

""

ATTN: CHARLIE GIBBS

01111

CAV208M45541 UP 8802 R2

UAS

SPERRY UNIVAC
1 - 1818 CORNWALL STREET
VANCOUVER B C

V6J 1C7

**PUBLICATIONS
REVISION**

Operating System/3 (OS/3)

Screen Format Services

Concepts and Facilities

UP-8802 Rev. 2

This Library Memo announces the release and availability of "SPERRY UNIVAC® Operating System/3 (OS/3) Screen Format Services Concepts and Facilities", UP-8802 Rev. 2

This revision reflects major changes to screen format services for the 8.0 release. Among these changes are:

- Dividing the home screen into two separate screens: the home screen and the characteristics screen
- Allowing the use of conditional indicators
- Inclusion of six optional screens that affect the run-time characteristics of a screen format
- Inclusion of four more dialog screens that affect the run-time characteristics of individual fields
- A step-by-step explanation of how to create a simple screen format
- A step-by-step explanation of how to create a format using indicators
- Extending the LIST function to include three more informational screens
- Job control and language enhancements, including COBOL support for multivolume workstations
- Additional support for nondomestic alphabets available to the user

All other changes are corrections or expanded descriptions applicable to features present in screen format services prior to the 8.0 release.

Destruction Notice: If you are going to OS/3 release 8.0, use this revision and destroy all previous copies. If you are not going to OS/3 release 8.0, retain the copy you are now using and store this revision for future use.

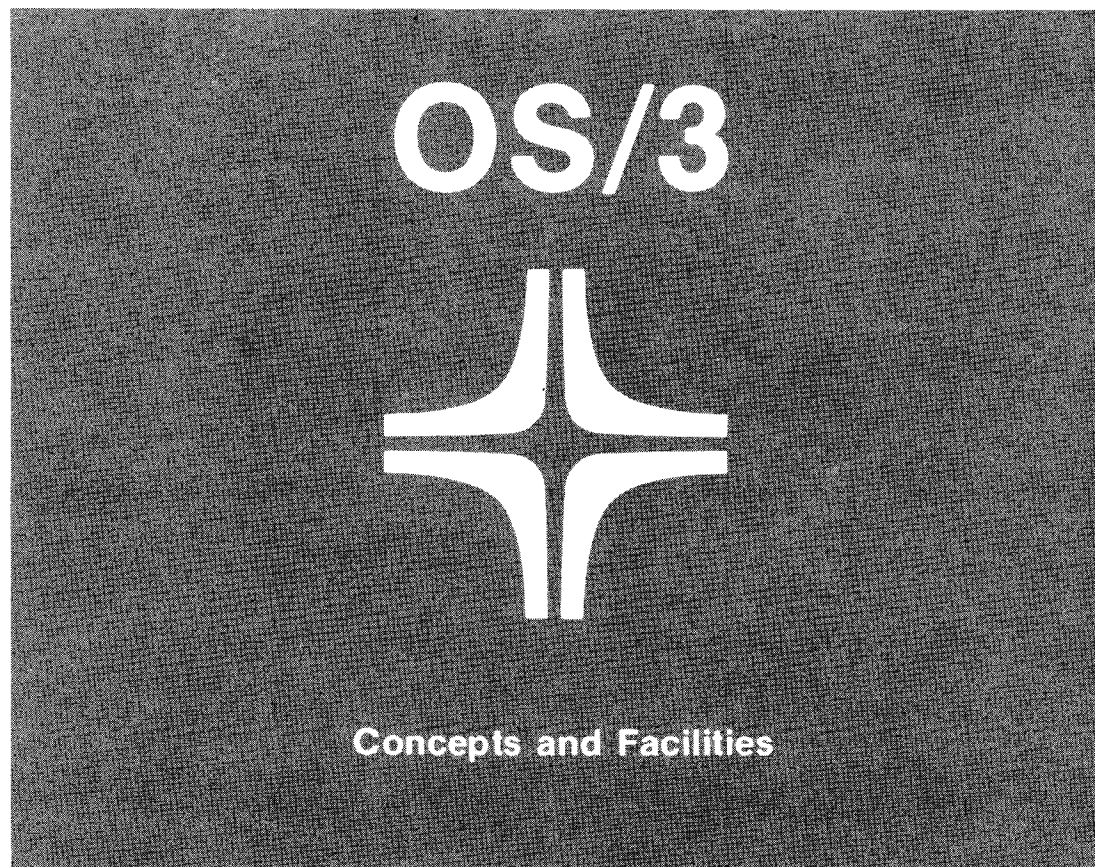
Copies of UP-8802 Rev. 1 will be available for 6 months after the release of 8.0. Should you need additional copies of this edition, you should order them within 90 days of the release of 8.0. When ordering the previous edition of a manual, be sure to identify the exact revision and update packages desired and indicate that they are needed to support an earlier release.

Additional copies may be ordered by your local Sperry Univac representative.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ and MZ	Mailing Lists A00, A05, B00, B01, 18, 18U, 19, 19U, 20, 20U, 21, 21U, 28U, 29U, 75, 75U, 76, and 76U (Cover and 189 pages)	Library Memo for UP-8802 Rev. 2 RELEASE DATE: September, 1982



Screen Format Services



This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please consult the appropriate release documentation or contact your local Sperry Univac representative.

Sperry Univac reserves the right to modify or revise the content of this document. No contractual obligation by Sperry Univac regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry Univac.

Sperry Univac is a division of the Sperry Corporation.

FASTRAND, SPERRY UNIVAC, UNISCOPE, UNISERVO, and UNIVAC are registered trademarks of the Sperry Corporation. ESCORT, MAPPER, PAGEWRITER, PIXIE, and UNIS are additional trademarks of the Sperry Corporation.

This document was prepared by Systems Publications using the SPERRY UNIVAC UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.

PAGE STATUS SUMMARY

ISSUE: UP-8802 Rev. 2
RELEASE LEVEL: 8.0 Forward

Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level
Cover/Disclaimer								
PSS	1							
Preface	1 thru 3							
Contents	1 thru 6							
1	1 thru 13							
2	1 thru 7							
3	1 thru 44							
4	1 thru 20 20a 21 thru 26							
5	1 thru 6							
6	1 thru 14							
7	1 thru 26 26a 27 thru 37							
Appendix A	1 thru 7							
Appendix B	1							
Appendix C	1							
Appendix D	1							
Appendix E	1							
Index	1 thru 17							
User Comment Sheet								

All the technical changes are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.



Preface

This manual is one of a series designed to instruct and guide the programmer in the use of the SPERRY UNIVAC Operating System/3 (OS/3). It specifically describes screen format services (SFS), the interactive component of OS/3 that provides an easy and effective method for inputting and outputting data.

The intended audience is the programmer with a basic knowledge of data processing, but with little programming experience.

This manual is divided as follows:

■ Section 1. Introduction

- Introduces you to screen formats as an input and output facility for programs.
- Describes the capabilities that SFS provides, via the screen format generator (SFG) and the screen format coordinator (SFC), for creating and using screen formats.

■ Section 2. Type Attributes and Editing Rules

Discusses a variation of COBOL editing rules to be observed when creating your own formats.

■ Section 3. Using the Screen Format Generator (SFG)

- Explains how to initiate the SFG session.
- Describes the home screen.
- Discusses the use of the HELP function.
- Describes the characteristics screen and the results produced by making various specifications on it.
- Explains the various dialog and optional screens.

- Section 4. Creating Screen Formats

Provides detailed instructions for creating your own screen formats.

- Section 5. Modifying Formats

Explains various methods for changing existing formats.

- Section 6. Additional Functions

Describes the functions that enable you to display and print existing formats and format information. Explains the functions that delete formats and terminate the SFG session.

- Section 7. Associating Formats with a Program: Using the Screen Format Coordinator (SFC)

- Explains how screen formats become functional with a program through the SFC.
- Discusses job control requirements.
- Introduces programming requirements related to RPG II, COBOL, FORTRAN IV, assembler language, ESCORT, and IMS interfaces.

- Appendix A. Job Control Considerations

Provides more information about the USE SFS job control statement. Describes the job control language statements necessary to use more than one workstation (applicable to RPG II programmers only).

- Appendix B. Help Screens

Shows the various help screens you can request during creation of a screen format.

- Appendix C. Function Keys

Describes the various function keys you can use with screen format services.

The following publications are referenced in this manual and are either useful or necessary to the programmer working with screen format services.

System 80

- Consolidated data management macro language user guide/programmer reference, UP-8826
- Assembler user guide, UP-8913
- FORTRAN IV programmer reference, UP-8814

- Interactive services commands and facilities user guide/programmer reference, UP-8845
- RPG II user guide, UP-8067
- 1974 ANSI COBOL programmer reference, UP-8613
- Job control user guide, UP-8065
- General editor user guide/programmer reference, UP-8828

Series 90 Systems

- Assembler user guide, UP-8061
- FORTRAN IV programmer reference, UP-8474
- Interactive services commands and facilities user guide/programmer reference, UP-8845
- RPG II user guide, UP-8067
- 1974 ANSI COBOL programmer reference, UP-8613
- Job control user guide, UP-8065
- General editor user guide/programmer reference, UP-8828



Contents

PAGE STATUS SUMMARY

PREFACE

CONTENTS

1. INTRODUCTION

1.1.	PROVIDING INFORMATION VIA FORMS	1-1
1.2.	PROVIDING INFORMATION VIA SCREEN FORMAT SERVICES	1-1
1.3.	SCREEN FORMAT GENERATOR	1-6
1.3.1.	Functions of the SFG	1-6
1.3.2.	Screen Format Components	1-8
1.3.2.1.	Display Constants	1-9
1.3.2.2.	Variable Data Fields	1-9
1.3.2.3.	Protected and Unprotected Parts - Low-Intensity Display	1-11
1.4.	SCREEN FORMAT COORDINATOR (SFC)	1-12

2. TYPE ATTRIBUTES AND EDITING RULES

2.1.	CONSIDERATIONS FOR VARIABLE DATA FIELDS	2-1
2.2.	TYPE ATTRIBUTES	2-1
2.3.	EDITING	2-2
2.3.1.	Editing Alphanumeric and Alphanumeric Fields	2-2
2.3.2.	Editing Numeric Fields	2-3
2.4.	RESULTS OF EDITING FORMAT FIELDS	2-4

3. USING THE SCREEN FORMAT GENERATOR (SFG)

3.1.	ACTIVATING THE SFG	3-1
3.2.	THE HOME SCREEN	3-1
3.2.1.	Functions Provided by the SFG	3-2
3.2.2.	New Format Name and Library	3-4
3.2.3.	Old Format Name and Library	3-4
3.3.	THE CHARACTERISTICS SCREEN	3-5
3.3.1.	Lowercase Translation (Line 2)	3-6
3.3.2.	Alphabet (Line 3)	3-6
3.3.3.	Original and Overlay Formats (Line 3)	3-6
3.3.4.	Erasing and Unlocking Options (Lines 4 and 5)	3-7
3.3.5.	Error Retry Counts (Line 6)	3-8
3.3.6.	Special Editing Characters (Line 6)	3-9
3.3.7.	Special Display Control (Line 7)	3-9
3.3.8.	Error Message Field (Line 8)	3-10
3.3.9.	Display Retention (Line 9)	3-10
3.3.10.	Function Keys (Line 10)	3-10
3.3.11.	Nondisplayed Field (Line 11)	3-11
3.4.	OPTIONAL SCREENS	3-11
3.4.1.	Conditional Indicators	3-12
3.4.2.	Conditional Erase/Replenish Screen	3-13
3.4.3.	Edit Screen	3-16
3.4.4.	Special Display Screen	3-17
3.4.5.	Error Message Screen	3-18
3.4.6.	Display Retention Screen	3-21
3.4.7.	Function/Command Key Screen	3-22
3.4.8.	Nondisplay Screen	3-23
3.5.	DIALOG SCREENS	3-24
3.5.1.	Dialog Screen 1 (General Field Characteristics)	3-26
3.5.2.	Dialog Screen 2 (Input Field Response, Default, and Replenishing)	3-31
3.5.3.	Dialog Screen 3 (Conditional Field Display)	3-33
3.5.4.	Dialog Screen 4 (Special Field Display)	3-34
3.5.5.	Dialog Screen 5 (Conditional Field Retention)	3-36
3.5.6.	Dialog Screen 6 (Conditional Field Protection)	3-37
3.5.7.	Dialog Screen 7 (Input Field Range Checking)	3-38
3.5.8.	Default Values for Dialog Screens	3-40
3.6.	COMPLETING SFG SCREENS - CURSOR POSITIONING AND OVERWRITING	3-41
3.7.	ERROR AND WARNING MESSAGES	3-43
3.8.	THE HELP FUNCTION	3-44

