE SERVICE TO TERMINATE FILE TAILORING      *
*   AND STORE THE OUTPUT AS MEMBER TELOUT                      *
*****/

      CALL ISPLINK('FTCLOSE ', 'TELOUT ');
      ZERMSG = 'TELB022 ';
      END PREPRRP;

      BRSEREP: PROCEDURE;

/*****
*   CALL THE BROWSE SERVICE TO                                 *
*   BROWSE THE FILE TAILORING OUTPUT TELOUT.                  *
*   *                                                         *
*   FOR VM, FTOOUTPUT MUST SPECIFY THE FILEID OF THE FILE     *
*   TAILORING OUTPUT MACLIB WITH THE MEMBER NAME SPECIFIED    *
*   AS A SEPARATE PARAMETER.  EXAMPLE:                        *
*   'CALL ISPLINK('BROWSE', FTOOUTPUT, 'TELOUT');'.           *
*   THIS FUNCTION IS NOT AVAILABLE UNLESS THE ISPF/PROGRAM    *
*   DEVELOPMENT FACILITY PROGRAM PRODUCT IS INSTALLED.       *
*****/

```

```
CALL ISPLINK('BROWSE', FTOUTPUT);
END BRSEREP;
```

```
EDITREP: PROCEDURE;
```

```
/*
* CALL THE EDIT SERVICE TO EDIT THE FILE TAILORING OUTPUT *
* TELOUT. *
* *
* FOR VM, FTOUTPUT MUST SPECIFY THE FILEID OF THE FILE *
* TAILORING OUTPUT MACLIB WITH THE MEMBER NAME SPECIFIED *
* AS A SEPARATE PARAMETER. EXAMPLE: *
* 'CALL ISPLINK('EDIT', FTOUTPUT, 'TELOUT');'. *
* THIS FUNCTION IS NOT AVAILABLE UNLESS THE ISPF/PROGRAM *
* DEVELOPMENT FACILITY PROGRAM PRODUCT IS INSTALLED. *
*/
```

```
CALL ISPLINK('EDIT', FTOUTPUT);
END EDITREP;
```

```
DSTRYRP: PROCEDURE;
```

```
/*
* CALL THE FTERASE SERVICE TO *
* DELETE THE FILE TAILORING OUTPUT TELOUT *
*/
```

```
CALL ISPLINK('FTERASE ', 'TELOUT ');
ZERMSG = 'TELBO23 ';
END DSTRYRP;
```

```
RFRSHTB: PROCEDURE;
```

```
/*
* CALL THE TBEND SERVICE TO *
* CLOSE THE TABLE TELBOOK WITHOUT SAVING IT ON DISK *
*/
```

```
CALL ISPLINK('TBEND ', TELBOOK);
CALL ISPLINK('TBOPEN ', TELBOOK);
CALL ISPLINK('TBSKIP ', TELBOOK);
ZERMSG = 'TELBO21 ';
END RFRSHTB;
```

```
RMVTABL: PROCEDURE;
```

```
    CALL ISPLINK('TBCLOSE ', TELBOOK);
```

```
/******  
*   CALL THE TBERASE SERVICE TO                               *  
*   DELETE THE TABLE TELBOOK FROM THE TABLE OUTPUT LIBRARY *  
*****/
```

```
    CALL ISPLINK('TBERASE ', TELBOOK);  
    CALL ISPLINK('TBCREATE ', TELBOOK, TABKEY, TABVRS);  
    ZERMSG = 'TELBO20 ';
```

```
END RMVTABL;
```

```
SAVETAB: PROCEDURE;
```

```
/******  
*   CALL THE TBSAVE SERVICE TO                               *  
*   WRITE THE TABLE TELBOOK FROM VIRTUAL STORAGE TO THE TABLE *  
*   OUTPUT LIBRARY                                           *  
*****/
```

