ISPF 101: Introduction to Panels, Messages and Variables

Clive Nealon
IBM Global Services Australia
SHARE 98, Winter 2002
cliven@au1.ibm.com

SHARE Nashville - ISPF Project Grid

IBM Software Solutions

	8:00am	9:30am	11:00am	1:30pm	3:00pm	4:30pm	6:00pm
Sunday March 3						**2600** 90 mins ISPF and SCLM Suite Open / Reqs Review ⦿	
Monday March 4	**2400** MVS Program Opening	**2601** ⦿ ISPF and SCLM Suite Trends and Directions	**0001** General Session	**2602** ○ ISPF 101: Introduction to Panels and Messages	**2603** ○ ISPF 102: Introduction to Skeletons and Tables	**2604** ○ ISPF 103: Introduction to ISPF Dialogs	**2608** ○L ISPF laboratory
Tuesday March 5	**2634** ⦿ IBM's SCLM Suite	**2635** ⦿ SCLM Suite 101: Intro. to Library Mgmt	**2636** ⦿ SCLM Suite 102: SCLM in Action	**2637** ⦿ SCLM Suite 103: SCLM Admin.	**2638** ⦿L SCLM Suite 104: Laboratory	**2632** ○ Dynamic ISPF	
Wednesday March 6				**2639** ⦿ SCLM Suite User Session	**2640** ⦿ SCLM Suite and InfoMan Integration Story	**2644** ⦿ SCLM Suite User Group	
Thursday March 7	**2625** ○ ISPF Hidden Treasures	**2627** ○ ISPF Panel Processing	**2605** ○ ISPF Behind the Scenes	**2646** ○ ISPF One Hour – Two Topics	**2649** 90 mins ISPF and SCLM Suite Reqs & Close ⦿		

ISPF and SCLM Suite sessions are in **Ryman Chambers A/B/C**, **except labs** (marked **L**), which are in **Presidential Chamber B**.

○ Attendance counts towards ISPF certification only

⦿ Attendance counts towards ISPF and SCLM Suite certification

Agenda

- ISPF Overview
  - What ISPF Does
  - Where to store the parts of a 'dialog'

ISPF Variables
  - Pools
  - Services

- Messages

- Panels

What ISPF Does

ISPF is the interface between your program and the user on TSO.

1. Present a screen to the user (called a Panel)
2. Get user input (into one or more variables)
3. Process the input.
4. Set messages and respond to the user.

Parts of an ISPF Dialog

Panels - Define the image on the screen. They can also provide some basic application logic.

Messages - Provide simple feedback to the user.

Skeletons - Provide a basic mechanism for formatting data to an output file. Usually used to generate JCL or listings.

Parts of an ISPF Dialog

Tables - Provide a means of storing data. Intended for relatively small amounts of data (Not meant to be a database)

Images (GUI only)

Parts of an ISPF Dialog

And of course...

Your program - Can be REXX, CLIST, Compiled language (COBOL, PL/I, ASSEMBLER, etc.), or APL.

Any language which can use 'standard linkage conventions' will work.

All ISPF services are called through **ISPEXEC** and **ISPLINK**

Parts of an ISPF Dialog

ISPF parts live in PDSes allocated to specific DD names. Each part is a member of a PDS.

ISPxLIB

where 'x' defines the type of the part...

Parts of an ISPF Dialog

ISPPLIB - Panels

ISPMLIB - Messages

ISPSLIB - Skeletons

ISPTLIB - Tables

ISPPROF - Tables

ISPTABL - Tables

These have special properties

ISPILIB - Images

Parts of an ISPF Dialog

REXX programs reside in SYSPROC or SYSEXEC.

CLISTs reside in SYSPROC.

Load modules are in ISPLLIB, STEPLIB, Linklist, LPA, or other places.

ISPF does not control these libraries.

Parts of an ISPF Dialog

There are other DD names that may be needed.

There are other ways to dynamically allocate parts of a dialog (LIBDEF, ALTLIB, TSOLIB, and other dynamic allocation products and service).

Tip: Use the **TSO ISRDDN** command to find parts (there is a help panel - F1)

Tip: Use existing parts as examples. Especially panels.