4. CREATING SCREEN FORMATS

4.1.	CREATING A SIMPLE SCREEN FORMAT	4-1
4.1.1.	Pass 1 – Providing the Layout	4-3
4.1.2.	Pass 2 – Specifying Type Attributes and Editing	4-5
4.1.3.	Pass 3 – Specifying I/O Directions and Initiating Dialogs	4-7
4.2.	CREATING MORE COMPLEX FORMATS (USING OPTIONAL AND DIALOG SCREENS)	4-10
4.2.1.	A Sample Format	4-10
4.2.1.1.	The Sample Home Screen and Characteristics Screen	4-12
4.2.1.2.	The Conditional Erase/Replenish Screen for MYFORMAT	4-13
4.2.1.3.	The Error Message Screen for MYFORMAT	4-14
4.2.1.4.	Completing Pass 1 for MYFORMAT	4-14
4.2.1.5.	Completing Pass 2 for MYFORMAT	4-15
4.2.1.6.	Completing Pass 3 for MYFORMAT	4-16
4.2.1.7.	Completing Dialog Screens for MYFORMAT	4-17
4.3.	OVERLAY FORMATS	4-20a
4.3.1.	The Need for an Overlay Format	4-20a
4.3.2.	Creation of an Overlay Format	4-22
4.4.	CREATING A FORMAT WITH DUPLICATE LINES	4-24

5. MODIFYING FORMATS

5.1.	CREATE-FROM AND MODIFY FUNCTIONS	5-1
5.1.1.	Modify Screen	5-2
5.1.1.1.	Modify Screen – Option 1 (CHANGE TEMPLATE)	5-3
5.1.1.2.	Modify Screen – Option 2 (CHANGE TYPE)	5-3
5.1.1.3.	Modify Screen – Option 3 (CHANGE I/O)	5-4
5.1.1.4.	Modify Screen – Option 7 (CHANGE TEMPLATE ONLY – NO INTERNAL CHANGES)	5-4
5.2.	MODIFICATION SUMMARY	5-6

6. ADDITIONAL FUNCTIONS

6.1.	DISPLAY FUNCTIONS	6-1
6.1.1.	SHOW Function	6-1
6.1.2.	LIST Function	6-1
6.1.2.1.	Summary Screen	6-3
6.1.2.2.	Special Editing Display	6-4
6.1.2.3.	Display Control and Error Message Display	6-5
6.1.2.4.	Function/Command Key Display	6-5
6.1.2.5.	The Format Display	6-6
6.1.2.6.	List Screen	6-6
6.1.2.7.	Sublist Screen	6-9
6.1.2.8.	Conditional Values Screen	6-11
6.1.2.9.	Input Record Screen	6-12
6.1.2.10.	Output Record Screen	6-12
6.1.3.	SPOOL Function	6-13
6.2.	DELETE AND TERMINATE FUNCTIONS	6-13

7.	ASSOCIATING FORMATS WITH A PROGRAM: USING THE SCREEN FORMAT COORDINATOR (SFC)	
7.1.	PROGRAMMER RESPONSIBILITIES	7-1
7.2.	JOB CONTROL	7-1
7.3.	PROGRAM CONSIDERATIONS	7-5
7.3.1.	RPG II	7-5
7.3.2.	COBOL	7-9
7.3.2.1.	Major Coding Considerations	7-10
7.3.2.2.	Using Indicators	7-18
7.3.2.3.	Additional Coding Considerations for Function Keys and Multivolume Workstations	7-21
7.3.2.4.	Considerations for Workstations with Interprogram Communication	7-26
7.3.3.	FORTRAN	7-26
7.3.4.	BAL	7-27
7.3.4.1.	Open a File (OPEN)	7-27
7.3.4.2.	Close a File (CLOSE)	7-28
7.3.4.3.	Retrieve a Record (DMINP)	7-29
7.3.4.4.	Select a Screen Format (DMSEL,SCREEN)	7-31
7.3.4.5.	Output a Record (DMOUT)	7-33
7.3.5.	ESCORT	7-34
7.3.6.	IMS Programming Considerations	7-35
7.4.	CONSIDERATIONS FOR THE WORKSTATION OPERATOR	7-37

APPENDIXES

A. JOB CONTROL CONSIDERATIONS

A.1.	THE USE SFS STATEMENT	A-1
A.2.	JOB CONTROL FOR MULTIPLE WORKSTATIONS	A-6

B. HELP SCREENS

C. FUNCTION KEYS

D. ALPHABET TABLE

E. SFS SUPPORT OF DISPLAY ATTRIBUTES

INDEX

USER COMMENT SHEET

FIGURES

1-1.	Screen Format Used to Supply Input Data	1-2
1-2.	Screen Format Used to Display Output Data	1-3
1-3.	Screen Format Used for Input and Output	1-3
1-4.	The Function of the SFG and SFC	1-5
3-1.	The Home Screen	3-2
3-2.	The Characteristics Screen	3-5
3-3.	Conditional Erase/Replenish Screen	3-14
3-4.	Edit Screen	3-16
3-5.	Special Display Screen	3-17
3-6.	Error Message Screen	3-19
3-7.	Display Retention Screen	3-21
3-8.	Function/Command Key Screen	3-22
3-9.	Nondisplay Screen	3-23
3-10.	Dialog Screen 1	3-26
3-11.	Dialog Screen 2	3-31
3-12.	Dialog Screen 3	3-33
3-13.	Dialog Screen 4	3-35
3-14.	Dialog Screen 5	3-36
3-15.	Dialog Screen 6	3-37
3-16.	Dialog Screen 7	3-38
4-1.	Typical Template (Pass 1) Screen	4-3
4-2.	Typical Type Attribute (Pass 2) Screen	4-6
4-3.	Typical Initial I/O (Pass 3) Screen	4-7
5-1.	Modify Screen	5-2
6-1.	Summary Screen	6-3
6-2.	Special Editing Display	6-4
6-3.	Display Control and Error Message Display	6-5
6-4.	Function/Command Key Display	6-5
6-5.	Screen Format	6-6
6-6.	List Screen	6-6
6-7.	Sublist Screen	6-9
6-8.	Conditional Values Screen	6-11
6-9.	Input Record Screen	6-12
6-10.	Output Record Screen	6-13
7-1.	Screen Formats and Accompanying RPG II Specifications	7-7
7-2.	Sample SCREEN01	7-12
7-3.	COBOL Program Entries for SCREEN01	7-14
7-4.	Sample SCREEN02	7-16
7-5.	COBOL Program Entries for SCREEN02	7-17
7-6.	COBOL Program Entries for Indicators	7-20

TABLES

2-1.	Editing Rules for Signed Output Fields	2-6
2-2.	Editing Examples	2-7
3-1.	Characteristics Screen, Related Optional Screens, and Related Dialog Screens	3-11
3-2.	Dialog Screens and Their Functions	3-25
3-3.	Contents of Lines 9 and 10 of Dialog Screen 1	3-30
3-4.	Intensity/Emphasis Selections	3-36
4-1.	Basic CREATE Function Steps	4-1
4-2.	CREATE Function Steps Using Optional and Dialog Screens	4-11
7-1.	Status Keys	7-22
7-2.	Effects of Connect-Free Reporting	7-24
7-3.	Effects of Function Key Reporting	7-25

1. Introduction

1.1. PROVIDING INFORMATION VIA FORMS

One of the most common methods of providing and receiving information is through the use of forms. Whether we are filling them out, sending them in, or receiving them, almost every facet of activity, from recreation to business, involves the use of forms at some point. Applying for loans, credit cards, a driver's license; subscribing to magazines; preparing tax returns; even writing a check: doing any of these things necessitates handling a form.

Naturally, we take forms for granted and are even annoyed with the frequency at which we are required to handle them. We realize, however, that they provide one of the easiest and most effective ways for information to be supplied as well as received. They ensure a certain amount of uniformity that is essential whenever a transfer of information from one point to another is required.

How does this concern you? Well, since forms are capable of eliminating wasted time and confusion in other environments by standardizing the flow of information, they can provide a similar service in the data processing environment too: specifically by taking the complexity out of input and output methods. For this purpose, Sperry Univac has designed screen formatting facilities.

1.2. PROVIDING INFORMATION VIA SCREEN FORMAT SERVICES

Screen format services is an interactive component of OS/3 that enables you to create formats (forms), not on paper, but on a video terminal – your workstation. Additionally, it provides you with the capabilities for easily maintaining these formats.

Once created, a format can be used:

- To input all the necessary data to a program
- To display output data from a program
- To supply input and display output

Inputting and outputting data via a screen format is extremely fast and simple. It is done on a fill-in-the-blanks basis. Practically anyone can input data to a program in this manner merely by keying in the information that the format requests. For output, the blanks of a screen format are automatically filled in with the proper information. Such output is easier to read as well as to interpret because it is formatted.

Figure 1-1 is an example of a screen format used for input purposes.

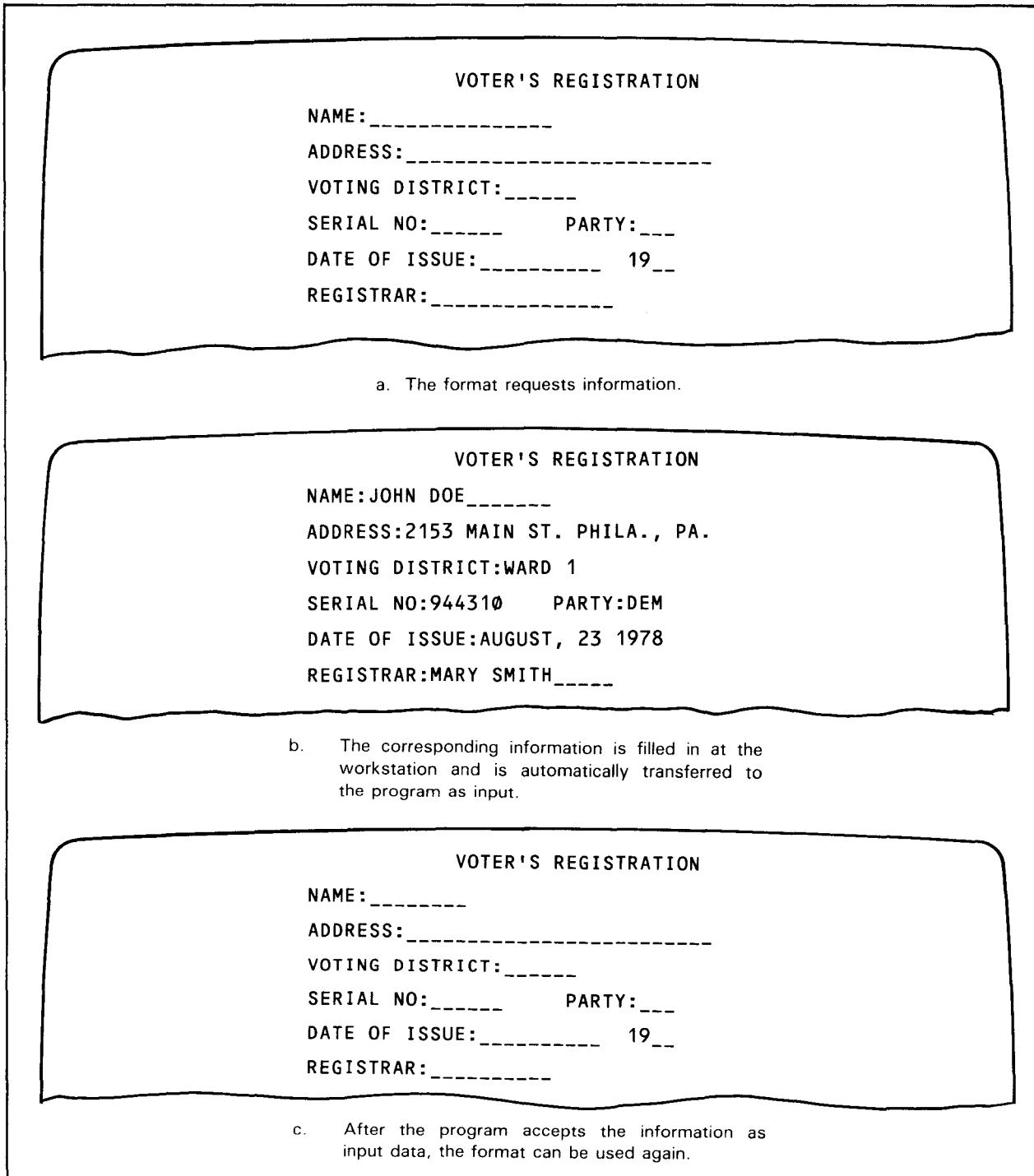


Figure 1-1. Screen Format Used to Supply Input Data

Figure 1-2 illustrates a screen format used for output purposes.