```
    CALL ISPLINK('TBSAVE ', TELBOOK);  
    ZERMSG = 'TELBO19 ';
```

```
END SAVETAB;
```

```
END TELESRV;
```


INDEX

B

BROWSE

- clist 46
- cobol 72
- exec2 58
- fortran 85
- pl1 98

C

clist

- BROWSE 46
- CONTROL 41
- DISPLAY
 - employee records 9, 10
 - telephone book 45
- EDIT 46
- EMPL panel 8
- FTCLOSE 46
- FTERASE 46
- FTINCL 46
- FTOPEN 45
- LOG 42
- panel CPANEL1, definition 38
- panel EMPL, display 8
- primary option menu panel 8
- SELECT 42
- SETMSG 42
- TBADD
 - employee records 10
 - telephone book 43
- TBBOTTOM 44
- TBCLOSE
 - employee records 11
 - telephone book 42, 47
- TBCREATE
 - employee records 9
 - telephone book 41, 47
- TBDELETE
 - employee records 10
 - telephone book 44
- TBDISPL 43
- TBEND 46
- TBERASE 47
- TBEXIST 45

TBGET

- employee records 9
- telephone book 45

TBMOD 44

TBOPEN

- employee records 9
- telephone book 41, 46

TBPUT

- employee records 10
- telephone book 44

TBQUERY 45

TBSARG 47

TBSAVE 47

TBSCAN 47

- TBSKIP 41, 44
- telephone book 46

TBTOP 43

TBVCLER 45, 47

TPANEL1 30

VGET 43

VPUT 42, 48

clist implementation

- employee records 8
- telephone book 38

cobol

BROWSE 72

CONTROL 63

define variables 16

DISPLAY

- employee records 18, 20
- telephone book 71

EDIT 72

EMPL panel 15

FTCLOSE 72

FTERASE 72

FTINCL 72

FTOPEN 71

LOG

- employee records 20

- telephone book 64

panel EMPL

- definition 15

- display 15

panel TPANEL1

- definition 60

- display 62

SELECT 64

SETMSG 65
 TBADD
 employee records 20
 telephone book 68
 TBBOTTOM 69
 TBCLOSE
 employee records 18
 telephone book 64, 73
 TBCREATE
 employee records 18
 telephone book 64, 73
 TBDELETE
 employee records 20
 telephone book 69
 TBDISPL 68
 TBEND 73
 TBERASE 73
 TBEXIST 70
 TBGET
 employee records 20
 telephone book 71
 TBMOD 69
 TBOPEN
 employee records 18
 telephone book 63, 73
 TBPUT
 employee records 20
 telephone book 69
 TBQUERY 71
 TBSARG 70
 TBSAVE 73
 TBSCAN 70
 TBSKIP 64, 69, 73
 TBTOP 68, 70
 TBVCLEAR 70
 TPANEL1 30
 VCOPY 71
 VDEFINE
 employee records 17
 telephone book 67
 VDELETE 67
 VGET 67
 VPUT 64, 67
 VREPLACE 71
 VRESET
 employee records 18
 telephone book 65
 cobol implementation
 employee records 15
 telephone book 60
 command, ISPSTART
 employee records 2
 telephone book 31
 command procedure iv
 common dialog components
 employee records 3
 telephone book 34
 CONTROL
 clist 41
 cobol 63
 exec2
 employee records 12
 telephone book 51
 fortran 77
 p11 89
 CPANEL1
 clist definition 38
 exec2, display 50
 exec2 definition 48
 telephone book 30, 34

 D
 data entry 4, 5
 data entry panel 4
 data set
 partitioned iv
 sequential iv
 declare statements, p11 27
 delete variables, p11 28
 description of processing
 employee records 2
 telephone book 33
 dialog components, common
 employee records 3
 telephone book 34
 data entry panels EMPLA, EMPLB 1
 employee records 1
 messages 1
 primary option menu EMPL 1
 telephone book 30
 dialog variables, cobol definition 16
 DISPLAY, clist
 employee records 9, 10
 telephone book 45
 cobol
 employee records 18, 20
 telephone book 71
 exec2
 employee records 12, 13
 telephone book 57
 fortran 22, 23
 p11
 employee records 26
 telephone book 97

E

EDIT

- clist 46
- cobol 72
- exec2 58
- fortran 85
- p11 98
- EMPL panel 1, 2, 8
 - cobol 15
 - definition 8
 - display 8
 - exec2 11
 - fortran 20
 - p11 25
- EMPLA panel 1
 - definition 4
 - display 4
- EMPLB panel 1
 - definition 5
 - display 6
- employee records 1
 - clist, EMPL panel 8
 - cobol
 - define variables 16
 - EMPL panel 15
 - common dialog components 3
 - data entry 4, 5
 - dialog components 1
