

**ISPF
STANDARDS
AND
GUIDELINES**

GUIDE ISPF PROJECT

NOVEMBER, 1987

RELEASE 1 - REVISION 2

GUIDE ISPF PROJECT

ISPF STANDARDS TASK FORCE

First Printing: March 15, 1985

Co-Chairmen:

Cindy Stegemeier
Pennzoil Co.
P. O. Box 2967
Houston, Texas 77252-2967
(713) 546-8318

Brian Vohs
Exxon Company, USA
P. O. Box 2180
Houston, Texas 77001
(713) 965-7534

ISPF STANDARDS AND GUIDELINES REVIEW TASK FORCE

First Revision: March 17, 1987

Co-Chairmen:

Alex Drescher
Rohm and Haas Company
Independence Mall West
Philadelphia, PA 19105
(215) 592-6834

Chris Gerken
Exxon Company, USA
P. O. Box 2180
Houston, Texas 77001
(713) 966-6277

Project Manager: Bob Zimmerman
CNA Insurance
CNA Plaza
Chicago, IL 60685
(312) 822-5719

Table of Contents

Management Summary	v
Introduction	1
Naming Standards	3
General	3
Requirements	3
Identifiers	4
Some Naming Standards Now in Use	6
Panels	9
Overview	9
Panel Layout	9
Design Considerations	18
Format for Specific Panel Types	21
Messages	31
Tables	35
File Tailoring	39
Models	41
User Customization of System-Wide Dialog Components	43
Concatenation Sequences for ISPF Libraries	45
CLISTS	49
Appendix A: Contributors	51
Bibliography	53
Index	57
Reader Comment Card	63

Table of Contents

Management Summary

The Interactive System Productivity Facility (ISPF) is an IBM product that improves productivity in the development of interactive applications. Introduced in 1981, use of ISPF has grown tremendously during the past several years. This growth can be attributed to the variety of user application requests that can be satisfied using ISPF. Because of the expanding usage of ISPF, it is becoming increasingly important for installations to have a method of standardizing ISPF development.

GUIDE membership has recognized and addressed this need by developing an ISPF standards and guidelines document. This document, which represents the combined expertise of many corporations, contains concise and clear development standards that reflect the experiences of corporations from around the country. Use of the "GUIDE ISPF Standards and Guidelines" document will bring to your shop a formalized ISPF development methodology that will maximize the usage of ISPF dialogs and improve the productivity of your dialog developers.

Listed below are some of the many benefits that you will realize by incorporating these standards and guidelines into your environment.

1. **AUTOMATIC STANDARDS REVISIONS**

With each new release of ISPF, GUIDE membership reviews and updates the document to keep it current with the product's technology. Therefore, GUIDE will insure that your installation's standards will keep pace with your ISPF environment.

2. **REDUCED DEVELOPMENT AND MAINTENANCE COSTS**

The document defines a repeatable approach to dialog development so that dialog developers do not have to "re-invent the wheel" with every new ISPF dialog application. This reduces application development time. Maintenance costs are also reduced because the dialog developer spends less time becoming acquainted with each application.

3. **APPLICATION PORTABILITY**

Adherence to standards promotes consistency between applications. It enables dialogs to remain uniquely identifiable while allowing them to function in a shared environment. This promotes portability; from user to user, CPU to CPU, and data center to data center.

4. **PRODUCT COMPATIBILITY**

IBM, as well as many of the more prominent vendors, conform to dialog development standards similar to those outlined in the "GUIDE ISPF Standards and Guidelines" document. By adhering to these standards, your applications will use conventions like those used in most major products available in today's market. This enables your environment to be more "user friendly" by promoting consistency between products and applications. It also reduces user training requirements because applications look and function in a similar

Management Summary

manner.

The "GUIDE ISPF Standards and Guidelines" publication is available to all GUIDE members. Use of it will enable your shop to take advantage of the latest techniques in ISPF dialog development. It will truly be an asset to your environment.

1. INTRODUCTION

This publication is intended to meet the needs of ISPF installations which are developing their own ISPF dialogs or installing IBM or other vendor products which use ISPF as their dialog manager. This is a continually evolving document and will be updated as necessary.

The objectives of this publication are to:

1. Establish STANDARDS for ISPF dialogs based on widely accepted practices which are consistent with the structure of the PDF product or use an ISPF or PDF function which has certain requirements. It is strongly recommended that each installation adopt and enforce these practices for their own internal standards.
2. Suggest GUIDELINES (naming conventions, library concatenations, making changes to system-wide dialog components, etc.) based on widely accepted practices which will enhance the usability and efficiency of PDF based user or vendor dialogs.

This document addresses standards for dialogs which operate in a PDF environment since PDF has the broadest base of users. Major emphasis is placed on how a dialog looks to the user (panel layouts, message formats, etc.). Other uses of ISPF dialog manager, such as developing prototypes for any on-line application (i.e. CICS, IMS, etc.), do not need to adhere to the standards and guidelines proposed in this document.

Points which are considered STANDARDS are identified by *bold, italicized* text. All other points are considered guidelines.

2. NAMING STANDARDS

2.1 GENERAL

Naming standards must be established for the member names of ISPF components (panels, messages, CLISTs, etc.) to prevent member name conflicts. Member names should also identify the dialog or application in which the member is used or the group which developed the dialog or application.

Conflicting member names are a concern because most installations run ISPF as a concatenation of libraries from different sources. The IBM ISPF and ISPF/PDF libraries are concatenated with other vendor ISPF libraries and with libraries of dialogs developed at the installation. Some installations have many different ISPF libraries corresponding to different groups of users. If no naming convention has been established, libraries which contain members of the same name may be concatenated. When multiple members of the same name exist in a concatenation, only the first member found with the specified name is accessible.

There may not be any one naming standard which is appropriate for every installation. The purpose here is to propose several possible naming standards and discuss their relative advantages and disadvantages.

Naming conventions should be established for all parts of the dialog (panels, messages, skeletons, CLISTs, tables, programs). Naming conventions should be established for all Profile pool variables within a given profile (identified by the NEWAPPL id). Naming conventions should be established for all Shared pool variables within a given NEWPOOL.

2.2 REQUIREMENTS

Any naming standard should meet the following requirements:

1. Member names must be unique across any libraries which are concatenated together. If the System Modification Program (SMP) is to be used to maintain the dialogs, each distinct component of each dialog must be uniquely named across all libraries that SMP will maintain.
2. The member name should identify the dialog or application which the component belongs to. The naming standard should also accommodate components which are used by many different dialogs.

3. The naming standard should use as few characters as possible, so that the dialog developer has some characters for his/her own use. Because the length of the names is limited to 8 characters, each character required by the naming standard must be justified.

In addition to the above, the naming standard should, ideally, encourage the use of mnemonics in names. Well-chosen mnemonics are useful in communicating information about the function of the component.

2.3 IDENTIFIERS

The member name should include some of the following identifiers:

1. SYSTEM ID - identify the system to which the dialog belongs

A system id identifies the application system but not the dialog to which a specific component belongs. A list of system ids must be maintained to ensure uniqueness.

Dialog components should not be named with a prefix of either ISR or ISP since this may conflict with elements distributed by IBM.

2. GROUP ID - identify the group or department which developed the dialog

If group id is used, the group which is responsible for the dialog is easily identified when problems occur or changes need to be made. This may not be important if each group has its own set of ISPF libraries concatenated. A list of group ids must be maintained to ensure uniqueness.