ISPF Variables

ISPF Dialogs rely on variables

Names are 8 characters.

Many formats...

FIXED	(numeric)
CHAR	(character)
...	
USER	(you decide)

ISPF Variables

Names starting with Z are reserved by ISPF.

Examples:

&Z = null (used for testing for empty variables)

&ZUSER = userid

&ZTIME = current time

&ZSCREEN = split screen number

&ZWIDTH = screen width

Tip: In panels, messages and skeletons, variables are usually represented with an ampersand prefix.

ISPF Variables

In REXX or CLIST, ISPF uses the REXX/CLIST variables (you don't need to do anything special).

- ➡ Only those with names of 8 characters or less.
- ➡ Stem variables are not allowed as ISPF variables.

ISPF Variables

IBM Software Solutions

Compiled programs use local storage as a location for ISPF variables.

VDEFINE - Associate program storage with a name.

VDELETE - Remove the association.

It is VERY IMPORTANT to have one VDELETE for each VDEFINE.

ISPF Variables

```
Call ISPLINK ( 'VDEFINE  ',
               'MYVAR1   '
               pgmvar,
               'CHAR',
               56);
       ... other services using the variable...
Call ISPLINK ('VDELETE ','MYVAR1 ');
```

Creates ISPF variable MYVAR1 over program variable pgmvar.

ISPF Variable Pools

IBM Software Solutions

Variables pools define sharing and persistence of variables.

Function pool - known only to your program

Shared pool - Can be shared between programs within an application (without sharing storage)

Profile pool - Can be shared between programs and saved across ISPF sessions. (Saved in DD name ISPPROF).

ISPF Variable Pools

Variable Services:

VGET - copy a variable from shared or profile pool.

VPUT - Copy a variable to shared or profile pool.

VCOPY - Copy variable contents from any pool.

VREPLACE - Create an 'implicit' variable from data at any storage location.

ISPF Variables

VGET and **VPUT** assume you did a VDEFINE or are using REXX or CLIST.

VCOPY and **VREPLACE** just copy data from and to storage but do not require the actual storage association (VDEFINE).

These two are not available in REXX or CLIST.

ISPF Messages

Messages are named:

ppppnnns

- Prefix: one to five alphabetic characters (A-Z, #, $, or @)
- Number: three numeric characters (0-9)
- Suffix (optional): one alphabetic character.

Examples:

ISRE016, EXIT532G, SK@949A

ISPF Messages

Message member names are determined by truncating the message ID after the second digit of the number.

Example:

Member **GTE08** may contain messages **GTE08**0 through **GTE08**9Z

ISPF Messages

Messages are used by the following services:

SETMSG - Show a message on the next display.

DISPLAY - Show a message as part of displaying a panel.

TBDISPL - Show a message as part of displaying a table.

LOG - Write a message to the ISPF log.

ISPF Messages

You can also read a message into a program with the **GETMSG** service.

This is an easy way to get simple national language translation capability for small amounts of translatable data.

ISPF Messages

ISPF Messages have:

Short message:

Up to 24 characters

Long message:

Up to 512 characters

Tip: Use the long message to tell how to fix the problem, not just to tell what went wrong.

Attributes: Alarm, associated help panel, severity of message, etc..

ISPF Messages

Continuation

Continue long messages with a SPACE-PLUS after the line to be continued.

```
'This is line one followed by ' +
'line two.'
```

ISPF Messages - Example

```
NEXT000 'No matching names'
'There were no data set names matching the specified' +
'pattern (&pattern).

NEXT001 '               '
'The name &nextname was retrieved using pattern "&PATTERN".'

NEXT002 'No more names'
'There were no additional ' +
'data set names matching the specified pattern (&pattern).'
```

Note that NEXT001 has no short message.

ISPF Messages

Examples:

ISPEXEC **SETMSG** MSG(EJG331)

ISPEXEC **DISPLAY** PANEL(MYPAN) MSG(EED001)

ISPEXEC **LOG** MSG(EJD331L)

ISPF Panels - Overview

Panels define what your users will see on the screen.