 - exec2, EMPL panel 11
 - fortran, EMPL panel 20
 - invocation 2
 - ISPSTART command 2
 - messages 7
 - organization 2
 - overview 1
 - p11
 - declare statements 27
 - delete variables 28
 - EMPL panel 25
 - included segment 27
 - variable definitions 27
 - processing description 2
- TBGET, p11 26
- exec2
 - BROWSE 58
 - CONTROL
 - employee records 12
 - telephone book 51
 - CPANEL1 30
 - DISPLAY
 - employee records 12, 13
 - telephone book 57

EDIT 58

EMPL panel 11, 12

FTCLOSE 58

FTERASE 58

FTINCL 58

FTOPEN 57

LOG

employee records 14

telephone book 52

panel CPANEL1

definition 48

display 50

panel EMPL, display 12

primary option menu panel 11

SELECT 52

SETMSG 52

TBADD

employee records 13

telephone book 54

TBBOTTOM 55

TBCLOSE

employee records 14

telephone book 52, 59

TBCREATE

employee records 12

telephone book 51

TBDELETE

employee records 14

telephone book 55

TBDISPL 54

TBEND 59

TBERASE 59

TBEXIST 57

TBGET

employee records 13

telephone book 57

TBMOD 56

TBOPEN

employee records 12

telephone book 51, 59

TBPUT

employee records 14

telephone book 55

TBQUERY 57

TBSARG 56

TBSAVE 59

TBSCAN 56

TBSKIP 51, 59

TBTOP 54, 56

TBVCLEAR 56, 57

VGET 53

VPUT 52, 54

exec2 implementation

employee records 11
telephone book 48

F

file iv

file tailoring skeleton 38

fortran

BROWSE 85

CONTROL 77

DISPLAY 22, 23

EDIT 85

EMPL panel 20

FPANEL1 30

FTCLOSE 85

FTERASE 85

FTINCL 84

FOPEN 84

LOG

employee records 24

telephone book 78

panel EMPL

definition 20

display 21

panel FPANEL1

definition 74

display 76

primary option menu panel 20

SELECT 78

SETMSG 78

TBADD

employee records 23

telephone book 81

TBBOTTOM 82

TBCLOSE

employee records 24

telephone book 78, 86

TBCREATE

employee records 22

telephone book 77, 86

TBDELETE

employee records 24

telephone book 82

TBDISPL 81

TBEND 86

TBERASE 86

TBEXIST 83

TBGET

employee records 22

telephone book 84

TBMOD 82

TBOPEN

employee records 22

telephone book 77, 86

TBPUT

employee records 23

telephone book 82

TBQUERY 84

TBSARG 83

TBSAVE 86

TBSCAN 83

telephone book 83

TBSKIP 77, 82, 86

TBTOP 81, 83

TBVCLER 83

VCOPY 80, 84

VDEFINE

employee records 21

telephone book 80

VDELETE

employee records 24

telephone book 80

VGET 80

VPUT 77, 80

VREPLACE 84

VRESET 78

fortran implementation

employee records 20

telephone book 73

FPANEL1

fortran definition 74

fortran display 76

telephone book 30, 34

FTCLOSE

clist 46

cobol 72

exec2 58

fortran 85

p11 97

FTERASE

clist 46

cobol 72

exec2 58

fortran 85

p11 98

FTINCL

clist 46

cobol 72

exec2 58

fortran 84

p11 97

FOPEN

clist 45

cobol 71

exec2 57

fortran 84
p11 97

I

included segment, p11 27
information display panel, TPANEL3 36
invocation
 employee records 2
 telephone book 31
ISPSTART command
 employee records 2
 telephone book 31

L

library iv
LOG
 clist 42
 cobol
 employee records 20
 telephone book 64
 exec2
 employee records 14
 telephone book 52
 fortran
 employee records 24
 telephone book 78
 p11
 employee records 26
 telephone book 90

M

MACLIB iv
main function routine 33
messages
 employee records 7
 telephone book 37

O

organization
 employee records 2
 telephone book 32
overview
 employee records 1
 telephone book 29

P

panel, CPANEL1
 clist 30
 clist definition 38
 exec2 30
 exec2 definition 48
 exec2 display 50
data entry 4
EMPL 1, 2
 clist definition 8
 clist display 8
 cobol definition 15
 cobol display 15
 exec2 definition 11
 exec2 display 12
 fortran definition 20
 fortran display 21
 p11 definition 25
 p11 display 25
EMPLA 1