3. DIALOG ID - identify the dialog to which the member belongs

Member names which identify the dialog to which the member belongs are useful in maintaining the ISPF libraries. All components of a dialog can be identified if each member name contains a dialog ID. When a problem with a specific member is found, it is easy to determine which dialog and which other ISPF components may be affected. If dialog id is used, a list of dialog ids must be maintained to ensure uniqueness.

Dialog id could be used in conjunction with system id or group id to further identify the component. For example, system id or group id could be defined as a 3-character code and dialogs within each system/group could be identified by a 1-character code.

4. TYPE - an identifier indicating the type of ISPF component (panels, clists, skeletons, etc.). In some cases this could even be broken down further; types of panels, for example.

The type indicator is useful in naming components which are maintained by SMP (because every member name must be unique across all libraries maintained by SMP). It can also be useful in differentiating between the different types of panels within a panel library.

2.4 SOME NAMING STANDARDS WHICH ARE IN USE NOW

It is important that each installation develop a standard for naming dialog components that will fit their needs. Some guidance is provided in this section through examples to assist in this development.

Because of the connection between message-ids and message member names, some of the following naming conventions cannot be applied to message member names.

The number of characters specified in the following naming standards are given as an example. Each installation will have different requirements for the number of characters used for an identifier.

1.

Characters Value

1-4	Dialog ID / Group ID / System ID
5	Type of component
6-8	specified by developer

Comments - this convention allows the dialog developer to choose a meaningful naming scheme for part of the member name, while still providing information which is useful in identifying the component.

2.

Characters Value

1-3	Dialog ID / Group ID / System ID
4	Type of component
5-8	each character is an alphameric character which indicates the option number at the 1st, 2nd, 3rd and 4th level of the dialog hierarchy

Advantages - the member name indicates where the ISPF component belongs in the dialog hierarchy.

Disadvantages - If the organization of the dialog is changed, the member names would need to be changed to reflect the new hierarchy.

3.

Characters Value

1	Type of component
2-5	Dialog ID / Group ID / System ID
6	alphanumeric character
7-8	Version

CHAPTER: 2 - Naming Standards

Advantages - Version may be useful information for some installations.

Disadvantages - Only 1 character is left for the developer to specify. Also, in a member selection list, the different types of panels which belong to a dialog are not grouped together.

4.

Characters Value

1	Installation Code
2-4	Dialog ID / Group ID / System ID
5	Type of component
6-8	specified by developer

Comments - Installation code may be useful information for installations which support more than one operating system.

5. IBM has no PUBLISHED naming standards for their ISPF member names. For the ISPF and ISPF/PDF products, they GENERALLY follow a convention of

Characters Value

1-3	ISR or ISP, depending upon the product to which the member belongs
4	ISPF/PDF option ID (e.g. E for Edit)
5-8	free-form

Installations should not begin member names of dialog components with the characters ISR or ISP because of this convention.

CHAPTER: 2 - Naming Standards

3. PANELS

3.1 OVERVIEW

1. Panels should be as similar in appearance as possible.
2. Where differences exist between the way two panel types work, there should be differences in appearance between the two types. These differences will serve as visual clues into the function available on/with the different panel types.
3. The above applies to data/information fields as well.

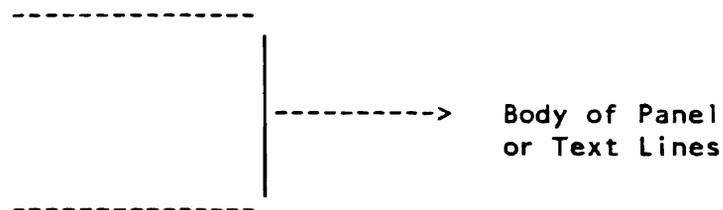
3.2 PANEL LAYOUT

1. FORMAT DIAGRAM

The panel format in Figure 1 provides a basis for the following discussion on panel layouts. The terms used in the description of each panel type are defined below.

----- TITLE LINE -----
COMMAND ==>

Function and/or Instruction Line



(Directive Line)

Figure 1. GENERAL PANEL FORMAT

2. TITLE LINE (LINE 1)

- a. *Should be the first line on the panel.*

- b. *Should be displayed in a high intensity output field.*
 - c. *Should be in upper case.*
 - d. *The title should be centered on the title line.*
 - e. *Separate the title from the dashes by at least 1 (and preferably 2) blanks.*
 - f. *The word TUTORIAL should appear on both ends of the line for tutorial panels.*
 - g. *Always use the default message area to display messages. This is an important part of the format to which dialog users have become accustomed. The default area for the short message is the upper right hand corner of the panel (line 1).*
 - h. *Short messages should always be displayed with all 24 characters (right justify the message and pad with blanks).*
3. **COMMAND LINE (LINE 2)**
- a. *All panels must have a selection "option," or "command" field. The "selection" field is used on tutorial panels, the "option" field is used on menu panels, and the "command" field is used on all others. The only allowable formats are:*
 - Selection ==>
 - or
 - Option ==>
 - or
 - Command ==>
 - b. *Should be in upper case.*
 - c. *Should be displayed in a high intensity output field.*
 - d. *The input field should only be terminated at the beginning of the next line.*
 - e. *On scrollable panels, the command line contains a 4-byte scroll field. It should begin in column 64 and have the following text field precede it:*
 - SCROLL ==>
4. **LONG MESSAGE LINE (LINE 3)**
- a. *Should always be blank. (Some non-essential output fields may be positioned on this line. This is the normal position for a time-of-day field).*

- b. Time-of-day field, if displayed, should be placed on the third line on the right hand side of the screen.

5. FUNCTION LINE (OPTIONAL, LINE 4)

- a. Can be used to display a subtitle on a panel, if one is necessary.
- b. Should be in upper and lower case, centered on the line.
- c. Should be displayed in a high intensity output field.

6. INSTRUCTIONS

- a. Can be used with or without a function line. To provide the user with information on what is to be entered on a panel when it may not be clear from the content of the panel. Instructions should be low intensity, upper and lower case. The position of the instructions depends on panel type, unless otherwise noted.
- b. *They should be used only when necessary and be as brief as possible and should always be in an active form. Use tutorial and help panels for lengthy explanations.*

The following panels have unique requirements:

1) Scrollable Panels (specifically their display).

Consist of one or more lines which instruct the user on the function of this panel.

Should include a list of the valid select codes (line commands).

2) Tutorial Panels

Can be used anywhere on the panel after text lines and before select option (if any).

Used specifically in instructing the user on navigating through the tutorial.

Should read: "The following topics are presented in sequence, or may be selected by number:" or "The following topics are presented only if selected by number:"

7. BODY OF PANEL

a. General

The layout and format of the body of the panel varies, depending on function. Specific body requirements for each type follows in the next chapter.

The Types are:

1) Non-Scrollable

Data Entry

Multi-Column Data Entry

Confirmation

Information

Tutorial Text

2) Scrollable/Dynamic

Table Display

Dynamic Panel/Graphic Areas

3) Selection

Menu

Tutorial Menu

4) Non-Display

Within the body there are input, output and text fields. Every panel will have some or all of these fields and they all must conform to the following:

- 1) Whenever possible the user should have a choice of attributes for a field.
- 2) Do not change the default attribute characters unless one of the default characters must appear within the text of the panel.
- 3) Input fields should be initialized to either a previously entered value, or a default value, if one can be determined.