- Output fields
- Input fields
- Colors
- Variables displayed or used for input

Panel logic provides some 'smarts'

ISPF Panels - Four types of panels

Selection panel

Can invoke programs, TSO commands, or other selection panels. Usually used for primary panels.

Invoked via ISPEXEC SELECT service

(Always have an assignment to **&ZSEL**)

```
ISPEXEC SELECT PANEL(ISR@PRIM)
```

ISPF Panels - Four types of panels

Display panels

Used for basic input and output operations.

An example is the Option 2 edit input panel.

Displayed via the DISPLAY service

```
ISPEXEC DISPLAY PANEL(EDITINP)
```

ISPF Panels - Four types of panels

Table Display Panels

Used to display formatted ISPF Tables.

An example is the Option 3.9 display (Command tables).

Displayed via the TBDISPL service

```
ISPEXEC TBDISPL TABLE(MYTABLE) PANEL(MYTBPANL)
```

ISPF Panels - Four types of panels

Tutorial Panels

Used to display Help panels.

Displayed when the HELP key is pressed. Specified in Panels or on Messages.

Basically the same as Display panels with a few extra keywords.

Tip: You can view a specific help panel by typing TUTOR *panel* on any ISPF command line.

ISPF Panels

Panels reside in DD name **ISPPLIB**

Each panel is a member.

Services use the member name to refer to the panel.

Input on panels ends up in ISPF Variables.

ISPF Panels - The basic sections

Panels have several sections:

)ATTR

Defines attribute bytes, e.g. input, output, colors

)BODY

Defines what the panel will look like

)INIT,)REINIT,)PROC

Panel Logic (simple setup, verifications, etc.)

)END

Defines the end of the panel definition (required)

ISPF Panels -)ATTR

The **)ATTR** section defines *Attribute bytes*

Attribute bytes, used in the **)BODY** section, indicate what follows:

# TYPE(TEXT) COLOR(RED)

means text following this byte is red.

@ TYPE(INPUT) CAPS(ON)

means an input field is beginning and the field will translate input to upper case.

ISPF Panels -)ATTR

Many different TYPEs:

TYPE(**TEXT**) - Text follows - may include
variables, e.g.: Hello &ZUSER

TYPE(**INPUT**) - Input field

TYPE(**OUTPUT**) - Output field (variable name)

CUA Types define colors based on the field type, such as panel title or normal entry field.

ISPF Panels -)BODY

The)BODY section is the 'WYSIWYG' section.

(what you see is what you get)

Use attribute bytes, plain text and variables.

```
)ATTR
  % type(text) intens(high)
  +  type(text) intens(low)
   _ type(input)
)BODY
%-------- Panel Title -----
%Command ===>_MYVAR     %
)END
```

%, + and _ are preset by ISPF. They are shown here as an example only.

This indicates the beginning of an input field. What the user types will be stored in variable MYVAR.

ISPF Panels - Panel Logic

Use Panel Logic in the)INIT,)REINIT and)PROC sections:

)INIT section - Run before panel display

)REINIT section - run before redisplay (usually before an error message set in the)PROC section is shown).

)PROC section - Run after the panel is displayed. Usually contains field verifications.

ISPF Panels - Panel Logic

Assignment statement

&MYVAR = &ZUSER

&MYVAR = 'Johnny is a &NICEGUY'

&CURSLOC = **.CURSOR**
(Control variables are special panel variables)

&ZSEL = **TRANS**(&COM 1,'CMD(A)' 2,&MYVAR)

ISPF Panels

Some control variables

.CURSOR - Field name where cursor is.

.CSRPOS - Cursor offset into field.

.HELP - Help panel associated with this panel

.MSG - Next message to display

.RESP - User or simulated response - END or ENTER.

.TRAIL - What remained after a TRUNC() operation

(TRUNC is a common function to pull options of the command line).