 definition 4
 display 4
EMPLB 1
 definition 5
 display 6
FPANEL1
 fortran 30
 fortran definition 74
 fortran display 76
TPANEL1
 cobol 30
 cobol definition 60
 cobol display 62
 p11 30
 p11 definition 87
 p11 display 88
TPANEL2
 definition 35
 display 35
TPANEL3
 definition 36
 display 36
panel TPANEL1 definition 87
partitioned data set iv
p11
 BROWSE 98
 CONTROL 89
 declare statements 27
 delete variables 28
DISPLAY

- employee records 26
- telephone book 97
- EDIT 98
- FTCLOSE 97
- FTERASE 98
- FTINCL 97
- FOPEN 97
- included segment
 - define variable 27
 - delete variables 28
- LOG
 - employee records 26
 - telephone book 90
- panel EMPL
 - definition 25
 - display 25
- panel TPANEL1, display 88
- primary option menu panel 25
- SELECT 90
- SETMSG 91
- TBADD
 - employee records 26
 - telephone book 93
- TBBOTTOM 94
- TBCLOSE
 - employee records 26
 - telephone book 90, 99
- TBCREATE
 - employee records 26
 - telephone book 90, 99
- TBDELETE
 - employee records 26
 - telephone book 94
- TBDISPL 93
- TBEND 98
- TBERASE 99
- TBEXIST 96
- TBGET
 - employee records 26
 - telephone book 96
- TBMOD 95
- TBOPEN
 - employee records 26
 - telephone book 89, 98
- TBPUT
 - employee records 26
 - telephone book 94
- TBQUERY 96
- TBSARG 95
- TBSAVE 99
- TBSCAN 95
- TBSKIP 90, 94, 98
- TBTOP 93

- TBVCLEAR 95, 96
- TPANEL1 30
- variable definitions 27
- VCOPY 96
- VDEFINE
 - employee records 28
 - telephone book 92
- VDELETE
 - employee records 28
 - telephone book 92
- VGET 92
- VPUT 90, 92
- VREPLACE 97
- VRESET 91
- p11 implementation
 - employee records 25
 - telephone book 86
- primary option menu 1
- primary option menu panel EMPL
 - clist 8
 - cobol 15
 - exec2 11
 - fortran 20
 - p11 25

S

- SELECT
 - clist 42
 - cobol 64
 - exec2 52
 - fortran 78
 - p11 90
- sequential
 - data set iv
 - file iv
- SETMSG
 - clist 42
 - cobol 65
 - exec2 52
 - fortran 78
 - p11 91

T

- table display panel, TPANEL2 35
- TBADD, clist
 - employee records 10
 - telephone book 43
- cobol
 - employee records 20

- telephone book 68
- employee records 3
- exec2
 - employee records 13
 - telephone book 54
- fortran
 - employee records 23
 - telephone book 81
- p11
 - employee records 26
 - telephone book 93
- TBBOTTOM
 - clist 44
 - cobol 69
 - exec2 55
 - fortran 82
 - p11 94
- TBCLOSE, clist 42
 - employee records 11
 - telephone book 47
- cobol
 - employee records 18
 - telephone book 64, 73
- exec2
 - employee records 14
 - telephone book 52, 59
- fortran
 - employee records 24
 - telephone book 78, 86
- p11
 - employee records 26
 - telephone book 90, 99
- TBCREATE, clist
 - employee records 9
 - telephone book 41, 47
- cobol
 - employee records 18
 - telephone book 64, 73
- employee records 2
- exec2 59
 - employee records 12
 - telephone book 51
- fortran
 - employee records 22
 - telephone book 77, 86
- p11
 - employee records 26
 - telephone book 90, 99
- TBDELETE, clist
 - employee records 10
 - telephone book 44
- cobol
 - employee records 20

- telephone book 69
- employee records 3
- exec2
 - employee records 14
 - telephone book 55
- fortran
 - employee records 24
 - telephone book 82
- p11
 - employee records 26
 - telephone book 94
- TBDISPL
 - clist 43
 - cobol 68
 - exec2 54
 - fortran 81
 - p11 93
- TBEND
 - clist 46
 - cobol 73
 - exec2 59
 - fortran 86
 - p11 98
- TBERASE
 - clist 47
 - cobol 73
 - exec2 59
 - fortran 86
 - p11 99
- TBEXIST
 - clist 45
 - cobol 70
 - exec2 57
 - fortran 83
 - p11 96
- TBGET, clist
 - employee records 9
 - telephone book 45
- cobol
 - employee records 20
 - telephone book 71
- exec2
 - employee records 13