4) Padding

- a) *Unused padding characters should not need to be removed by the user (i.e. use the PAD keyword in the)ATTR section of the panel to pad input fields).*
 - b) *If it is necessary to pad an input field with something other than null characters, always pad with the user's defined pad character unless special application requirements dictate otherwise. This is mainly if the user MUST see the exact length of the input field.*
- 5) *Be consistent in the use of displayed values (i.e. always use "Y" and "N" or "YES" and "NO").*
- 6) *To minimize keystrokes use TRUNC and TRANS in the)PROC section of the panel definition. If this is done, the same displayed values should always be truncated or translated consistently throughout the dialog.*

b. Specific Field Types: Format and Design Considerations

- 1) General
- 2) Input Fields: Singly Occurring

+Description Field%====>_Input Field + (Information)

a) Description field

- i. *Should be brief, limited to one line when possible and never exceeding two lines.*
- ii. *Should be lined up to the left of or if necessary placed just above the corresponding input field.*
- iii. *Should be displayed in a low intensity output field.*
- iv. *Should be in upper and lower case.*

b) Standard Arrow Symbol (====>)

- i. *All input fields should be immediately preceded by the ISPF standard arrow symbol (====>).*
- ii. *Should be used ONLY to precede input fields.*

- iii. *Should always be displayed in a high intensity output field.*
- iv. *Line up the arrows vertically, within logical groups, on the panel. If possible, line up all of the arrows on the panel. The text fields which precede the arrow symbols should be left-justified within logical groups.*

For example: SYSOUT CLASS ===>
 LOCAL PRINTER ID ===>

c) Input Field

- i. Should be displayed using high intensity.

d) Information

- i. *Used to display the possible values for the corresponding input field.*
- ii. *Should be displayed in a low intensity output field.*
- iii. *Should be in upper and lower case.*
- iv. The recommended values for Yes/No fields are Y and N.
- v. If there are only 2 valid entries for an input field, list the entries separated by 'or', (i.e. Y or N) to be consistent with ISPF/PDF panels.
- vi. To list more than two valid entries for an input field, separate the items by a comma and a blank. For example, ISPF/PDF Version 2 uses the following format on its 3.4 panel:

(QUICK, SHORT or LONG)

3) Input Fields: Repeating

a) For vertically repeating fields

- i. Should be arranged in a columnar fashion beneath their respective headings.
- ii. Most should be displayed using low intensity; highlight only the most important fields.

- iii. Should be padded with the user's defined pad character.
- iv. Align numeric data to the right when a table is displayed by storing the values in the table with the correct number of leading blanks or by using JUST(RIGHT).

b) For horizontally repeating fields

- i. Should be preceded by the standard ISPF arrow symbol (==>).
- ii. Should be displayed in a high intensity input field.
- iii. Should be padded with the user's defined character.

c) The Select Field

- i. Should be the left-most field for ease of use.
- ii. Should be displayed in a high intensity input field.
- iii. For functions that have ISPF/PDF equivalents, use the same code, i.g.

B - Browse
D - Delete
E - Edit
P - Print
S - Select

- iv. If selection codes differ from their use in ISPF/PDF, then the instructions should state this.

4) Output Fields

- a) *High intensity - should be used sparingly. Low intensity should be the predominant attribute for fields in the body of the panel.*
- b) *Should not be padded with any characters other than nulls or blanks.*

5) Text Fields

- a) *High intensity should be used sparingly. Low intensity should be the predominant attribute for fields in the*

body of the panel. The only universally highlighted sequence of characters are the title line, the command line, and the arrows preceding the input fields.

- b) Use upper and lower case in text fields; it has proven to be more readable than text which is all in upper case.

8. TEXT LINES

The text lines are used to give the user a brief description of the current dialog status. They are primarily used in informational and confirmation panels.

a. Informational

- 1) *Should be a brief explanation of the processing status.*
- 2) *Should be displayed using low intensity, and highlighting only important information.*

b. Confirmation

- 1) *Should be a brief explanation of the situation which requires confirmation.*
- 2) *Should be displayed using low intensity, and highlighting only important information.*

c. Tutorial

- 1) *Should highlight only important information.*
- 2) *When explaining a dialog input field, display the field on the tutorial panel as it appears on the dialog panel (except that it should be a text or output field).*

9. DIRECTIVE LINE

The directive line directs the user on how to proceed from this panel. The wording of the line should be consistent with ISPF/PDF and varies depending on panel type.

a. Non-Scrollable Panels

- 1) To indicate an action, display the following on line 23 (code so the entire message is on one line):

Press ENTER key to process;
Enter END command to terminate.

- 2) If ENTER and END meanings are different from normal PDF use, use two lines to display the instructions (this will help draw attention to the instructions):

Press ENTER key to -----
Enter END command to -----

where dashes indicate action. For example,

Press ENTER key to DELETE the data set
Enter END command to CANCEL the delete request

- 3) If two directive lines are necessary, place them on lines 22 and 23.
- 4) *The directive line(s) should be in the following format: the words END and ENTER are in upper case and displayed in high intensity output fields, the rest of the line(s) should be in upper and lower case and displayed in low intensity output fields.*
- 5) Example of a standard format directive line, coded using default attribute characters (code on one line):

+Press%ENTER+key to process;
Enter%END+command to terminate.

b. Scrollable Panels

- 1) Instructs the user on how to exit from the table display or from dynamic panel areas. Wording should be similar to non-scrollable.

c. Menus

- 1) A directive line may be used to explain how to return to a higher level selection menu.

d. Tutorials

- 1) *In tutorials, the directive line is only used on a panel which continues to another panel. The correct format of the directive line is: "(Continued on next page)"*
- 2) *Should be displayed in a high intensity output field.*
- 3) *Should be consistently placed within the body of the panel (preferably centered on line 23).*

3.3 DESIGN CONSIDERATIONS

1. *Installation-developed panels must be consistent with the format of ISPF/PDF panels.*
2. *To display large amounts of information, use more panels with less information per panel rather than fewer panels with more information per panel.*
3. *The majority of the panel should be displayed in low intensity output and in upper and lower case.*

Do not overuse high intensity. High intensity draws attention to important fields on a panel when it contrasts with the majority of the panel fields. Too much high intensity on a panel minimizes this effect.