ACCOUNT STATUS REPORT

ACCOUNT NO:1827841	DATE:01/03/82
CREDIT:375.00	
DEFICIT:250.00	BALANCE:125.00

a. The screen format displays output.

ACCOUNT STATUS REPORT

ACCOUNT NO:1922640	DATE:01/03/82
CREDIT:250.00	
DEFICIT:100.00	BALANCE:150.00

b. The format can be used to display output of this type as long as necessary.

Figure 1-2. Screen Format Used to Display Output Data

Figure 1-3 illustrates how a screen format can serve both input and output purposes.

PERSONAL CREDIT REPORT

NAME:JOHN DOE		
ADDR:1552 MAIN ST.	STATE:PA	ZIP:19140
ACCOUNT NO:193-A564		
BALANCE:350.00		
PAYMENT:_____	DATE:__/__/_	

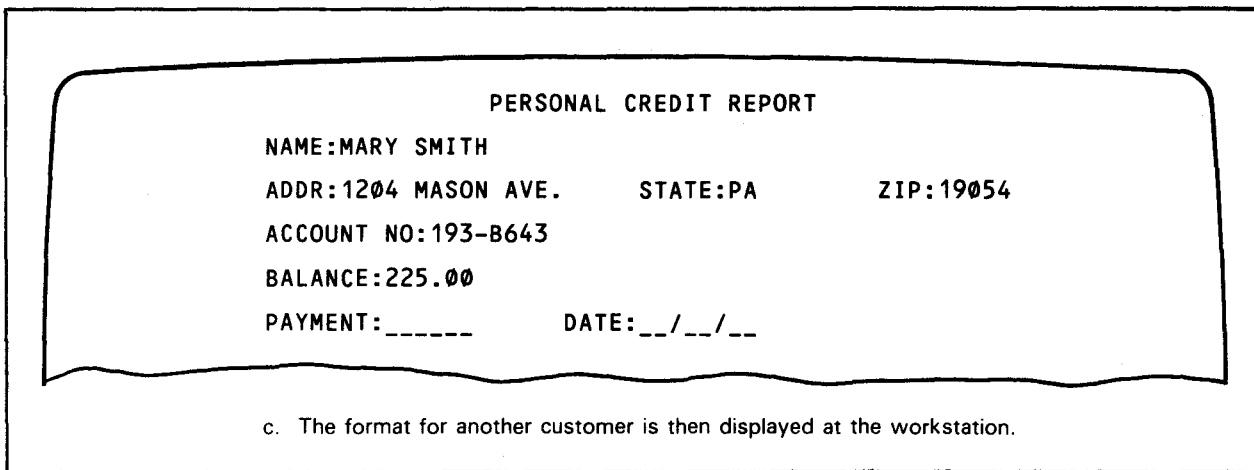
a. The format displays information for a particular customer and requests information as well.

PERSONAL CREDIT REPORT

NAME:JOHN DOE		
ADDR:224 PINE ST.	STATE:PA	ZIP:19102
ACCOUNT NO:193-A564		
BALANCE:350.00		
PAYMENT:25.00	DATE:12/23/82	

b. The workstation operator changes the customer's address and makes the appropriate entries for PAYMENT and DATE. Information on the format is passed to the program.

Figure 1-3. Screen Format Used for Input and Output (Part 1 of 2)



PERSONAL CREDIT REPORT

NAME:MARY SMITH

ADDR:1204 MASON AVE. STATE:PA ZIP:19054

ACCOUNT NO:193-B643

BALANCE:225.00

PAYMENT:_____ DATE:___/___/___

c. The format for another customer is then displayed at the workstation.

Figure 1-3. Screen Format Used for Input and Output (Part 2 of 2)

You create and use screen formats through the two software elements that make up screen format services: the screen format generator and the screen format coordinator.

The screen format generator (SFG) is the software element that allows you to generate screen formats at the workstation. During generation, you specify the layout of each format as well as other characteristics that best enable the format to meet the input and output requirements of a particular program. The SFG automatically stores the completed formats (as an FC-type module) for you either on a system library (\$Y\$FMT) that resides on the SYSRES volume or on a MIRAM user library that you specify when you create the formats. The SFG makes these formats readily available for you to modify, delete, or simply look at whenever necessary.

The screen format coordinator (SFC) becomes active when you actually want to use an existing format with a program.

The SFC automatically gets the format you request and displays it on the workstation screen so that filling in the blanks can begin. The SFC is, as its name implies, responsible for coordinating the transfer of input and output data between your program and the format that's displayed on the workstation screen.

Figure 1-4 illustrates how the SFG and SFC process your formats.

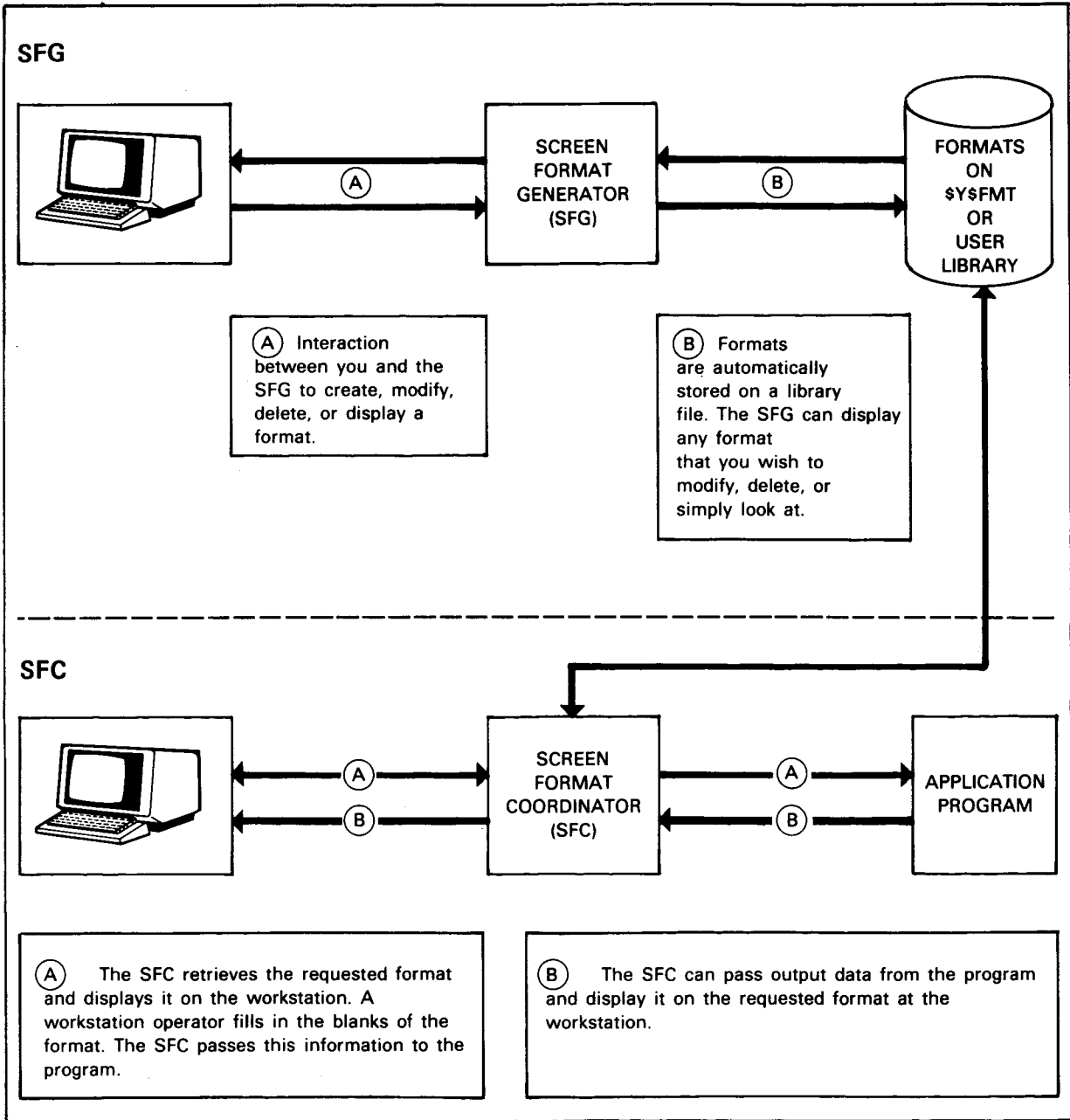


Figure 1-4. The Function of the SFG and SFC

Together, the SFG and the SFC provide you with everything you need to create, maintain, and ultimately use screen formats.

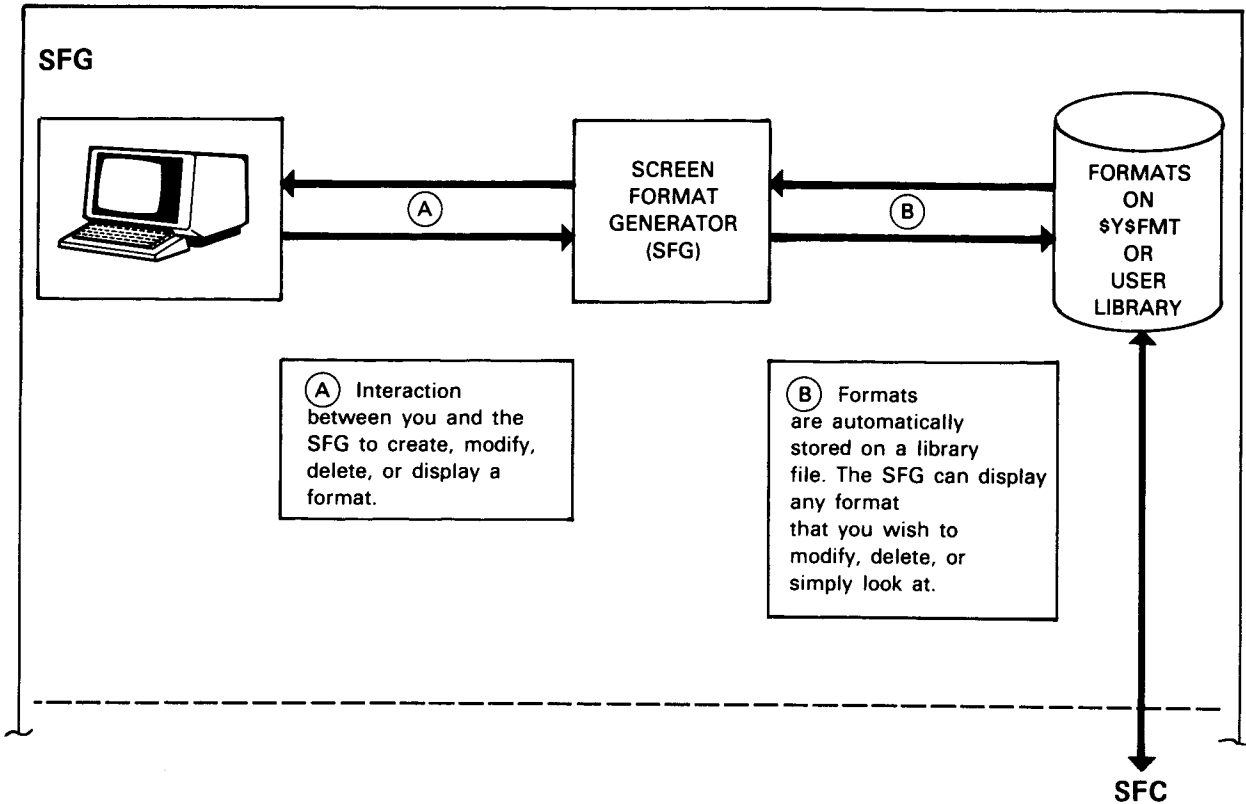
NOTE:

To be used successfully for screen format services, remote communications terminals (the UTS 200 or UTS 400) must have the field protect feature. If this feature is not provided, an error message (SF16) will occur whenever the operator enters data.