ISPF Panels

IBM Software Solutions

IF - ELSE

Indentation sensitive

Comments use /* */

```
IF (&A = 1,2,3)  /* if a is 1, 2 or 3 */
   &B = 2
   .MSG = XGG001
ELSE &B = 4  /* can be on same line as else */
&C= 'This is outside of the IF/ELSE logic */
```

Clause on same or next line(s)

ISPF Panels

IF - ELSE

The expression can include

Simple comparison : IF (&A = 3)

List if comparisons : IF (&A NE 4,5,6)

Use of control variables: IF (.MSG = &Z)

Imbedded verify: IF (VER (&DSN,NB))

ISPF Panels

IF - ELSE

= or EQ	Equal
¬= or NE	Not equal
> or GT	Greater than
< or LT	Less than
>= or GE	Greater than or equal
<= or LE	Less than or equal
¬> or NG	Not greater than
¬< or NL	Not less than

Tip: Panel language does **not** do math

eg : IF (&A=&C+1)

ISPF Panels

VER - The VERIFY statement

Used to Verify input fields:

```
VER (&A, NB,RANGE,6,19)

VER (&A,NB)

VER (&A,LIST,4,7,JOE,MARY, MSG=XYZ001)

VER (&PROJECT,DSNAME)
```

VER (variable [NONBL] *keyword* [,MSG=value])

Keywords:

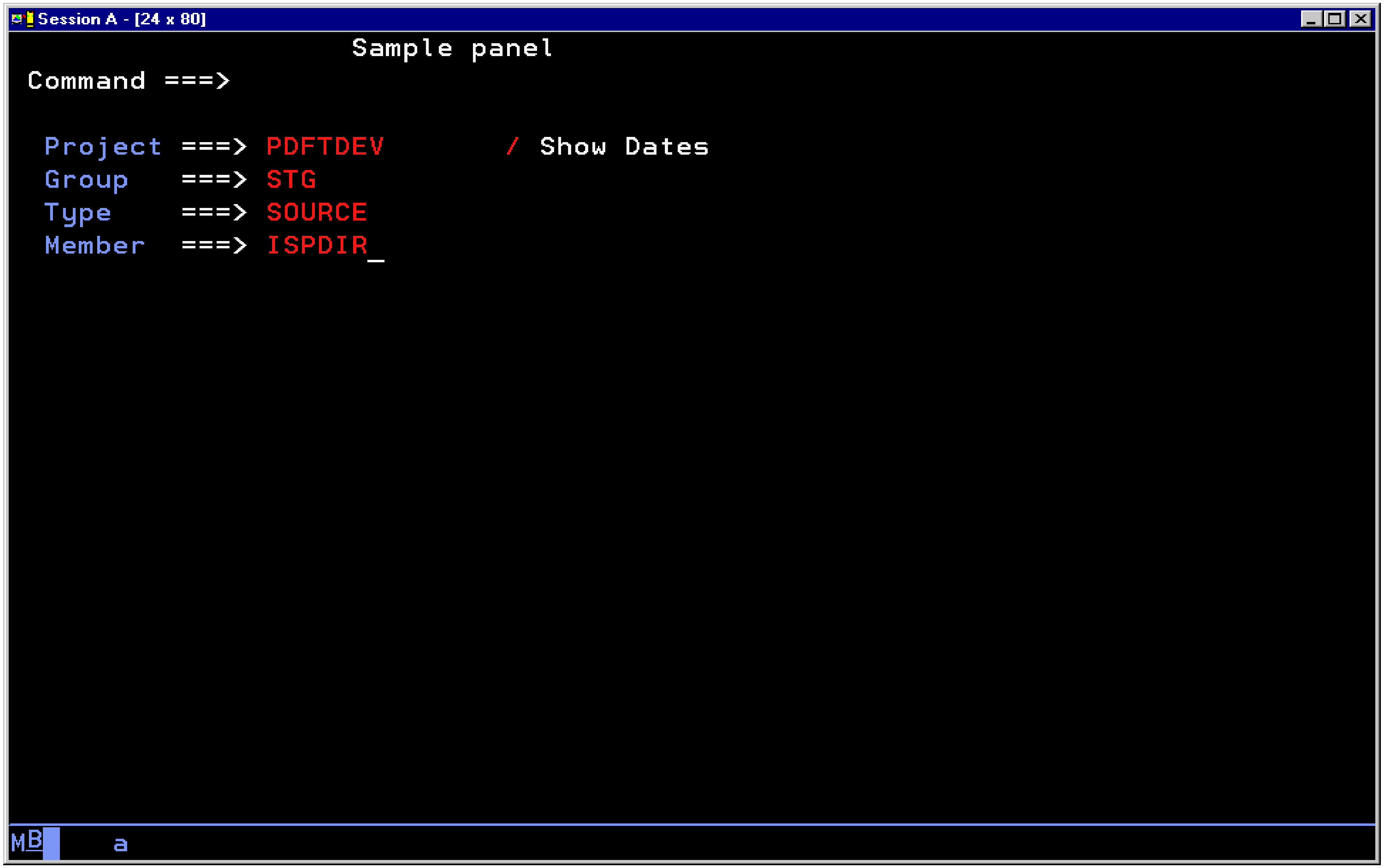
ALPHA	LEN,relational-operator,expected-length
ALPHAB	LIST,value1[value2...]
BIT	LISTV,varlist
DBCS	LISTVX,varlist
DSNAME	LISTX,value1,value2,...
EBCDIC	MIX
ENUM	NAME
FILEID	NUM
HEX	PICT,string
INCLUDE[IMBLK] value1[,value2]	RANGE,lower,upper