4. *Be careful in the use of extended data stream attributes (color, highlighting, etc.) The user needs to know where to focus his/her attention on the screen. An excess of color and/or highlighting on a panel is confusing and can also cause eye fatigue. The use should determine the color attribute whenever possible. A panel design dependent on color may be invalid if any user of the panel has any form of color blindness.*
5. *All dialogs should make use of the HELP facility. All panels must have an associated HELP panel or, at the very least, a general HELP panel as opposed to no HELP at all.*
6. *Fixed panel formats should generally be designed for an installation's smallest screen size. Vendors must design their products to operate on the smallest screen ISPF supports (24 rows by 80 columns).*
7. *The last line of a "fixed" panel should always be blank to allow the user to split-screen without losing any information.*
8. *The dialog developer should also consider the PFSHOW command when designing panels. Up to four additional lines may be left blank at the bottom of the screen to accommodate PFSHOW output. An alternative is to place descriptive information which a more experienced user may not care about on these lines. PFSHOW output would then overlay an area of lesser importance or at least of lesser concern. For non-scrollable panels, input fields should line up vertically, if at all possible. This allows the most effective use of the cursor positioning keys.*

9. Be careful when using Dynamic Areas for ERROR/HELP information. Dynamic Areas are cannot be pre-processed and may hinder performance goals.
10. Identify panels with comments (who, where, calls who, called by, etc.) in the)ATTR section of the panel definition.
11. *All panel definitions should contain all sections - the)ATTR,)BODY,)REINIT,)INII and)PROC. Any sections which are not actually used by a panel should be left empty. Table display panels should also contain a)MODEL section. This can be re-enforced at an installation by encouraging the use of MODELS for panel development.*
12. Most used, or required input fields should be placed near the top of a panel with less important, lesser used, or optional fields lower down.
13. To minimize keystrokes use TRUNC and TRANS in the)PROC section of the panel definition. If this is done, the same displayed values should always be truncated or translated consistently throughout the dialog.
14. Minimize the use of lengthy application commands within a dialog. Panels should be used to build and issue any lengthy commands. For example, assume the following command is used to perform a search:

```
SEARCH TABLE FIRST (JOHN) MIDDLE (Q) LAST (DDE)
FOR (EMPLOYEE_NUMBER)
```

The user should not be required to enter the command. Instead, the dialog should use panels to gather the information and should construct and issue the command.

15. When designing a scrollable panel with a select field, put the select field on the left hand side of the screen.
16. Each line should begin with the panel's low intensity output attribute character unless some other attribute is necessary. This prevents unintentional continuation of a field from the previous line.
17. *When writing dialogs to be used by non-programmers, consider using the word "screen" instead of "panel" in panel text which refers to panels being displayed. "Screen" is the correct term for the physical image displayed on the terminal and is also more likely to be understood by non-programmers.*
18. For non-scrollable panels, input fields should line up vertically, if at all possible. This allows the most effective use of the cursor positioning keys.

19. For scrollable panels (table displays), each row should constitute a logical grouping of data.

Align numeric data to the right by storing the values in the table with the correct number of leading blanks or by using JUST(RIGHT).

3.4 FORMAT FOR SPECIFIC PANEL TYPES

1. Non-Scrollable Panels

a. Data Entry Panel

-----TITLE FOR DATA ENTRY PANEL-----
COMMAND ==>

Instructions

DESCRIPTION FIELD 1 ==> INPUT FIELD 1 (information)
DESCRIPTION FIELD 2 ==> INPUT FIELD 2 (information)
DESCRIPTION FIELD 3 ==> INPUT FIELD 3 (information)

Directive Line

----- (end of panel example) -----

b. Multiple Column Data Entry Panel

-----TITLE FOR MULTIPLE COLUMN DATA ENTRY PANEL-----
COMMAND ==>

Instructions

Heading 1	Heading 2	Heading 3	Heading 4
-----	-----	-----	-----
-----	-----	-----	-----
-----	-----	-----	-----
-----	-----	-----	-----

----- (end of panel example) -----

- Should be organized so that the user enters data across a row instead of down a column. In other words, logical groupings of data (e.g., data items related to a single record) should be input across a row, not down a column. The reason for this is that the cursor can be moved across the screen more easily than it can be moved down the screen.

Columnar Headings:

- *Should normally be left-justified.*
- *Should be displayed in a high intensity output field.*
- *Should be in upper and lower case.*

c. Information Panel

```
----- TITLE FOR INFORMATION PANEL -----  
COMMAND --->
```

Text lines

```
----- (end of panel example) -----
```

- When a dialog function executes for a long time without any user interaction (e.g. when searching a large file), an INFORMATION panel may be used to inform the user of the status of the processing. The panel itself is a standard panel containing only text fields. Before the panel is displayed, the CONTROL service is invoked with the DISPLAY and LOCK parameters. The panel is displayed with the keyboard locked and the function continues processing without waiting for any user response.
- Although the user cannot respond to the panel (other than by using the "break" key), the dialog has informed the user of its status. An information panel can be used to give the user an idea of how much processing has been done and how soon it will be finished.

d. Confirmation Panel

```
-----TITLE FOR CONFIRMATION PANEL -----  
COMMAND ==>
```

Text lines

Directive lines

```
----- (end of panel example) -----
```

- When a user must confirm an important function (e.g. file deletion, record update) in a dialog, a confirmation panel may be used. The text lines of a confirmation panel are used to describe the process or function of the task. The directive line directs the user and describes what process will occur based on user action or input.

e. Tutorial Text Panel

TUTORIAL-----TITLE FOR TUTORIAL TEXT PANEL----- TUTORIAL
COMMAND ===>

Tutorial Text

Directive Line

----- (end of panel example) -----

- *Set &ZUP on every tutorial panel.*
- Whenever necessary, use continuation tutorial panels to explain a dialog panel. Using more panels with less text per panel is more effective than using fewer panels with more text per panel.
- All dialogs should have an overview topic included in their tutorial. An overview is useful for orienting users who are unfamiliar with the dialog.

2. Scrollable Panels

a. Table Display Panel

```

----- TITLE FOR TABLE DISPLAY PANEL -----
COMMAND ==>                                     SCROLL ==>

Instructions

Directive Line

      Column2      Column3      Column4      Column5
Select  Heading      Heading      Heading      Heading
-----  -----      -----      -----      -----

----- (end of panel example) -----

```

- For table displays that use more than one screen line to display one row of the table, a line of dashes (or a blank line) may increase the readability of the display.
- Should be organized so that the user enters data across a row instead of down a column. In other words, logical groupings of data (e.g. data items related to a single record) should be input across a row, not down a column. The reason for this is that the cursor can be moved across the screen more easily than it can be moved down the screen.

Column Headings:

- *Should normally be left-justified.*
- *Should be displayed in a high intensity output field.*
- *Should be in upper and lower case.*

b. Dynamic Panels/Graphic Areas

- 1) Use of dynamic and graphic areas must always observe the standards for panel layout (e.g. title line, command line, fig.).
- 2) Dynamic areas can be used in conjunction with parts of other panel types and fields (singly occurring input fields on data entry panels, etc.) but the format must be consistent with that type of field.

3. Selection Panels

a. Menu Panel

-----TITLE FOR SELECTION MENU-----

OPTION ==>

Instructions

```
1  OPTION 1 -   Brief description of option 1
2  OPTION 2 -   Brief description of option 2
3  OPTION 3 -   Brief description of option 3
4  OPTION 4 -   Brief description of option 4
```

X EXIT

Directive Line

----- (end of panel example) -----

- *Align each of the three sections vertically.*
- *Put the options with the highest probability of being chosen at the top of the list to reduce reading time. If there is no obvious pattern for ordering options, put them in alphabetical or numeric order.*
- If both alphabetic and numeric options are used on a menu, list the numeric options in order first, followed by the alphabetic options, in order.
- *When the x (EXIT) option is displayed, place it near the bottom of the panel, distinctly separated from the rest of the options.*
- For those options which correspond to ISPF/PDF options, use the same Option Codes and Option Titles as are used in ISPF/PDF.

Option Lines:

- Option Code

Should be numbers (preferred) or capitalized letters.

Should be on separate lines.

Use Arabic numerals, not Roman numerals.

Should be displayed in a high intensity field.

Must not use '.' or ')' with the Option Code.

Use 'X' for EXIT only.

- Option Title

Should be in upper case.