1.3. SCREEN FORMAT GENERATOR

1.3.1. Functions of the SFG

Let's concentrate on the functions of the SFG:



The SFG lets you create a format, modify (change) a format, display a format, and delete a format. With the ease of use characteristic of the SFG, practically anyone can perform these tasks – there's no need to be an experienced computer programmer.

You begin a format's creation (generation), for example, by answering a series of questions, collectively called the home screen, that are displayed at the workstation (see 3.2). You then key in the layout for the desired format. An example of such a layout is:

```
                PERSONAL CREDIT REPORT   TYPE:_  DATE:___/___/___  
NAME: _____  
ADDR: _____ STATE:___  ZIP: _____  
ACCOUNT NO: _____  
PAST DUE AMOUNT: _____  
NEW BALANCE: _____  PAYMENT DUE DATE:___/___/___  
MINIMUM PAYMENT: _____
```

There are several steps, or passes, in which you can provide more specific information about the format, including whether it will be used for input, output, or both. These passes (explained in Section 4) are easy to complete and are generally the same for every format, so that after you've created one format, subsequent creations and modifications become almost second nature.

Other SFG functions are even simpler to complete. Before changing a format, for example, you might want to examine it to decide exactly what changes need to be made. In such cases, the SFG can automatically provide you with a display of any format stored on the library file. You need only supply the format's name. The format's name is likewise the criterion for deleting any format from the library. You supply the name, and the SFG does the work.

The SFG has additional features to make any of the format-related functions, particularly creation and modification, even easier to complete.

■ Error Indication

The SFG automatically notifies you of a variety of mistakes as soon as they are made. In most cases, the error is blinked on the workstation screen. In some instances, a specific error message is displayed. Thus, you are made aware of and can correct certain errors as soon as they occur.

■ Prompting

If you're not quite sure how to begin creating or modifying a format or if you're having difficulty along the way, you can ask the SFG for help simply by pressing a particular workstation function key (F13). The SFG then displays the instructions necessary to complete the operation causing the difficulty. You can be prompted in this fashion through as much or as little of any function (creating a format, modifying a format, etc) as necessary.

1.3.2. Screen Format Components

You now know that the SFG enables you to create, modify, delete, or display a format with a minimum of difficulty. Before we begin a more detailed discussion of these functions, however, you should know a little more about the nature of the format itself.

Formats can vary in size and complexity, as shown in the following screens.

```
          SALES REPORT
    PRODUCT: _____
    SALESMAN: _____
    TOTAL SALES: _____
    DATE: _____
```

```
          CLASS RANK
    NAME: _____
    GRADE: _____
    GRADE POINT AVERAGE: _____
    CLASS RANK: _____
```

```
          WAREHOUSE INVENTORY CONTROL
    PROD. DESC: _____
    STOCK NO: _____
    MFR: _____
    QTY ON HAND: _____
    QTY ON ORDER: _____
    REORDER QTY: _____
    QTY DISCOUNT: __ % FOR QTYS OVER: _____
```

No matter how simple or complex, a format consists of only two components: display constants and variable data fields.

1.3.2.1. Display Constants

Display constants are portions of a screen format that never vary. For the most part, they serve as either headings or titles to introduce the information that will eventually fill in the blanks that follow. The shaded portions of the following format are display constants.

```
          SALES REPORT
PRODUCT:-----
SALESMAN:-----
TOTAL SALES:-----
DATE:-----
```

Once a format is displayed on the workstation screen in the first step of its creation, you cannot alter display constants in any of the successive steps. To change a display constant, you must initiate one of the SFG functions for modifying the format after it's created.

Likewise, when a format is being used to input data to or output data from a program, a display constant cannot accidentally be changed or blanked out either by the workstation operator or by any screen activity that results from the program. The program can never reference a display constant.

Other characters on a format, such as the slashes (/) separating the month, day, and year in a date, are also display constants. As a general rule, any character, other than an underline, that you enter when you first lay out the format is considered a display constant.

1.3.2.2. Variable Data Fields

Those portions of a format that are used either to input data to or to output data from a program are variable data fields. When creating a format, you always indicate variable data fields by underlines, with the number of underlines representing the external length of the field.

In successive steps during a format's creation, you make specifications as to the type of variable data a field receives (numeric, alphabetic, etc). Once the format is laid out, the remainder of the creation process deals exclusively with specifying everything from general to detailed characteristics for each variable data field in the format. The field's input and output directions are also specified, and since this (inputting and outputting) is the purpose behind generating formats in the first place, we'll discuss I/O directions briefly here.

Every variable data field must belong to one of three I/O classifications: input, output, or both input and output (bidirectional).

■ Input

Those fields designated during creation as input fields can only be used by the workstation operator to enter input data to a program at run time.

■ Output

Those fields designated during creation as output fields can only be used to display output data from the system at run time.

■ Both

Those fields designated during creation as bidirectional can be used either to input data to the program or to display output data from the system at run time.

The first time a format is displayed (when a program selects the format), input fields are (usually) represented with underlines. For example:

NAME: _____

Output fields, on the other hand, always display data.

NAME: MARY SMITH

An output field will never be displayed as

NAME: _____

unless the program supplies that field with underline characters.

A bidirectional field is always expected to receive output the first time a format is displayed. In other words, it acts initially as an output field.

NAME: MARY SMITH

Depending on the program specifications, a bidirectional field can continue to display output, or it can accept input, or it can do both. In the preceding field (NAME), the workstation operator could, for example, change the last name simply by entering a new name over it.

NAME: MARY JONES

The program accepts the new name as input.

The format shown in Figure 1-1 (1.2) contains only input fields. The fields in Figure 1-2 could be either output fields displaying output or bidirectional fields displaying output. In Figure 1-3a, the fields NAME, STATE, ZIP, ACCOUNT NO, and BALANCE could be either output or bidirectional fields. You can be sure, however, that ADDR is a bidirectional field since the address is changed in Figure 1-3b. The fields PAYMENT and DATE are strictly input fields.

The procedure for specifying I/O directions for variable data fields is discussed in 3.5.1.

NOTE:

Any reference to fields in the remainder of this manual refers to variable data fields only.

1.3.2.3. Protected and Unprotected Parts – Low-Intensity Display

As you'll recall from 1.3.2.1, once defined, display constants cannot be altered either while successive passes of the creation process are being performed or while the completed format is being used with a program. These and other parts of a screen format with similar characteristics can be described as protected.

Depending on the particular part, there are differences in the extent to which it is protected. Also, in most cases the SFG automatically protects a part simply by virtue of its characteristics. In some cases, however, you can specify the protection yourself.

■ Display Constants

Display constants are automatically protected as soon as you finish laying out the format. They remain so throughout the rest of the format's creation and also throughout any subsequent use of the format with a program until the entire format display is erased. Display constants are therefore totally protected and can only be changed via one of the SFG's modification functions.

■ Output Fields

As explained in 1.3.2.2, a field that is specified for output during a format's creation is used exclusively to display variable data from the system. Displayed output data cannot be changed or erased by the workstation operator. Only new output coming from the system can overwrite previously displayed output. Naturally, output will be erased whenever the entire screen display is erased. It is in this sense that output fields are automatically protected.

■ Specific Characters Used for Editing Input or Bidirectional Fields

At a certain point during creation of the format, you can specify editing for certain variable data fields by keying in COBOL-like picture clauses. If editing is specified for an input or bidirectional field, you have the option of specifying protection for certain characters in the picture string: the comma (,) and the decimal point (.). The SFG automatically protects a fixed currency symbol (\$). These characters become a sort of edit mask and, because they are protected, cannot be altered by a workstation operator or by the activity of the system.

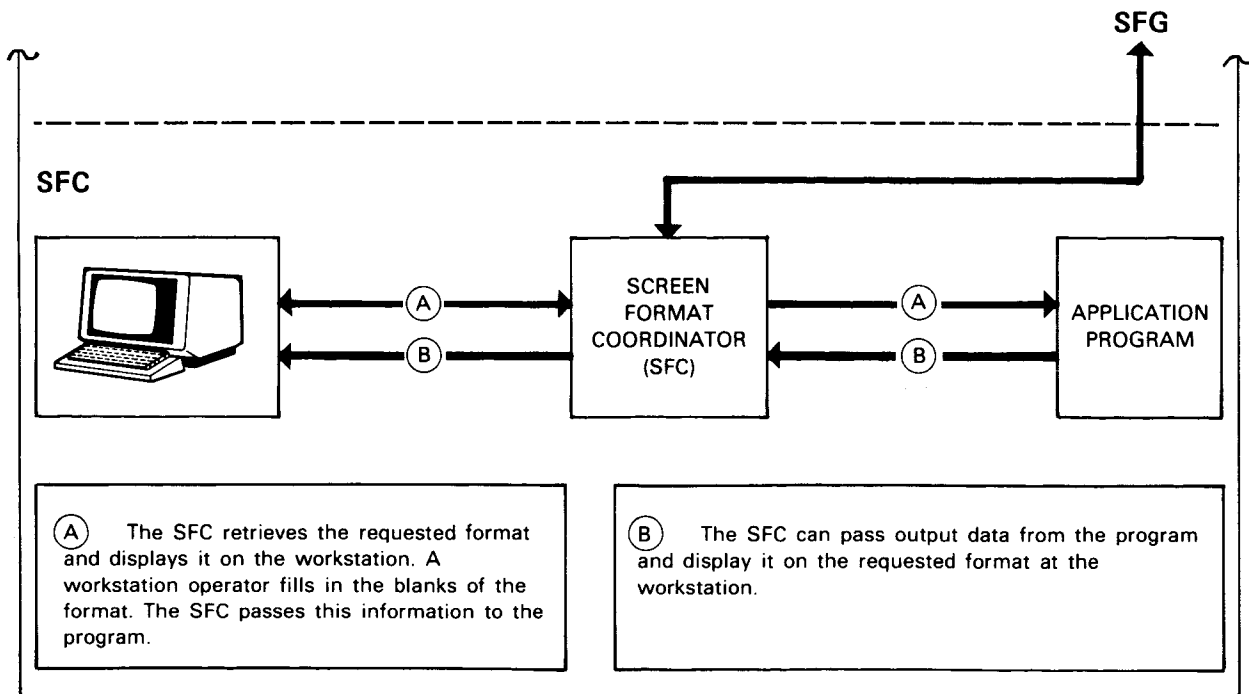
Although you should know what protection means, it isn't necessary for you to memorize which parts of a screen format are protected since the SFG makes the distinction. Protected parts are displayed on the workstation in low intensity. Low-intensity display simply means that the protected parts (characters) appear less bright than those that are unprotected.

NOTE:

During the creation or modification functions, there are several forms displayed by the SFG (the home screen, for example) the answers to which provide certain information about the format. Although not physically related to the format layout, portions of these displays are also protected and are displayed in low intensity.

1.4. SCREEN FORMAT COORDINATOR (SFC)

As mentioned earlier, the SFC becomes active when you want to use an existing format with a particular program for the purpose of inputting and outputting.



The SFC is responsible for retrieving the format (F-type module in \$Y\$FMT or your user library) you requested and displaying it on the workstation screen. Since the format is completely separate from your program, the SFC must also coordinate the flow of data between the program and the format.

In addition to transferring data back and forth, the SFC performs other necessary operations, such as checking or validating any input data that is filled in by the workstation operator. The data is checked against the specifications that you made for the field when you created or modified the format.

As will be explained in Section 4, there is a specific pass, or step, in the format's creation (for example, pass 2 - type attribute screen) where you indicate whether the field is to receive numeric, alphanumeric, or alphabetic data. If a field does not receive this type of data, the SFC will not accept the data as input to the program. Incorrect input is blinked on the workstation screen.

Since the SFC performs these and similar operations automatically, there's no need for you to be concerned with its activities at any time during the format's use. Simply be aware that the SFC handles the format according to the specifications you made during creation or modification via the SFG.

Although the SFC assumes the bulk of the responsibility for handling program input and output, you are responsible for such things as the proper job control statements associated with screen format services and for identifying as well as calling that format within the program. Specifics concerning job control statements and the necessary program statements as they relate to the particular language being used are discussed in Section 7.



2. Type Attributes and Editing Rules

2.1. CONSIDERATIONS FOR VARIABLE DATA FIELDS

As mentioned in 1.3.2.2, when you create a format, you specify variable data fields to handle the program's input and output data. In a specific step during the format's creation, you must define the type of data that each field is to receive. Data for variable fields can be in one of the following types: alphabetic, numeric, or alphanumeric. These are referred to as field type attributes.