 - telephone book 57
- fortran
 - employee records 22
 - telephone book 84
- p11
 - employee records 26
 - telephone book 96
- TBMOD
 - clist 44
 - cobol 69

exec2 56
 fortran 82
 pl1 95
 TBOPEN, clist
 employee records 9
 telephone book 41, 46
 cobol
 employee records 18
 telephone book 63, 73
 exec2
 employee records 12
 telephone book 51, 59
 fortran
 employee records 22
 telephone book 77, 86
 pl1
 employee records 26
 telephone book 89, 98
 TBPUT, clist
 employee records 10
 telephone book 44
 cobol
 employee records 20
 telephone book 69
 employee records 3
 exec2
 employee records 14
 telephone book 55
 fortran
 employee records 23
 telephone book 82
 pl1
 employee records 26
 telephone book 94
 TBQUERY
 clist 45
 cobol 71
 exec2 57
 fortran 84
 pl1 96
 TBSARG
 clist 47
 cobol 70
 exec2 56
 fortran 83
 pl1 95
 TBSAVE
 clist 47
 cobol 73
 exec2 59
 fortran 86
 pl1 99
 TBSCAN
 clist 47
 cobol 70
 exec2 56
 fortran 83
 telephone book 83
 pl1 95
 TBSKIP, clist 41, 44
 telephone book 46
 cobol 64, 69, 73
 exec2 51, 59
 fortran 77, 82, 86
 pl1 90, 94, 98
 TBTOP
 clist 43
 cobol 68, 70
 exec2 54, 56
 fortran 81, 83
 pli 93
 TBVCLEAR
 clist 45, 47
 cobol 70
 exec2 56, 57
 fortran 83
 pl1 95, 96
 telephone book 29
 clist implementation 38
 cobol implementation 60
 common dialog components 34
 CPANEL1
 clist 30
 exec2 30
 dialog components 30
 exec2 implementation 48
 file tailoring skeleton 38
 fortran implementation 73
 FPANEL1, fortran 30
 invocation 30
 ISPSTART command 31
 main function routine 33
 messages 37
 organization 32
 overview 29
 pl1 implementation 86
 processing description 33
 TBSKIP, clist 44
 TPANEL1
 cobol 30
 pl1 30
 terminology
 command procedure iv
 file iv
 library iv
 TPANEL1, cobol, display 62

cobol definition 60
p11 definition 87
p11 display 88
telephone book 30, 34
TPANEL2 panel
definition 35
display 35
TPANEL3 panel
definition 36
display 36

V

variable definitions, p11 27
variables, cobol 16
VCOPY
cobol 71
fortran 80, 84
p11 96
VDEFINE, cobol
employee records 17
telephone book 67
fortran
employee records 21
telephone book 80
p11
employee records 28
telephone book 92

VDELETE
cobol 67
fortran
employee records 24
telephone book 80
p11
employee records 28
telephone book 92

VGET
clist 43
cobol 67
exec2 53
fortran 80
p11 92

VPUT
clist 42, 48
cobol 64, 67
exec2 52, 54
fortran 77, 80
p11 90, 92

VREPLACE
cobol 71
fortran 84
p11 97

VRESET, cobol
employee records 18
telephone book 65
fortran 78
p11 91

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

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

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

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

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

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

Cut or Fold Along Line

What is your occupation? _____

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

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

ST34-2085-0
(Formerly SC34-2085-0)

Reader's Comment Form

Cut or Fold Along Line

Fold and tape

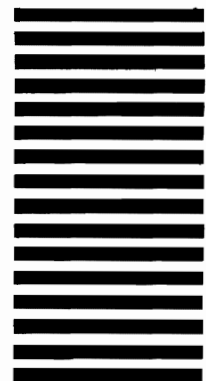
Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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



POSTAGE WILL BE PAID BY ADDRESSEE:

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

Fold and tape

Please Do Not Staple

Fold and tape

ISPF: Dialog Management Services Examples

Printed in U.S.A.

ST34-2085-0
(Formerly SC34-2085-0)



ST34-2085-0
(Formerly SC34-2085-0)

ISPF : Dialog Management Services Examples

Printed in U.S.A.

ST34-2085-0
(Formerly SC34-2085-0)