- Option Description

Should be worded as a statement, not as a question.

Should begin with a capital letter.

b. Tutorial Menu

TUTORIAL--TUTORIAL SELECTION MENU TITLE--TUTORIAL
OPTION ==>

Tutorial Text

Instruction Line

- 1 - Brief Description of Option 1
- 2 - Brief Description of Option 2

----- (end of panel example) -----

- *Set &ZUP on every tutorial menu panel.*
- *Use a tutorial hierarchy that has a logical order (i.e. follows the hierarchy of the dialog). Ideally, there should be a minimum of one tutorial panel for every dialog panel.*
- All dialogs should have an overview topic. An overview is useful for orienting users who are unfamiliar with the dialog.

Tutorial Text:

- *Should be displayed using low intensity, and highlighting only important information.*
- *When explaining a dialog input field, display the field on the tutorial panel as it appears on the dialog panel (except that it should be a text or output field).*

Option Lines:

- Option Code

Should be numbers (preferred) or letters.

Should be displayed in a high intensity output field.

When a tutorial option corresponds to a selection menu option, the same option code should be used for both.

If the characters B, S, T, U, or I are used as selection options on a tutorial panel, the user will not be able to use these abbreviations to specify the tutorial commands BACK, SKIP, TOP, UP, and INDEX. The full tutorial commands will still function. However, if B, S, I, U, or I do need to be used as tutorial options then the user must be informed of the need to use the full tutorial command to execute that function.

- Option Description

Describes the specific area covered by the tutorial option.

Align the option codes and the option descriptions vertically.

Put the options with the highest probability of being chosen at the top of the list to reduce reading time. Match the order of the options to those found on the corresponding Selection Menu. If there is still no obvious pattern for ordering options, put them in alphabetical order.

4. Non-Display Panel

- a. The format of a non-display panel should be the same as that of an Information panel since the CONTROL NONDISPL services might be changed to a CONTROL DISPLAY LOCK during development of the application. The panel could then be used to help debug the application.

4. MESSAGES

4.1 GENERAL

1. All messages should be clear, concise, in plain language and in vocabulary that is familiar to the user. In a panel message if more information is needed provide it in the long message area.
2. Use standard punctuation.
3. Messages should have a consistent format. Place any user action first in the long message, followed by an explanation of the error or the consequences of the action.

4. Use positive statements wherever possible, e.g.

"Select an option from the list below"
instead of
"Option selected is invalid".

5. Use active rather than passive statements, e.g.

"Enter the name of the data set"
instead of
"Data set name must be entered"

6. *Do not use CLIST, EXEC or program output facilities to display messages. All messages should be displayed using the ISPF message facility.*

4.2 SPECIFIC

1. Always use the default message areas to display ISPF messages.
2. Messages should not overlay any panel fields (other than text fields used to display time-of-day, date or userid).

3. Each message should have both short and long text except. If a short message cannot be meaningful enough, a long message should be displayed immediately (without a short message). This is to avoid cryptic short messages. A HELP panel should be associated with each message.
4. Always use .ALARM=YES with error messages but error messages only. ".ALARM=YES" stops commands stacked on the COMMAND line.
5. Informative messages, when used consistently are very helpful and reassuring to users. They should be used wherever it is appropriate in the dialog to inform users of actions that are taking place.
6. The .HELP parameter should be used whenever there is a specific tutorial panel which relates to the message.
7. Position the cursor to the field in error when a message is displayed; this helps the user to find and correct the error.
8. Use the value from the field in error in the long message to increase the informational content (the same value should be displayed in the short message, if possible).
9. Use dialog variables in messages to make the message more meaningful. When possible, the dialog variable should be used in both the short and long message. For example, an input dialog could display the record number that was processed as part of the "record processed" short and long messages.
10. If processing efficiency is an important consideration, guidelines may be established concerning the size of message members. Message members will be accessed most efficiently if the total size of the member is kept under the blocksize of the data set. For example, if the blocksize of the library is 6160 (the recommended blocksize), then a message member of 77 lines ($77 \times 80 = 6160$) is the maximum that will fit in one block.
11. Short messages should be uppercase with not message numbers; long messages should be upper/lower case.

12. Whenever possible, long messages should include unique message numbers. This allows for easier communication and reference and reduces language dependency. Documentation, both on-line and in printed form should reference these message numbers.

13. Log messages should include, at the least, the same information that is displayed on the screen when information for an action is written to the log as well as being displayed on the panel. In this manner the user can easily identify the message in the log with the activities of the session. The following will accomplish this:

```
ISPEXEC SETMSG MSG(DTSM001)
ISPEXEC LOG    MSG(DTSM001)
```

14. The GETMSG service should not be used to generate messages for display on a panel or for placement in the log. Standard methods (such as the SETMSG or LOG services) should be used to directly create the required message. The GETMSG service should be used only when it is necessary to place the message information in a location other than a panel or the log. For example, the GETMSG service should be used when extracting information for placement in an application audit trail data set.

15. Messages should indicate exactly what values, actions or fields the messages are referring to. In panels, the additional use of cursor positioning is useful to enhance this.

5. TABLES

5.1 GENERAL

Tables should only be used for one-to-one mappings. ISPF tables were designed to provide a way to store, process and display a basic array structure. Attempts to use ISPF tables as a sophisticated database will result in extremely complex (and slow) table processing logic. Do not use a table row as a linked list.

5.2 SPECIFIC

1. Do not redefine the scroll PF keys to different functions. User confidence is undermined when the meanings of the PF keys change from one dialog to another.

In the event that you must redefine the scroll PF keys, encourage the use of the PFSHOW command so users can see how keys are defined.

2. If possible, do not use extension variables in a table. The presence of extension variables will add complexity to table processing which will make the dialog harder to maintain. Using extension variables in permanent tables will complicate the logic of every dialog which accesses the table.

Extension variables may be necessary for displaying error or informational messages within a table row.

3. *Display informative messages during user processing of the table. When the message is related to a function command, use the ISPF message facility. When the message is related to a particular row, display the message in a column of the table display which has been reserved for that purpose. For permanent tables, this can be best accomplished by using an extension variable to display the message. For temporary tables, define a message field in each row when the table is created, to avoid the use of extension variables.*
4. *The first input field in the table display should be the leftmost field. To be consistent with ISPF/PDF table displays, this field should start in column 1.*

5. *Emulate PDF processing of tables as closely as possible (for example, member selection lists in PDF option 3.1).*
6. The processing of the command field can be done either before or after processing the rows. To provide PDF consistency, however, processing the command field after processing the rows is suggested.
7. *When designing a dialog which displays tables, a decision must be made about the processing of the END and RETURN commands. When the user enters END or RETURN, there may be row modifications and/or application commands which have not yet been processed. If row processing is NOT done after an END or RETURN command, the user will be allowed to exit the table display without correcting errors which have previously been detected. If row and command field processing IS done before allowing the END or RETURN to take effect, a CANCEL (or equivalent) command should be provided to allow the user a chance to abort the operation.*

Using CANCEL is suggested since it is consistent with EDIT.