With the SFG, you have the option of specifying editing for numeric fields and, to some extent, for alphabetic and alphanumeric fields as well. Since editing can be specified during creation of the format, your program has only to indicate such things as assumed decimal places (v) and signs (s) whenever appropriate, but not edit masks.

In Section 4, we'll discuss specifying type attributes and editing characters when creating a screen format. However, knowing the rules for type attributes and editing beforehand can be helpful.

2.2. TYPE ATTRIBUTES

The rules for indicating field types are similar to the rules applying to COBOL picture clause expressions:

■ Alphabetic Fields

You use one or more A's to indicate that a particular field will receive alphabetic data when the format is in use. However, duplication factors (as A(2) in COBOL to indicate an alphabetic field that is two characters in length) are not supported.

A field defined as alphabetic can, at run time, contain any combination of the 26 characters of the alphabet and the space. These characters will be left-justified with blank padding.

■ Numeric Fields

You use one or more 9's to indicate that a format field can receive numeric data. Duplication factors are not supported. A field defined as numeric can, at run time, contain any combination of the numeric characters 0 through 9. A field of this type will be right-justified with zero padding.

■ Alphanumeric Fields

You can use any combination of X's, A's, or 9's to represent an alphanumeric field. Duplication factors are not supported. The positions indicated by X's can, at run time, contain any combination of EBCDIC characters. Positions indicated by 9's or A's can only contain the numbers 0 through 9 and alphabetic characters respectively. A field defined as alphanumeric will be left-justified with blank padding.

2.3. EDITING

Once you decide whether a field is to be alphabetic, alphanumeric, or numeric, your next consideration is whether you want to edit the field. Editing can be specified for any field in the format, although in some cases certain types of editing and edit characters restrict the field to handling output data only.

2.3.1. Editing Alphabetic and Alphanumeric Fields

You may use an exclamation point (!) to edit alphabetic and alphanumeric fields as shown in the following examples.

```
AAA!AAA  
XX!XX!XX  
XX!99!AA
```

This character reserves the position where it appears so that an insertion character that you select (such as a dash or a slash) can replace it. Positions indicated by the exclamation point are always protected and are never counted in the field's internal length. A field edited in this manner may be used for input, output, or both.

You may use the characters 0 (zero) and B in an alphanumeric field as shown in the following examples.

```
XXBXX00XX  
99BB0AA
```

These characters reserve the positions where the numeral 0 and the space will be automatically inserted at run time. An alphanumeric field using either of these two characters may not use the exclamation point and vice versa. Also, such a field may be used only to display output data. The 0 and the B are counted in the external but not the internal field length.

NOTE:

External field length refers to the appearance of the field on the workstation screen, whereas internal length refers to the length of the field when referenced by the user program.

2.3.2. Editing Numeric Fields

You can edit a numeric field by using an exclamation point (!), insertion characters, or suppression.

The exclamation point gives you the same capabilities for numeric fields as it does for alphabetic and alphanumeric fields. A numeric field using the exclamation point may not, however, use any other form of editing and vice versa. Other SFG editing rules are similar in most aspects to COBOL editing rules.

The valid characters you can use for insertion or suppression editing are: asterisk (*), Z, plus sign (+), minus sign (-), CR, DB, decimal point (.), comma (,), blank (B), zero (0), and currency sign (\$).

■ Insertion Editing

There are three types of insertion editing that you can specify during format generation: simple, fixed, and floating.

- Simple Insertion

You may insert the comma (,), the blank (B), and zero (0) into any numeric field as shown in the following examples.

```
9,999
9BB99900
```

The comma may be used in an input, output, or bidirectional field. Fields using the B or 0 are always output fields.

- Fixed Insertion

You may insert the currency sign (\$), plus sign (+), minus sign (-), credit symbol (CR), debit symbol (DB), comma (,), and decimal point (.) into any numeric field as shown in the following examples.

```
$9,999.99
+9.99
9,999.99CR
```

Fields using CR or DB are always output fields.

- Floating Insertion

You may express the currency sign (\$), plus sign (+), and minus sign (-) as floating signs as shown in the following examples.

\$\$\$9.99
+999

The sign represents the leftmost position (before the first significant digit) into which it (\$, +, or -) will be inserted. A field edited in this manner may only be used for output.

■ Suppression/Replacement Editing

You may do this type of editing in one of two ways:

- The character Z represents the suppression of leading 0's (up to the first significant digit) and their replacement with blanks as shown in the following examples.

ZZ,ZZ9
\$Z,ZZ9.99

- The asterisk (*) represents the suppression of leading 0's (up to the first significant digit) and their replacement with the * (e.g.,** 9.99).

A field edited either way may only be used for output.

Refer to the 1974 ANSI COBOL programmer reference for more information concerning insertion and suppression/replacement editing for numeric fields.

2.4. RESULTS OF EDITING FORMAT FIELDS

You define type attributes and editing for variable data fields in pass 2 of the format's generation (see 4.1.2). In a later step (dialog screen 1, 3.5.1), you specify the characters to replace any exclamation points in a field, and optionally you can specify protection for certain characters in edited numeric fields (that are input or bidirectional), thus creating an edit mask.

Editing affects the display of a field when the format is presented at the workstation and consequently the way in which data can be entered (for input and bidirectional fields) by the workstation operator. Editing also affects the field's internal length, and in some cases it restricts a field to handling output data only.

How editing determines the workstation display and operator entries is especially obvious with edited numeric fields that are used for input.

Assume that during the format's creation a numeric input field is described as follows:

\$9,999.99

Since the SFG automatically protects a fixed currency sign, the field will be displayed as follows when the format is presented at the workstation:

\$_____

As long as the program indicates the number of decimal positions (two in this case), the workstation operator can make entries similar to any one of the following:

\$1,234.56

\$1.23____

\$1234.56_

\$1234_____

Decimal point alignment is done automatically (by the SFC). If no decimal point is entered (e.g., \$1234), it is assumed to be to the right of the rightmost digit (\$1234.). Errors are indicated by blinking if the workstation operator makes an incorrect entry, such as:

\$123456__

The entry is in error because only four digits to the left of the decimal point are permitted (as indicated by the editing specified during the format's creation and the number of decimal places provided for in the program). If the workstation operator enters too many digits to the right of the decimal point (e.g., \$123.456), the value is accepted but the last digit (6) is truncated.

Suppose you want this field to be displayed on the workstation as follows:

\$,____.____

In this case, you perform the same editing (\$9,999.99), but in addition you specify protection for the comma and the decimal point. The workstation operator must then enter the value in its proper position according to the field display. If the value is 1.23, the operator enters:

\$0,001.23.

Now consider format fields that use the sign +, -, CR, or DB. If you describe an input field as +999 or -999 during format generation, it will appear as _ _ _ when the format is displayed at the workstation. As long as the program indicates a sign for this field (e.g., S999 as in COBOL), the workstation operator can enter any 3-digit value with or without a sign. For example:

+123

-123

123_

The value with no sign is assumed to be positive.

For output fields using a sign, the (COBOL) rules in Table 2-1 apply.

Table 2-1. Editing Rules for Signed Output Fields

If the Field Is Defined with	And the Output Value Is	Then the Output Is Displayed on the Format with
+ (+999)	Positive or zero	+ (+123)
	Negative	- (-123)
- (-999)	Positive or zero	Space (123)
	Negative	- (-123)
CR (999CR)	Positive	Two spaces (123)
	Negative	CR (123CR)
DB (999DB)	Positive or zero	Two spaces (123)
	Negative	DB (123DB)

Fields that use the exclamation point (!) for editing are displayed on the format with whatever character you choose to replace the exclamation point with. Assume that you define the following field during the format's generation (pass 2).

SOCIAL SECURITY:999!99!9999

Later on during generation, you select a dash (-) to replace the exclamation point. If the field is used for input, it will be displayed as follows:

SOCIAL SECURITY:___-___-___

The operator can make the appropriate entries. If the field is used for output, it will display the data that is passed from your program to the format. For example:

SOCIAL SECURITY: 195-44-8867

Remember that edit characters used for simple or fixed insertion (such as B, O, \$, +, -, CR, DB, !, and ,) are never counted in the field's internal length. Also, fields containing any acceptable combination of the floating signs \$, +, or -; the insertion characters CR or DB; or the suppression/replacement characters * or Z may only be used to display output data.

Table 2-2 lists some editing examples and the effects of that editing on the format fields.

Table 2-2. Editing Examples

Field Type	Type Attributes and Editing Specified	External Length	Internal Length*	Field Display
Alphabetic	AAA	3	3	INPUT: _ _ _ OUTPUT: ABC
	AAA!AAA	7	6	INPUT: _ _ _ _ _ _ _ OUTPUT: ABC-DEF (if dash is selected for replacement)
Alphanumeric	XXX	3	3	INPUT: _ _ _ OUTPUT: # 1A
	XXI99!AA	8	6	INPUT: _ _ _ _ _ _ _ OUTPUT: # 1-23-AB (if dash is selected for replacement)
	XXB9900XX	9	6	OUTPUT; ONLY: A3 4500BC
Numeric	999	3	3	INPUT: _ _ _ OUTPUT: 123
	99!99!99	8	6	INPUT: _ _ _ / _ _ / _ _ OUTPUT: 12/24/80 (if slash is selected for replacement)
	\$9,999.99	9	6	INPUT: \$ _ _ _ _ _ _ _ or (if , and . are protected) INPUT: \$ _ , _ _ _ _ _ _ _ OUTPUT: \$1,234.56
	ZZ9.99	6	5	OUTPUT ONLY: 123.45 or 12.34
	\$\$9.99	6	4	OUTPUT ONLY: \$12.34 or \$ 1.23

*Internal lengths indicated are based on the internal usage of display.



3. Using the Screen Format Generator (SFG)

3.1. ACTIVATING THE SFG

The first thing you must do to perform any of the SFG functions is activate the SFG. Do this on the workstation:

1. Key in LOGON and your user ID.
2. Press the XMIT key.
3. Get into system mode.
4. Key in RV SFGEN.
5. Press the XMIT key.

A formatted display called the home screen then appears on the workstation screen. The home screen is like a fill-in-the-blanks form: you enter the function you want to perform, as well as information about your screen format.

3.2. THE HOME SCREEN

The home screen (Figure 3-1) is always the first display presented whenever you activate the SFG. On it, you choose one of the eight SFG functions. If you're working on a format that already exists, you provide its name and library. If you're creating a new format, you name it, indicate where it will be stored, and, if necessary, allocate file space for it.

```

1. FUNCTION (1): 1 CREATE      2 CREATE-FROM    3 MODIFY      4 DELETE
2.                5 SHOW        6 LIST        7 SPOOL       8 TERMINATE
3. OLD FORMAT NAME (_____) IS ON THE FOLLOWING LIBRARY:
4. FILE NAME: ($Y$FMT          ) VOLUME: (RES      )
5.
6. NEW FORMAT NAME (_____) IS TO BE STORED ON THE FOLLOWING LIBRARY:
7. FILE NAME: ($Y$FMT          ) VOLUME: (RES      )
8. IF THIS FILE DOES NOT EXIST, ALLOCATE (002) CYLINDERS. INCREMENT IS (01) CYL.
9.
10. **FUNCTION KEYS ARE:F1-GO TO HOME SCREEN, F5-BREAKPOINT SPOOL FILE,
11.     F13-HELP, F14-EXIT HELP, F20-RESTORE SCREEN

```

Figure 3-1. The Home Screen

Subsection 3.6 discusses how to position the cursor and how to overwrite values on the screen.

3.2.1. Functions Provided by the SFG

On line 1 of the home screen, you specify one of the eight functions provided by the SFG:

```

FUNCTION (1):  1 CREATE      2 CREATE-FROM    3 MODIFY      4 DELETE
                5 SHOW        6 LIST        7 SPOOL       8 TERMINATE

```

■ CREATE

Use this function to create a screen format. The first step is to complete and transmit the home screen. The SFG then displays another fill-in-the-blanks screen called the characteristics screen. This screen asks you for format-related information and offers several options. Some of these options have other screens associated with them. If you choose these particular options, their optional screens are also displayed.

Once you've provided all the necessary information about your format, a blank screen is displayed for you to lay out your screen format. You supply any display constants and indicate any variable fields and their lengths. In two subsequent passes, you assign type (editing) attributes and input/output usages for each field on the format.

You also have the option of providing more information for each field in the format by initiating dialogs for those fields. Once your format is complete and transmitted, the home screen is displayed for you to choose another function.