ISPF Panels

```
)BODY
%                   Sample panel
%Command ===>_ZCMD                                    +
+
+ Project%===>_PRJXXX  +     _Z%Show Dates
+ Group  %===>_G1      +
+ Type   %===>_TYPENAME+
+ Member %===>_MEMNAME +

)INIT
  &ZCMD = &Z            /* blank out command line */
  .ZVARS ='(SHOWDATE)' /* Z used for long names  */
  VGET (PRJXXX G1 TYPENAME MEMNAME showdate) SHARED
  IF (&SHOWDATE NE &Z) &SHOWDATE = '/'
  IF (.MSG EQ &Z) .CURSOR = MEMNAME /*set cursor */
)REINIT
  REFRESH (*)
)PROC
  IF (&SHOWDATE NE &Z) &SHOWDATE = '/'
  VER (&PRJXXX  ,NB,NAME)
  VER (&G1      ,NB,NAME)
  VER (&TYPENAME,NB,NAME)
  VER (&MEMNAME ,NB,NAME)
  VPUT (PRJXXX G1 TYPENAME MEMNAME SHOWDATE) SHARED
)END
```

ISPF Panels



```
                    Sample panel
Command ===>

 Project ===> PDFTDEV        / Show Dates
 Group   ===> STG
 Type    ===> SOURCE
 Member  ===> ISPDIR_
```

ISPF Panels - Testing

Use Dialog Test

Option 7.2 - displays panels and messages

Start ISPF in TEST mode from the TSO READY prompt to insure that you get the latest copy of each panel.

```
ISPF TEST
```

More information

ISPF Dialog Developers Guide and Reference

Use ISRDDN to find existing panels

Use the ISRDTLCV edit macro to make DTL generated panels more readable.

Copy, Steal, Modify, Change, Pillage, ...

Internet Resources...

List Servers:

ISPF-L list server

send note to listserv@listserv.nd.edu with

SUBSCRIBE ISPF-L yourname

in the body of the note

ISPF-L is shadowed as newsgroup bit.listserv.ispf-l but you must be subscribed to post to the newsgroup.

IBM-MAIN list server

send note to listserv@bama.ua.edu with

SUBSCRIBE IBM-MAIN yourname

in the body of the note.

Search for listserver information at http://www.lsoft.com

Newsgroups:

On **news.software.ibm.com**

ISPF newsgroup
ibm.software.ispf

SCLM newsgroup
ibm.software.sclm

Internet Resources...

Useful Web Sites:

http://www.software.ibm.com/ad/ispf
ISPF home page (not very active)

http://www.redbooks.ibm.com
Redbooks produced for ISPF and SCLM

http://somebody.home.mindspring.com
OS/390 and ISPF Tools and toys (lots of ISPF extensions, some samples)
maintained by Doug Nadel

http://www.cbttape.org
The CBT tape, maintained by Sam Golob

http://planetmvs.com
A good starting point for everything OS/390...
maintained by Dave Alcock