8. Any variable which is specified in a MODEL line but which is not part of the actual table (as defined using the TBCREATE service) must be cleared before each re-display of the table. It is recommended that this be done in the INIT section as opposed to the REINIT section or in a CLIST, EXEC or program. This prevents propagation of the value of the variable in one row of the table through each row of the displayed table. (This type of variable is often used to display a SELECT CODE field for line commands).
9. The enqueue mechanism provided by ISPF is intended to prevent concurrent updating of the same table by multiple users. The mechanism is based on the assumption that all users updating a given table will have the same first library definition for ISPTLIB. Dialog developers should be aware of this when designing dialogs which will be used by concurrent users to update a table. A recommended approach to alleviate this problem is to use the LIBRARY parameter on the TBOPEN, TBCLOSE and TBCREATE service. A unique ddname can be allocated to the input and output tables so that conflicts don't arise with ISPTLIB concatenation and ISPTABL. Using the LIBDEF service provides another mechanism by which this can be done to reduce confusion and conflicts.
10. Make certain that you sort the table in a sequence that will be logical to the user before displaying it. If possible sort and store the table as it will be referenced.

11. When a table is being accessed for the READ only, make sure that the NOWRITE option is specified on the TPOPEN service. This will insure that the table can be accessed concurrently for input only by multiple users.

6. FILE TAILORING

6.1 GENERAL

Each installation should examine the use of file tailoring in the development of their dialogs for on-line applications; misuse or inappropriate use of this facility can significantly degrade performance.

6.2 SPECIFIC

1. Skeletons should be commented, just as programs, CLISTs, EXECs and panels should be commented.
2. Use temporary ISPF output files wherever possible as opposed to handling permanent allocations.
3. When referencing a temporary output file use the ddname supplied in the ZTEMPN system variable as opposed to the dsname in ZTEMPF. This will provide more flexibility especially when the temporary file is allocated to VIO.
4. Avoid using)IM. It complicates debugging; use FTINCLs. Also, always check the return code from the the FTINCL calls.
5. The complexity of skeletons should be limited, just as the complexity of programs are limited for the purposes of understanding and maintenance.
6. Tabbing, blank lines (")BLANK") and alignment should be used to keep generated output, as well as skeleton source, readable.
7. The ")CM" statement should be used to document:
 - a. Which dialog components use the skeleton.
 - b. How the output file is used.
 - c. Any unique features of the skeleton or environment.

8. If EDIT or BROWSE is going to be used on the output dataset then the file tailored output dataset cannot be VIO. If the output is to be BROWSEd or EDITted then make sure the ISPCTLx dataset(s) are allocated to real DASD. In all cases check the ZTEMPF and ZTEMPN variables to make sure that a real dataset is available for EDIT or BROWSE before attempting the operation.

7. MODELS

7.1 SPECIFIC

1. Dialog developers should be strongly discouraged from making their own copy of the model menus (ISREMCMD, ISREMPNL, etc.). If they do create their own copies of these menus, they will not have access to changes to these menus made in new versions of ISPF/PDF or changes made by the installation. Each installation should provide a mechanism for users and/or groups to add their own models. This can be done by adding U (User) and G (Group) options to the model menus accessed by everyone in the installation. Following are two approaches to adding models. The first method is recommended because it does not complicate the MODEL command used to imbed the IBM-supplied models.

- a. Method 1:

Place User and Group options on the Class menu and User, Group, and Installation options on the model options menu for each class.

Advantages - Standard ISPF models are accessible at the first level of the hierarchy (e.g. the TBCREATE model can be imbedded in a CLIST using the command 'MODEL TBCREATE').

Disadvantages - User models are placed one level lower in the hierarchy than the IBM models.

- b. Method 2:

Insert a menu between the Class menu and each of the option menus. On the inserted menu, place options for ISPF services and Installation, Group and User options.

Advantages - All model options are at the same level in the hierarchy; the model hierarchy can be organized more logically.

Disadvantages - The command used to imbed a model into a data set (without displaying the model menus) requires an additional qualifier to identify which section the model belongs to (ISPF, Installation, Group or User). For example: the command required to imbed the TBCREATE model would be MODEL 1.TBCREATE if IBM-supplied ISPF models were option 1. Also, if users are not familiar with which models are IBM-supplied ISPF and which are 'home-grown', they may have

to search one section before discovering that it is contained in the other. This search may be frustrating because an 'END' command on any model menu terminates the entire MODEL command.

2. Within the skeleton member, NOTEs should be included to explain any options or parameters available to the user or areas which need to be customized by the user. The text within NOTE lines should be in mixed upper and lower case to improve readability.
3. All models which are longer than ten lines should begin with a NOTE that explains the use of the model.

8. USER CUSTOMIZATION OF SYSTEM-WIDE DIALOG COMPONENTS

8.1 GENERAL

1. Dialog developers should be strongly discouraged from overriding system-wide dialog components with their customized version of these components. When ISPF users are allowed to override system-wide components, two problems occur:
 - a. System-wide components are inadvertently overridden by users, resulting in dialog errors which can be difficult to diagnose.
 - b. When modifications are made to dialog components in the system-wide libraries, the user-customized components prevent the user from seeing new or enhanced features. The most common examples of this are user-customized Primary Option menus. A user overriding the Primary Option menu is unaware of changes or additions which have been made in the menu in the system-wide libraries.
2. A program which front-ends the ISPF command processor can be used to enforce a concatenation sequence which your installation has defined as standard.

9. CONCATENATION SEQUENCES FOR ISPF LIBRARIES

A standard concatenation sequence for ISPF libraries should be established by each installation. For purposes of this discussion, we have identified four categories of libraries:

1. IBM-supplied ISPF and/or ISPF/PDF libraries
2. System wide applications (Installation-developed and Vendor)
3. Group applications (Installation-developed and Vendor)
4. User applications (Installation-developed and Vendor).

Two methods exist for allocating and concatenating ISPF libraries:

1. Pre-allocation of the ISPF files via:
 - a. DDNAMES in the TSO logon proc.
 - b. TSO ALLOCATE commands.
 - c. LINK and ACCESS the ISPF minidisks.
 - d. CMS FILEDEF commands.
2. Using the ISPF LIBDEF facility.

Each installation must determine how they are going concatenate all the datasets used by ISPF for each user. The following can serve as guidelines:

1. IBM supplied libraries for ISPF and PDF should be pre-allocated for all users.
2. Files supplied by vendors or which contain dialog elements written by an installation which are used by all or a large group of users should be pre-allocated to SPF.
3. Vendor or installation written dialogs which need to be limited to a subset of users can be either pre-allocated just by the users who need to execute them or the ISPF LIBDEF facility can be used

CHAPTER: 9 - Concatenation Sequences for ISPF Libraries

to dynamically allocate the the appropriate files prior to selecting the dialog. Reasons for limited dialog access are:

- a. Only one group needs access to a particular vendor-supplied application.
 - b. One group has purchased its own vendor-supplied application and does not want anyone else to have access to it.
 - c. One group has purchased a proprietary application and is restricted by the licensing agreement to be the only users of the application.
4. If an installation has many ISPF files consider using LIBDEF to allocate libraries for infrequently used applications. Pre-allocating many ISPF libraries increases ISPF initialization overhead since it must open each one every time ISPF is invoked.
 5. If the total number of ISPF files is not great, pre-allocating may simplify the maintenance of the system.
 6. If an installation modifies IBM dialog components, do so to a file other than the one distributed by IBM. This will insure that your changes are not removed by routine IBM system service (IBM's panel, message, skeleton and table files are all SMP maintained in MVS; maintenance to VM mass replaces distributed dialog elements).
 7. Consider using a similar technique for any non-IBM supplied vendor package.