- **CREATE-FROM**

Use this function to create a new format by modifying an existing format. You can change the format as a whole, or you can change specifications on some or all of the fields in the format. The SFG saves the old format.

- **MODIFY**

Use this function basically as you use the CREATE-FROM function. The only difference is that the SFG doesn't save the old format. The new format replaces the old format.

- **DELETE**

Use this function to delete an existing format.

- **SHOW**

Use this function to display an existing format and its type attributes on the workstation screen.

- **LIST**

Use this function to display information for each field within an existing format. The display includes such information as the creation date, the location of the fields on the screen format (x,y coordinates), input/output usage of the fields, internal and external field lengths, range checking and replenishing values, and conditional indicators associated with the field.

- **SPOOL**

Use this function to get a printout of an existing screen format, along with the field information provided by the LIST function.

- **TERMINATE**

Use this function to terminate the SFG session and return control to the system.

3.2.2. New Format Name and Library

Every screen format must have a name and library. When you're creating a format, name it and indicate where it will be stored via lines 6 and 7 of the home screen:

```
NEW FORMAT NAME (_____) IS TO BE STORED ON THE FOLLOWING LIBRARY:
FILE NAME: ($Y$FMT                ) VOLUME: (RES          )
```

Line 6 asks for the format name. The name can be eight characters and can contain only the letters A to Z, the numerals 0 to 9, and the special characters

“ # \$ @ and ?

Spaces are not allowed within the name.

Lines 6 and 7 ask for the library where the format will be stored. \$Y\$FMT is a system file for storing formats; it is the default for FILE NAME. \$Y\$FMT resides on the system resident volume (RES). Thus, RES is the default for VOLUME. If you choose the defaults, no keyin is required. You're ready to transmit the home screen. You can, however, store your format on an alternate MIRAM library. To do so, overwrite \$Y\$FMT and RES with the appropriate file name and volume name.

If the alternate file you specify doesn't exist (hasn't been allocated), allocate space for it on the volume. Line 8 asks about the file space you need:

```
IF THIS FILE DOES NOT EXIST, ALLOCATE (002) CYLINDERS. INCREMENT IS (01) CYL.
```

If you need more than two cylinders for your file, overwrite the 002 (the default) with the number you need. The same applies for INCREMENT, which is the number of extra cylinders added to your file each time the file fills up.

The CREATE function isn't the only function that requires a new format name. The CREATE-FROM function also needs a new name because you're creating a whole new format based on the characteristics of an old one. The old format itself does not change, nor does its name.

3.2.3. Old Format Name and Library

The functions that require the old format name and library are:

CREATE-FROM	SHOW
MODIFY	LIST
DELETE	SPOOL

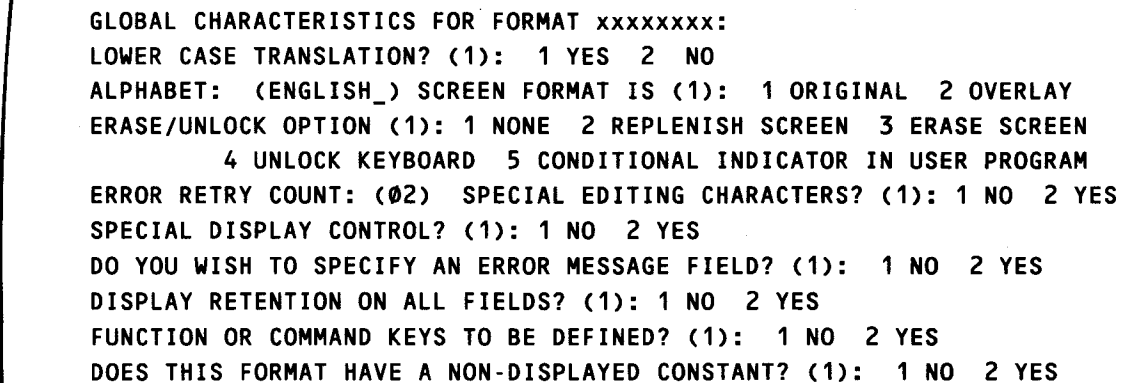
On lines 3 and 4, enter the name and library of the format you want to work with. If the format is stored on `$$FMT`, no further keyin is required. Just transmit the home screen. If the format is stored on some other library, enter the file name and volume where the format is located and then transmit the home screen.

NOTE:

To get a listing of existing format names, issue the `FSTATUS` command. For an explanation of this command and accompanying parameters, refer to the interactive services concepts and facilities user guide/programmer reference.

3.3. THE CHARACTERISTICS SCREEN

When you're creating a new format, the next screen displayed after the home screen is the characteristics screen. This screen (Figure 3-2) asks for more information about your format.



```
1. GLOBAL CHARACTERISTICS FOR FORMAT xxxxxxxx:
2. LOWER CASE TRANSLATION? (1): 1 YES 2 NO
3. ALPHABET: (ENGLISH_) SCREEN FORMAT IS (1): 1 ORIGINAL 2 OVERLAY
4. ERASE/UNLOCK OPTION (1): 1 NONE 2 REPLENISH SCREEN 3 ERASE SCREEN
5. 4 UNLOCK KEYBOARD 5 CONDITIONAL INDICATOR IN USER PROGRAM
6. ERROR RETRY COUNT: (02) SPECIAL EDITING CHARACTERS? (1): 1 NO 2 YES
7. SPECIAL DISPLAY CONTROL? (1): 1 NO 2 YES
8. DO YOU WISH TO SPECIFY AN ERROR MESSAGE FIELD? (1): 1 NO 2 YES
9. DISPLAY RETENTION ON ALL FIELDS? (1): 1 NO 2 YES
10. FUNCTION OR COMMAND KEYS TO BE DEFINED? (1): 1 NO 2 YES
11. DOES THIS FORMAT HAVE A NON-DISPLAYED CONSTANT? (1): 1 NO 2 YES
```

Figure 3-2. The Characteristics Screen

Some selections on the characteristics screen prompt additional screens. These are called optional screens and are discussed in 3.4. Optional screens further define certain run-time characteristics of your format. They automatically appear in sequence after you complete and transmit the characteristics screen.

Note that the first line of the characteristics screen always contains the format name you supplied on the home screen:

```
GLOBAL CHARACTERISTICS FOR FORMAT xxxxxxxx:
```

where xxxxxxxx is the 8-character name of your format.

3.3.1. Lowercase Translation (Line 2)

Line 2 asks you to specify whether lowercase alphabetic characters, entered by the workstation operator for the format's variable data fields, will be translated to uppercase before being passed as input to your program.

LOWER CASE TRANSLATION? (1): 1 YES 2 NO

As indicated by the parentheses, the default value for the question is 1 (YES). If you accept the default, lowercase alphabetic characters provided as input by a workstation operator are translated to uppercase before they are passed to the program. If you select 2 (NO), lowercase alphabetic characters aren't translated to uppercase.

NOTE:

No lowercase translation may be performed on Katakana workstations; therefore, you must overwrite the default value with 2 (NO), specifying no lowercase translation.

3.3.2. Alphabet (Line 3)

If you're creating (CREATE or CREATE-FROM) or modifying (MODIFY) a format, line 3 lets you indicate whether input provided by a workstation operator at run time is in a language other than English:

ALPHABET: (ENGLISH_)

English is the default alphabet. Appendix D lists the other valid alphabets for which translation tables are available.

3.3.3. Original and Overlay Formats (Line 3)

An original format is simply the average screen format whose purpose is inputting and outputting information to and from your program. An overlay format serves the same general purpose; however, it is created for use with an original format. Your response to this part of line 3 indicates whether the format you're creating or modifying is an overlay or an original.

SCREEN FORMAT IS (1): 1 ORIGINAL 2 OVERLAY

An overlay format is created separately and may start at the beginning of any line on the screen. When selected by your program, it completely replaces (overlays) a portion of or acts as an addition to the original format that's already displayed. At this time, only the overlay format is used for inputting and/or outputting because it's the only format recognized. The original format remains unchanged and exists only for display purposes. Creation of an overlay format and specific examples of its use are discussed in 4.3.

3.3.4. Erasing and Unlocking Options (Lines 4 and 5)

When your program first displays a format on a screen, output-only and bidirectional fields contain data from your program, and input fields are ready to be filled in. When a workstation operator completes the format and transmits it back to your program, the next thing that occurs depends on the erase/unlock option you specified when you created the format. Your choices are displayed on lines 4 and 5 of the characteristics screen:

```
ERASE/UNLOCK OPTION (1):  1 NONE  2 REPLENISH SCREEN  3 ERASE SCREEN  
                          4 UNLOCK KEYBOARD  5 CONDITIONAL INDICATOR IN USER PROGRAM
```

■ NONE

This is the default option. It indicates that when a workstation operator transmits a format the SFC verifies the input data. When verification is completed, the screen format itself remains on the workstation screen and the keyboard remains locked until your program requests the display of the same or a new format. This option offers you the most flexibility, in that your format isn't redisplayed automatically. If you intend to use conditional indicators to control your format's run-time characteristics (discussed later in this section), specify either this option or the `CONDITIONAL INDICATOR IN USER PROGRAM` option.

■ REPLENISH SCREEN

This option indicates that when a workstation operator transmits a format the SFC verifies the input data. Then, if verification is successful, the SFC immediately replenishes the input-only fields with a stringed constant (normally underlines) without waiting for your program to actually accept the input. The keyboard is then unlocked. Thus, a workstation operator can begin filling out another screenful of data without waiting for your program to process the previous input data. This option is designed specifically for data entry applications that simply gather information from one format with a minimum of input verification. The formats should contain only input fields and should rely only on the verification that the SFC can perform.

■ ERASE SCREEN

This option indicates that after your program displays a format and an operator transmits the contents back to your program, the format is erased and the keyboard is unlocked. This option is recommended for applications that only intermittently use screen formats. It is not recommended for continuous data entry applications because there is no looping back to redisplay a format.

■ UNLOCK KEYBOARD

This option indicates that when an operator transmits a format, the SFC verifies the data, leaves the keyboard unlocked, and immediately allows a workstation operator to enter new data. It doesn't wait for your program to accept the input before it allows an operator to enter the next screenful of data. In this respect, it resembles the REPLENISH SCREEN option. The difference between the two is that UNLOCK KEYBOARD doesn't replenish input fields with a stringed constant (typically underlines). Instead, it leaves the screen display as is, where input fields contain the previously entered data. All a workstation operator has to do is overwrite the old contents of the field with new data. Like REPLENISH SCREEN, this option is recommended for data entry applications where formats contain only input fields that require a minimum of verification.

■ CONDITIONAL INDICATOR IN USER PROGRAM

This option lets you set conditional indicators that determine your format's status. Depending on the conditions you specify and test for in your program, the input fields are replenished, the entire format is redisplayed, or the whole format is erased. You supply conditional erase/replenish indicators on an optional screen called the conditional erase/replenish screen. This screen (as well as any other applicable optional screens) appears after you complete and transmit the characteristics screen. For an explanation of the conditional erase/replenish screen, see 3.4.2.

A user program may erase the screen format or replenish input values by issuing explicit statements (e.g. a BAL programmer would use consolidated data management macros). The keyboard may also be unlocked this way. NONE or UNLOCK KEYBOARD should be chosen in this case. See 7.3.4 and the consolidated data management macro language user guide/programmer reference for the data management macros that can be used for screen control.

NOTE:

Do not specify REPLENISH SCREEN, ERASE SCREEN, or UNLOCK KEYBOARD if you intend to use range checking for any field in your format.

3.3.5. Error Retry Counts (Line 6)

Your answer to this question determines the number of times a workstation operator may attempt to correct any input that is entered incorrectly.

ERROR RETRY COUNT: (02)

The value 2 is the default value for the number of times correction may be attempted. However, you may specify any value up to 99 by overwriting the default. If you specify 0, the operator can't perform a retry.

At run time, incorrect input is blinked (or displayed with the display attributes you select for fields entered in error) to the workstation operator. If the operator can't correct the input and the value specified for the retry count is reached, then an error status is returned to your program. You may make provisions within your program for this situation.

3.3.6. Special Editing Characters (Line 6)

This line asks if you'll use special editing characters for numeric edited fields:

SPECIAL EDITING CHARACTERS? (1): 1 NO 2 YES

If you specify YES, an optional screen called the edit screen appears after you've transmitted the characteristics screen. The edit screen is discussed in 3.4.3.