Each installation is encouraged to protect any proprietary or company sensitive dialogs with external security. Do not rely on the ISPF library concatenations to limit access to dialogs.

An installation may wish to consider the LIBDEF service to break up elements for different dialogs.

Advantages of using LIBDEF:

1. Allocation of files can be deferred until the dialog is invoked. This reduces ISPF initialization overhead.

CHAPTER: 9 - Concatenation Sequences for ISPF Libraries

2. ISPF initialization overhead is reduced since some ISPF files do not have to be allocated (they will be allocated later by LIBDEF).
3. Dialog elements can be maintained in their own libraries. This may simplify maintenance of the dialogs.
4. The libraries are only allocated by the individuals executing the dialogs, not all ISPF users.

Disadvantages of using LIBDEF:

1. Additional overhead when selecting a dialog which uses LIBDEF services.
2. Constant allocation and unallocation of files when entering and exiting a dialog which uses LIBDEF.

CHAPTER: 9 - Concatenation Sequences for ISPF Libraries

10. CLISTS

10.1 GENERAL

CLISTS can be used in dialogs to speed development and ease maintenance. It is, however, an interpretive language and its use must be weighed against performance concerns. If a requirement of a dialog is faster execution than a non-interpretive language should be used (COBOL, FORTRAN, ASSEMBLER, PL/1, etc.)

A developer may elect to write a dialog initially using clists and later re-write that portion of the dialog in a more efficient language.

10.2 SPECIFIC

1. *CLISTS should be invoked by prefixing a percent sign (%) to the CLIST name:*

%clistname

If the percent sign is not used to invoke a CLIST, line mode display is initiated and the screen is cleared.

2. Global clist variables should not be used by CLISTS which are part of a dialog. ISPF variable services should be used.
3. *ATTN exits should not be used in ISPF CLISTS; their use will cause unpredictable results.*
4. *The TERMIN command procedure statement is not supported under ISPF and must not be used.*
5. ISPF file-tailoring and the PDF edit-macro facilities should be used to format output dataset (JCL, input or output files, control cards, etc.). The use of TSO EDIT in a CLIST to perform such functions should be discontinued in favor of using these standard facilities, which are specifically designed for tailoring output data.

6. An installation may consider using the LIBDEF service instead of TSO ALLOCATE commands. This may simplify migrating from CLISTs to other programming languages or to VM (REXX or EXECs).

APPENDIX: A - Contributors

Appendix A

CONTRIBUTORS

Fred Bauer

**Jeffrey Multach
Ameritech Services**

**Don Paul Beletsky
Computer Associates**

**Joe Nader
Carolina Power and Light Company**

**Sharon L. Brian
Texas Instruments Inc.**

**Neil H. Nix
Lockheed-Georgia Co.**

**Art Burchfield
IBM**

**Ross Pavlac
First National Bank of Chicago**

**Elie Cassorla
IBM Watson Research Center**

**Scott Peterson
Southern California Edison**

**Shelley Conrey
IBM**

**Rita Schaaf
United Airlines**

**Dale L Cybela
Deluxe Data Systems**

**Bert Shankman
Federal Reserve Board**

**Alex Drescher
Rohm and Haas Company**

**Jerry Smith
McDonnell Douglas**

**Chris Gerken
EXXON USA**

**Cindy Stegemeier
Pennzoil Corp.**

**Joe Jankowski
New York Life Insurance**

**Gus Trujillo
Continental Bank**

**Brian Johnson
The City of Winnipeg**

**Brian J. Vohs
Exxon Co. USA**

**Al Johanson
Kraft, Inc.**

**Keith Warltier
IBM ISG**

**James W. Lorenger
Lockheed - California Company**

**Kent Williams
Santa Fe Railway**

**John McConnell
AT&T Information Systems**

**Sam Woodruff
Northwestern Mutual Life**

**Harvey Miller
Fireman's Fund Insurance / CPD**

**Bob Zimmerman
CNA Insurance**

APPENDIX: A - Contributors

Bibliography

BIBLIOGRAPHY

- Bauer, Fred. "ISPF Experience at Moore-McCormack Lines". GUIDE 54 Proceedings, May, 1982, pp. 749-754.
- Biafore, John. "Enhancing SPF by Adding Menus". GUIDE 50 Proceedings, May, 1980, pp. 1123-1146.
- Bork, Alfred. Learning with Computers. Digital Press, Bedford, Mass., 1981, 286 pp.
- Brooks, Jr., Frederick P. The Mythical Man-Month. Addison-Wesley, Reading, Mass., 1975.
- Brown, P.J. "Error Messages; the Neglected Area of the Man/Machine Interface?". Communications of the ACM, April, 1983, pp. 246-249.
- Clark, I.A. "Software Simulation As a Tool For Usable Product Design". IBM Systems Journal, No. 3, 1981, pp. 272-293.
- Currier, Richard L. "The Secret of 'User Friendliness'". ICP Interface, Jan-Feb 1983, pp. 15-16.
- Dean, M. "How a Computer Should Talk to People". IBM Systems Journal, No. 4, 1982, pp. 424-453.
- DeVoe, Cort. "SPF - A Tutorial". GUIDE 53 Proceedings, November 8-13, 1981, pp. 517-536.
- Fitzgerald, Kathy. "Flawed Screen Formats Corrected by Designers". Software News, 12/7/81, pg. 40.
- Fried, Louis. "Nine Principles for Ergonomic Software". Datamation, November, 1982, pp. 163-166.
- Galitz, Wilbert. "Human Engineering in Screen Design". Journal of Systems Management, May, 1983, pp. 6-11.
- Gilb, Tom, and Weinberg, Gerald M. Humanized Input Techniques for Reliable Keyed Input. Winthrop Publishers, Cambridge, Massachusetts, 1977, 283 pp.
- Gilb, Tom. "Laws of Unreliability". Datamation, March, 1975, pp. 81-85.
- Gutz, Steve, Wasserman, Anthony I., and Spier, Michael J. "Personal Development Systems for the Professional Programmer". Computer, April, 1981, pp. 45-53.

Bibliography

- Hansen, W. J. "User Engineering Principles for Interactive Systems". Proceedings of the Fall Joint Computer Conference, 39, AFIPS Press, Montvale, New Jersey, 1971, pp. 523-532.
- Helander, G.A. "Improving System Usability for Business Professionals". IBM Systems Journal, No. 3, 1981, pp. 294-305.
- Hoernes, Gerry. "Using the System Productivity Facility (SPF) to Provide Function to the End User". GUIDE 52 Proceedings, May 10-14, 1981, pp. 445-465.
- Hoernes, Gerry. "SPF Control and Flow". SHARE 58 Proceedings, March, 1982, pp. 1541-1550.
- IBM Corporation. IBM 3270 Screen Design Student Text. IBM Corp., 1974, 69 pp. IBM #SR20-4441.
- Johnson, Bengt, et al. "The Human-Computer Interface in Commercial Systems Investigations of Dialog Design Factors and Usability with Alphanumeric Display Terminals". 1981, NTIS document PB81-191785.
- Joslin, Phil H. "System Productivity Facility". IBM Systems Journal (reprint no. G321-5155), No. 4, 1981, pp. 388-406.
- Kidder, Tracy. The Soul of a New Machine. Atlantic-Little, Brown, 1981, 293 pp.
- Kostyun, Jan. "IBM ISPF dialog manager; More than meets the eye". Computerworld, September 24, 1984, p. 69.
- Kreiser, Frederick Jr. "Using the 'Structured Programming Facility' of TSO". GUIDE Proceedings, November, 1975, pp. 746-750 and 1221-1225.
- Martin, James. Design of Man-Computer Dialogues. Prentice-Hall, Englewood Cliffs, New Jersey, 1973, 559 pp.
- Maurer, M. E. "Full-screen testing of interactive applications". IBM Systems Journal (reprint no. G321-5194), Vol. 22, No. 3, 1983, pp. 246-261.
- Morland, D. Verne. "Friendliness". Datamation February, 1982, pg. 224.
- Morland, D. Verne. "Human Factors Guidelines for Terminal Interface Design". Communications of the ACM, July, 1983, pp. 484-494.
- Norman, Donald L. "Design Rules Based on Analyses of Human Error". Communications of the ACM, April, 1983, pp. 254-258.

Bibliography

- Pavlac, Ross. "SPF User Experience". GUIDE 53 Proceedings, November 8-13, 1981, pp. 662-670.
- Pavlac, Ross. "ISPF Hazards and Pitfalls". GUIDE 56 Proceedings, May, 1983, pp. 800-810.
- Pavlac, Ross. "ISPF Table Handling; A Report from the Combat Zone". This paper was presented at GUIDE 57, November 6-11, 1983. Contact the GUIDE ISPF Project Manager for a copy (see address on Reader Comment Card).
- Pirsig, R. Zen and the Art of Motorcycle Maintenance. Morrow and Co., 1974.
- Relles, Nathan, and Price, Lynne A. "A User Interface for Online Assistance". Proceedings of the 5th International Conference on Software Engineering IEEE, 1981, pp. 400-408.
- Roberts, Teresa L., and Moran, Thomas P. "The Evaluation of Text Editors; Methodology and Empirical Results". Communications of the ACM, April, 1983, pp. 265-283.
- Schneiderman, Ben. Software Psychology
Human Factors in Computer and Information Systems. Winthrop Publishers, Cambridge, Massachusetts, 1980, 320 pp.
- Schneiderman, Ben. "Designing Computer System Messages". Communications of the ACM, September, 1982, pp. 610-611.
- Snyders, Jan. "User Manuals That Make Sense". Computer Decisions, April, 1981, pp. 127-186.
- Stegemeier, Cindy. "ISPF Dialog Development Methods". GUIDE 58 Proceedings, March 1984, pp. 837-847.
- Sterling, Theodor D. "Guidelines for Humanizing Computerized Information Systems; A Report from Stanley House". Communications of the ACM, November, 1974, pp. 609-613.
- Wasserman, Anthony I. "The Design of 'Idiot-proof' Interactive Programs". Proceedings of the National Computer Conference, 42, AFIPS Press, Montvale, New Jersey, 1973, pp. M34-38.
- Wasserman, Anthony I. "User Software Engineering and the Design of Interactive Systems". Proceedings of the 5th International Conference on Software Engineering, IEEE, 1981, pp. 387-393.
- Weinberg, Gerald M. The Psychology of Computer Programming, Van Nostrand Reinhold, New York, 1971.

Bibliography

Weiss, Edmond H. "Fair-Weather Friendliness". Datamation, November, 1983, pp. 327-328.

6

&ZUP 25, 29

%

% sign
See CLISTS

A

ATTN exits
See CLISTS

C

CLISTS 49
 % sign, invoking a CLIST with 49
 ATTN exits 49
 file tailoring vs. TSO EDIT 49
 global variables 49
 line mode display 49
 TERMIN command 49
command line
 See panels
confirmation panels
 See panels
customization
 See ISPF libraries

D

default attribute character
 See panel, attribute character
directive line
 See panels
dynamic areas
 panels 19

E

END command
 See tables
extended data stream
 See panels

INDEX

F

- file tailoring
 - batch vs. on-line execution 39
 - skeletons 39
 - uses of 39
- function line
 - See panels

G

- global variables
 - See CLISTs
- graphic areas
 - See panels

H

- HELP
 - See panels

I

- information panels
 - See panels
- instruction line
 - See panels
- intensity
 - See panels
- ISPF libraries
 - concatenation sequence
 - enforcing installation standard 43
 - customization of system-wide ISPF components 43
- ISPTLIB
 - See tables

L

- LIBDEF 45, 46
- library concatenation sequence 45
- line mode display
 - CLISTs 49
- long message line
 - See panels

M

- member names
 - See naming standards
- menus
 - See selection panels
- messages 31
 - alarm 32
 - default message area 31
 - designing 31
 - format 31
 - help panels 32
 - informative 32
 - message data set blocksize and member size 32
 - optimizing message retrieval 32
 - output facilities 31
 - overlying panel fields 31
 - positioning the cursor 32
 - punctuation 31
 - vocabulary 31
- messages in table displays
 - See tables
- models 41
 - adding models and classes 41
 - model hierarchy 41
 - two possible methods 41
 - NOTE lines 42

N

- naming standards 3
 - examples 6
 - identifiers
 - dialog id 4
 - group id 4
 - system id 4
 - type 5
 - member names 4
 - IBM standard 7
 - messages 6
 - variables 3
- non-display panels
 - See panels

O

- option line
 - See panels

P

INDEX

- panel design
 - See panels, design considerations
- panels 9
 - appearance 9
 - command line 10
 - confirmation 24
 - description field, for input fields 13
 - design considerations 18
 - directive line 16
 - menus 17
 - non-scrollable panels 16
 - scrollable panels 17
 - tutorials 17
 - dynamic 26
 - dynamic areas 19
 - extended data stream 18
 - fields 12
 - function line 11
 - general layout 9
 - graphic areas 26
 - HELP 18
 - information, on input fields 14
 - informational 23
 - input fields 13, 19
 - input fields, repeating 14
 - instruction line 11
 - intensity 18, 19
 - long message line 10
 - multi-column data entry 22
 - non-display 30
 - non-scrollable 21
 - padding character 13, 15
 - scrollable 11, 26
 - sections 19
 - select field 15, 19
 - selection panels 27
 - option line 27
 - standard arrow symbol 13
 - table display 26
 - text fields 15
 - text lines 16
 - confirmation 16
 - informational 16
 - tutorial 16
 - title line 9
 - TRANS 19
 - TRUNC 19
 - tutorial 11
 - tutorial menu 29
 - tutorial text 25, 29
 - types 12

INDEX

PFSHOW 18
pre-processed panels 19

R

repeating input fields
 See panels, input fields
RETURN command
 See tables

S

screens
 See panels
scrollable panels
 See panels
selection panels
 See panels
singly occurring
 See panels, input fields
SMP (System Modification Program)
 member name requirements 3

T

table display panels
 See panel
tables 35
 command field processing 36
 extension variables 35
 ISPTLIB enqueue 36
 messages 35
 MODEL line 36
 processing the END or RETURN command 36
 processing user-modified rows 36
 providing a CANCEL command 36
 scroll PF keys 35
 select code field' 36
 TBCREATE service 36
 values propagated through table rows' 36
TBCREATE service
 See tables
TERMIN command
 See CLISTS
title line
 See panels
TRANS 19
TRUNC 19

INDEX

TSO EDIT

See CLISTs

tutorial menu

See panels

tutorial text panel

See panels

V

variables

See naming standards