3.3.7. Special Display Control (Line 7)

Line 7 asks if you want to change the display properties of the different types of fields in your format:

SPECIAL DISPLAY CONTROL? (1): 1 NO 2 YES

If you choose YES, you prompt the display of an optional screen called the special display screen (3.4.4). On the special display screen, you can indicate the intensity and emphasis for display constants, input and output fields, and fields where an operator has made an error. If you accept the default, NO, the default display properties are:

- Low intensity and no emphasis for display constants and output fields
- Normal intensity and no emphasis for input fields
- Blinking for fields in error

3.3.8. Error Message Field (Line 8)

You have the option of specifying where error messages from your program are displayed on your format, under what conditions they should appear, and how they look when they are displayed. To request an error message field, respond to line 8 of the characteristics screen:

DO YOU WISH TO SPECIFY AN ERROR MESSAGE FIELD? (1): 1 NO 2 YES

If you answer (2) YES, you prompt the display of the optional error message screen (3.4.5.)

3.3.9. Display Retention (Line 9)

Line 9 lets you request conditional display retention for your format. Conditional display retention means that, under certain circumstances, the contents of all variable data fields remain displayed on the screen after a workstation operator transmits the format. Input fields aren't replenished with underlines (or other stringed constant); instead, they retain the values the workstation operator keyed in. Similarly, output and bidirectional fields don't receive new values from your program. The only thing the SFC actually sends out to your format is the field display attributes (i.e., the intensity and emphasis). Conditional display retention has various uses that are discussed in more detail in 3.4.6. To request display retention for your format, respond to line 9 of the characteristics screen:

DISPLAY RETENTION ON ALL FIELDS? (1): 1 NO 2 YES

By responding with 2 (YES), you prompt the display of an optional screen called the display retention screen (3.4.6).

3.3.10. Function Keys (Line 10)

You can assign your own functions or commands to function keys 1 through 22 on the keyboard. At run time, an operator simply depresses the appropriate keys to execute the function you've assigned to it in your program. By replying (2) YES to line 10 of the characteristics screen,

FUNCTION OR COMMAND KEYS TO BE DEFINED? (1): 1 NO 2 YES

you prompt the display of the function/command key screen (3.4.7), where you select the keys you want to use, along with their run-time indicators.

3.3.11. Nondisplayed Field (Line 11)

Each format is allowed one field that communicates with your program but never appears on the workstation screen. When you enter 2 (YES) on line 11 of the characteristics screen,

DOES THIS FORMAT HAVE A NON-DISPLAYED CONSTANT? (1): 1 NO 2 YES

you prompt the display of the nondisplay screen (3.4.8), where you specify how the field will be used.

3.4. OPTIONAL SCREENS

There are seven optional screens associated with the characteristics screen. Each one gives you additional control over some run-time feature of your format.

The selections from the characteristics screen that prompt optional screens are listed in Table 3-1 and discussed in the following paragraphs. Note that whatever you specify on an optional screen affects every field in your format at run time. Some of the functions that optional screens provide, however, can be specified at the field level via a dialog screen. (Dialog screens are discussed in 3.5.) For example, you can specify display retention at the screen level as well as at the field level. However, at run time, your field level specification always overrides what you specify at the format level. For easy reference, the dialog screens that correspond to optional screens are also listed in Table 3-1.

Table 3-1. Characteristics Screen, Related Optional Screens, and Related Dialog Screens

Selection from Characteristics Screen	Resulting Optional Screen	Corresponding Dialog Screen
ERASE/UNLOCK OPTION (5): CONDITIONAL INDICATOR IN USER PROGRAM	Conditional erase/ replenish screen	
SPECIAL EDITING CHARACTERS? (2): YES	Edit screen	Dialog screen 2
SPECIAL DISPLAY CONTROL? (2): YES	Special display screen	Dialog screen 4
DO YOU WISH TO SPECIFY AN ERROR MESSAGE FIELD? (2): YES	Error message screen	
DISPLAY RETENTION ON ALL FIELDS? (2): YES	Display retention screen	Dialog screen 5
FUNCTION OR COMMAND KEYS TO BE DEFINED? (2): YES	Function command/key screen	
DOES THIS FORMAT HAVE A NON- DISPLAYED CONSTANT? (2): YES	Nondisplay screen	

If you request any optional screens, they automatically appear, in the order discussed in the following subsections, after you transmit the characteristics screen. Once you complete and transmit the optional screens you requested, you're presented with a blank screen for laying out your format (pass 1, 4.1.1).

Some general rules to keep in mind when working on optional screens are:

1. If you change your mind about using an optional screen but the screen has already come up on your workstation, just transmit the screen without entering anything on it. The SFG then accepts the default values for that option.
2. You must correct all errors (indicated by blinking or message) before you can successfully transmit an optional screen. To ignore warning messages, position the cursor at the end of the screen and press the transmit key.
3. You can initiate the HELP function at any point.
4. Rules concerning the position of the cursor are the same for optional screens as for the home screen (3.6.)

Many optional screens, as well as many dialog screens, require conditional indicators. Before we discuss optional screens, it will help you to understand what these indicators do.

3.4.1. Conditional Indicators

Conditional indicators are a means of *conditionally* controlling certain run-time features of your format. In your program, you define a set of conditions under which you want a particular feature of your format (or a single field) to change. To each condition, you assign a number that serves as an indicator. You enter this indicator number on the optional screen or dialog screen associated with the feature you want to control. At run time, your program tests to see if that condition is true. If it is, it "turns on" the indicator. The indicator acts as a flag to the SFC, telling it to alter that feature of the format or field.

You can associate a single indicator with more than one optional or dialog screen. That is, one indicator number can control several different run-time features of your format. For example, a single indicator can cause your program to display an error message, highlight a field in error, and even protect the other fields that an operator entered correctly.

There are four ways of specifying how an indicator should work:

- (Y nnn) syntax

This is how the indicator is normally used. The Y implies that some action is taken when a condition is true and your program turns an indicator on.

- (N nnn) syntax

By overwriting the Y with an N (the Y is not protected), you specify that some action is taken only when an indicator is off (that is, only when the condition in your program is false).

- (Y 000) syntax

By entering zeros instead of a number or by leaving the indicator field blank, you set up an unconditional indicator. It means that some action (like display retention) always applies. (In effect, the condition is always true.)

- (N 000) syntax

This, too, is an unconditional indicator. It means that some action (again, like display retention) never applies. (The condition is always false.) You can achieve the same effect by leaving the field blank, rather than filling in zeros.

Use of the unconditional syntax is sometimes discouraged or redundant, depending on the application. Where this is true, it is noted in the following discussions of optional screens and dialog screens.

The fields for the indicator numbers that appear on optional and dialog screens are three character spaces long. If your indicator is less than three characters, you still enter the first number at the first character slot. For example, you enter indicators 1, 12, and 123 as follows:

(Y 1__) (Y 12_) (Y 123)

NOTE:

RPG II users are limited to indicator numbers 1 to 99.

3.4.2. Conditional Erase/Replenish Screen

Remember that each time an operator transmits a format, the next step the screen format coordinator takes depends on how you responded to lines 4 and 5 of the characteristics screen when you created the format:

ERASE/UNLOCK OPTION (5): 1 NONE 2 REPLENISH SCREEN 3 ERASE SCREEN
4 UNLOCK KEYBOARD 5 CONDITIONAL INDICATOR IN USER PROGRAM

By selecting 5 (CONDITIONAL INDICATOR IN USER PROGRAM), you prompt the display of the conditional erase/replenish screen (Figure 3-3). This screen lets you specify two indicators that conditionally control your screen format's status. The two indicators you can specify are a replenish indicator and an erase indicator.

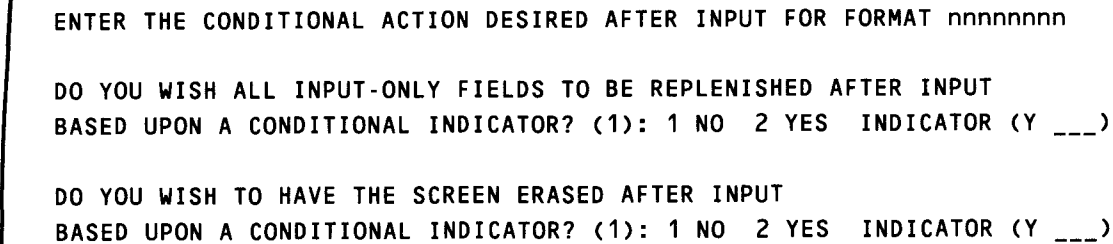
- 
1. ENTER THE CONDITIONAL ACTION DESIRED AFTER INPUT FOR FORMAT nnnnnnnn
 - 2.
 3. DO YOU WISH ALL INPUT-ONLY FIELDS TO BE REPLENISHED AFTER INPUT
 4. BASED UPON A CONDITIONAL INDICATOR? (1): 1 NO 2 YES INDICATOR (Y ___)
 - 5.
 6. DO YOU WISH TO HAVE THE SCREEN ERASED AFTER INPUT
 7. BASED UPON A CONDITIONAL INDICATOR? (1): 1 NO 2 YES INDICATOR (Y ___)

Figure 3-3. Conditional Erase/Replenish Screen

When you specify a replenish indicator, this is what happens:

■ Step 1

Your program displays a format at a workstation screen. (The format can contain any combination of output, input, and bidirectional fields.)

■ Step 2

A workstation operator fills in and then transmits your format.

■ Step 3

The SFC accepts the data from the format and performs its own validation on the input. If successful, it then passes the input to your program. The keyboard remains locked.

■ Step 4

Your program performs its own validation of the input. If and when the input meets a condition your program is testing for, it decides that only the input-only fields should be replenished. Your program then turns on your replenish indicator and signals the SFC.

■ Step 5

The SFC replenishes your input-only fields with a stringed constant (like underlines) and unlocks the keyboard. *It doesn't replenish bidirectional and output fields.* That is, these fields don't receive new output from your program. They retain the values they had when the operator transmitted your format at step 2. The associated attributes of output and bidirectional fields are rewritten, however.

When the SFC replenishes the input fields and unlocks the keyboard, the workstation operator can enter new data. This time when the operator transmits the format, the complete record that the SFC accepts and sends back to your program consists of the new data the operator just supplied, as well as the data that was retained in the output and bidirectional fields.

To enter a replenish indicator on the conditional erase/replenish screen, you must first answer the question on line 4 and then enter an indicator number:

DO YOU WISH ALL INPUT-ONLY FIELDS TO BE REPLENISHED AFTER INPUT
BASED UPON A CONDITIONAL INDICATOR? (1): 1 NO 2 YES INDICATOR (Y ___)

The erase indicator you supply on line 7 of the conditional erase/replenish screen lets your program conditionally erase the entire screen format. The erase indicator acts as a signal to the SFC to completely erase, and thus end, the display of a particular format. Typically, you use an erase indicator to signal the end of a workstation session. Like the replenish indicator, your program decides when to turn an erase indicator on, based on the input it receives from a workstation operator.

The steps described for the replenish indicator apply also to the erase indicator through step 3. At that point, the following occurs:

■ Step 4

Your program validates the input. If it meets criteria, your program turns the erase indicator on and signals the SFC.

■ Step 5

The SFC, in response, erases the workstation screen and unlocks the keyboard.

To enter an erase indicator on the conditional erase/replenish screen, you must first answer the question on line 7 and then enter an indicator number:

DO YOU WISH TO HAVE THE SCREEN ERASED AFTER INPUT
BASED UPON A CONDITIONAL INDICATOR? (1): 1 NO 2 YES INDICATOR (Y ___)

So far, we've discussed what happens when either a replenish or an erase indicator is on. The other possibility, not yet covered, is what happens when neither indicator is on. When this is the case, your format remains under the normal control of your program (as it would be if you specified NONE as your erase/unlock option). The screen format remains on the workstation screen and the keyboard remains locked until your program requests the display of the same or a new format or, perhaps, turns on other indicators that affect your format's run-time characteristics.

For a sample format that uses the conditional erase/replenish option, see 4.2.

NOTE:

Your replenish and erase indicators take precedence over any other indicators you specify. All other indicators are ignored in this type of output operation.

3.4.3. Edit Screen

The edit screen (Figure 3-4) is displayed only if you elect, via line 6 of the characteristics screen (SPECIAL EDITING CHARACTERS? (1): 1 NO 2 YES), to replace certain numeric edit characters with those of your choice.

You specify a special edit character by overwriting the character that appears in parentheses. You could, for example, overwrite the currency sign (\$) with the pound symbol (#). You must then use the special edit character (#) instead of the currency sign when completing the type attribute screen in pass 2. Thus, a numeric edited field might be displayed on the completed format as #9,999.99 instead of \$9,999.99.

Using the edit screen, you may select a character other than the underline (___) to replenish all input and bidirectional fields. You can also do this on a field-by-field basis by using dialog screen 2 (3.5.2).

```

ENTER EDIT CHARACTERS IN PARENTHESES TO BE USED FOR FORMAT nnnnnnnn
STANDARD CHAR : MEANING:                = CHAR TO BE USED
      $ : CURRENCY SIGN                  = ($)
      . : DECIMAL SYMBOL                  = (.)
      , : THOUSANDS PUNCT.                = (,)
      * : REPLACEMENT CHAR                = (*)
      CR: CREDIT ON NEGATIVE              = (CR)
      DB: DEBIT ON NEGATIVE               = (DB)
      - : REPLENISH CHAR                  = (___)
NOTE: MAKE SURE NO SYMBOL IS AMBIGUOUS WITH ANY VALID PICTURE STRING CHARACTER.

```

Figure 3-4. Edit Screen

NOTE:

For Katakana, the default currency sign on the edit screen is the yen (¥) instead of the dollar sign (\$).

Some general rules pertaining to the completion of the edit screen are:

1. Special edit characters must be the same length as the original.
2. With the exception of the replenishing character, you may not select any of the following as special edit characters: !, 9, A, X, 0 (zero), B, E, +, - (minus), Z, L, H, or the underline.
3. Special edit characters must not be ambiguous and must be different from each other.
4. If you enter an unacceptable edit character, it will be blinked. The edit screen can't be transmitted until you correct all errors.
5. If you decide you don't want to select any special edit characters, simply transmit the edit screen as is.

6. You may request help or return to the home screen by using the appropriate function keys (F13 and F1).
7. If you want your program to receive the underline character (⎵) that an operator keys in to an input or bidirectional field, you must specify a blank as the replenish character. Otherwise, if you use the underline as your replenish character, the SFC automatically translates any underlines it detects to blanks. Thus, the underline the operator keys in will never make it back to your program.

Note that if you use blanks as your replenish character, the workstation operator can't see where input fields begin and end. (No stringed constant appears after a display constant.) If the workstations you use are model 2 (UTS 40D) terminals, you can get around this restriction. The model 2 workstations let you treat underlines as a display attribute that you specify on a field-by-field basis. See dialog screen 4 (3.5.4) for details.

3.4.4. Special Display Screen

By requesting special display control on the characteristics screen, you prompt the special display screen to appear as shown in Figure 3-5.

```

1.  ENTER THE DISPLAY PROPERTIES FOR FORMAT nnnnnnnn:
2.
3.          DISPLAY CONSTANTS          (B)          (____)
4.          OUTPUT VARIABLES          (B)          (____)
5.          INPUT CAPABLE FIELDS      (A)          (____)
6.          FIELDS ENTERED IN ERROR  (A)          (5____)
7.
8.          INTENSITY:                  EMPHASIS:
9.          A. NORMAL INTENSITY         1. UNDERLINED          4. REVERSE VIDEO
10.         B. ALTERNATE (LOW INTENSITY) 2. COLUMN SEPARATORS  5. BLINK FIELD
11.         3. STRIKE THRU BARS

```

Figure 3-5. Special Display Screen

The special display screen lets you select the display properties for the four types of fields in your format: display constants, output variables, input variables (those defined as input-only or bidirectional), and fields where an operator has made an error entering data.

When the special display screen first appears, it contains the SFG defaults for intensity and emphasis (the letters A and B and the numbers 1 through 5 refer to the key that appears at the bottom of the screen). The defaults are:

- Output fields: low intensity (B), no emphasis
- Input fields: normal intensity (A), no emphasis
- Display constants: low intensity (B), no emphasis
- Fields in error: normal intensity (A), blinking emphasis (5)

You can change one or all the SFG defaults by entering your own choices. For each field, you can specify one intensity and up to five different types of emphasis. For example, you can specify that fields in error be displayed in normal intensity along with blinking, underlining, and reverse video, all at the same time. You would simply fill in line 6 of the special display screen as follows:

	INTENSITY	EMPHASIS
FIELDS ENTERED IN ERROR	(A)	(145__)

When specifying display combinations, keep in mind the operator who will be using your format. Some display combinations could result in eyestrain.

The specifications you make on the special display screen become the new defaults for your format. As you'll see in 3.5.4, you can override the format specification for individual fields via dialog screen 4.

NOTE:

Not all display properties are supported by all devices. If the type of workstation where your format is being used doesn't support the display combinations you selected, the system selects alternate display properties at run time. Appendix E lists these alternate display properties.

3.4.5. Error Message Screen

If you want the ability to display an error message from your program on a workstation screen, you must designate a field for it on your format. Through the error message screen, you can specify where and when the error message is displayed, as well as how it should look. You prompt the error message screen by responding as follows to line 8 of the characteristics screen:

DO YOU WISH TO SPECIFY AN ERROR MESSAGE FIELD? (2): 1 NO 2 YES

Figure 3-6 is a display of the error message screen as it first appears.

1.	ENTER THE FOLLOWING INFORMATION FOR YOUR ERROR MESSAGES FOR FORMAT nnnnnnnn	
2.	MESSAGE FIELD NAME IS:	(ERRORMSG)
3.	NUMBER OF LINES IN ERROR MESSAGE:	(1) ENTER 1 OR 2
4.	DISPLAY ERROR MESSAGE ON LINE NUMBER:	(LAST) ENTER NUMBER OR 'LAST'
5.	ERROR MESSAGES CONDITIONALLY DISPLAYED:	INDICATOR (Y ___)
6.	DISPLAY ATTRIBUTES: INTENSITY (A)	EMPHASIS: (____)
7.		
8.	INTENSITY:	EMPHASIS:
9.	A. NORMAL INTENSITY	1. UNDERLINED 4. REVERSE VIDEO
10.	B. ALTERNATE (LOW) INTENSITY	2. COLUMN SEPARATORS 5. BLINK FIELD
11.		3. STRIKE THRU BARS

Figure 3-6. Error Message Screen

The screen can be analyzed as follows:

■ Line 1

Displays the name of the error message field. The SFG assigns the name ERRORMSG to the error message field. If you prefer, you can enter your own, application-oriented name. It must be unique and must begin with an alphabetic character and can be a maximum of eight characters. It can contain any valid alphabetic character, the numerals 0 to 9, and the special characters \$, #, a, x, @, and ?. No embedded spaces are allowed.

■ Line 3

Asks you to designate one or two lines (60 or 120 characters) for your error message. The default is one line. Note that even if the text of your error message is less than 60 or 120 characters, you should assign either 60 or 120 characters for it in your program. This ensures that when your error message is displayed, the entire field is filled with either text or blanks. Otherwise, since the SFC expects either 60 or 120 characters, it could fill the field in with meaningless data from your work area.

■ Line 4

Displays the location of the error message field. The error message field location defaults to the last line of the screen. You can, however, specify an actual line number (or numbers for 2-line messages) indicating where you want the error message to appear. You should be careful, however, not to choose a line that will contain input or bidirectional fields. If you inadvertently do so, the screen format generator won't detect it; at run time it could result in erroneous data being returned to your program, or you'll get an error message from the SFC.

■ Line 5

Asks you for a conditional indicator. If your program detects an error when it validates input, it can turn on the error message indicator. The SFC, in response, displays the error message on your format. (You can also use the N nnn form for the indicator, where as long as the indicator is on, the error message is not displayed. When the indicator is off, it is displayed. Unconditional indicators aren't suitable for this feature.)

■ Line 6

Displays the default display properties (intensity and emphasis qualities) for the error message field. The defaults are normal intensity (A) and no emphasis. To specify your own choice of display properties, overwrite the default values, using the key at the bottom of the screen for reference. You can combine one intensity with as many as five types of emphasis.

■ Lines 9 through 11

Display a key for entering your choice of display properties. The letters A and B refer to intensities, and the numbers 1 through 5 refer to types of emphasis.

NOTES:

1. *The error message field must be the last field you define in the output record area of your program.*
2. *The error message indicator overrides any other user-specified retain options.*

When the error message indicator is on, the display of the message isn't the only thing that occurs. The SFC also retains the contents of all variable data fields on the screen and rewrites the display attributes of each field. This gives you two important advantages.

First, your program typically displays an error message when an operator enters and transmits the wrong data. By retaining the contents of variable data fields, the operator can see exactly what he entered (and what caused your program to display the error message).

Secondly, by completely rewriting the field display properties (the intensity and emphasis of each field), the SFC can highlight the field where the operator made the error. This ability to highlight a field in error involves more than what the error message screen can provide. To highlight a field, you need to specify a conditional indicator that controls its display properties. You specify such an indicator via dialog screen 4 (3.5.4), the special display properties dialog.

3.4.6. Display Retention Screen

The display retention screen (Figure 3-7) lets you set an indicator that controls whether the contents of variable data fields are retained on the workstation screen.

1. DO YOU WISH THE DISPLAY IMAGE TO BE CONDITIONALLY RETAINED
2. FOR ALL FIELDS OF FORMAT nnnnnnnn? (1): 1 NO 2 YES
- 3.
4. INDICATOR (Y ___)
- 5.
6. (IF THIS IS SELECTED, AN OUTPUT COMMAND WILL AFFECT DISPLAY ATTRIBUTES ONLY
7. UNLESS OVERRIDDEN AT FIELD LEVEL VIA DIALOG 5.)

Figure 3-7. Display Retention Screen

Display retention occurs when your program receives data from your format and, based on the nature of the data, turns the retain indicator on. In response to the retain indicator, the SFC doesn't change the text of the workstation screen. Output fields retain the values they previously received from your program; they aren't replenished with new values. Bidirectional fields retain either the value supplied by your program or the value a workstation operator keyed in over it. Input fields are left with the data the workstation operator keyed in; they aren't replenished with underlines or other stringed constant.

The only thing the SFC actually rewrites to the screen is each field's display attributes (that is, their intensity and emphasis). If the display attributes don't change, an operator won't notice a difference in the format. However, if the SFC sends out new display attributes to a field (such as blinking or reverse video), the operator will notice a very visible difference.

You can use the display retention indicator with other indicators. For example, if your program detects an error in the input it received from your operator, it can turn on two indicators: a retain indicator that preserves the data on the screen, thus allowing an operator to see what he entered, and an indicator that changes the display attributes of the field in error, such that the field stands out from the others. When the workstation operator corrects the field in error and transmits the screen, the SFC accepts the newly corrected value the operator entered, as well as the other values that were retained on the screen. (See dialog screen 4 (3.5.4) for an explanation of how to conditionally control field display attributes.)

You can also specify display retention at the field level via dialog screen 5. Note, however, that your field-level indicator always overrides the format-level indicator. This being the case, you can selectively retain all or just some of the fields on your format.

NOTE:

The first time a format is written to a workstation screen, the SFC ignores any retain indicators you specify.

To supply an indicator on the display retention screen, you must first answer the question on line 2:

DO YOU WISH THE DISPLAY IMAGE TO BE CONDITIONALLY RETAINED
FOR ALL FIELDS OF FORMAT nnnnnnnn? (2): 1 NO 2 YES

You then supply the indicator number on line 4. (If desired, you can use the unconditional Y 000 or Y _ _ _ forms of the indicator, as well as the N nnn form.)

3.4.7. Function/Command Key Screen

The function/command key screen (Figure 3-8) lets you designate which keys you assign your own program functions to. The allowable keys are function keys 1 through 22 on the workstation keyboard.

```

ENTER THE FUNCTION/COMMAND KEYS TO BE USED BY FORMAT nnnnnnnn

KEY#  ACCEPT?  RESPONSE  KEY#  ACCEPT?  RESPONSE  KEY#  ACCEPT?  RESPONSE
F1    (N)      (___)    F2    (N)      (___)    F3    (N)      (___)
F4    (N)      (___)    F5    (N)      (___)    F6    (N)      (___)
F7    (N)      (___)    F8    (N)      (___)    F9    (N)      (___)
F10   (N)      (___)    F11   (N)      (___)    F12   (N)      (___)
F13   (N)      (___)    F14   (N)      (___)    F15   (N)      (___)
F16   (N)      (___)    F17   (N)      (___)    F18   (N)      (___)
F19   (N)      (___)    F20   (N)      (___)    F21   (N)      (___)
F22   (N)      (___)

(USE FUNCTION KEY#13 FOR HELP. THIS WILL EXPLAIN INDICATOR SYNTAX.)

```

Figure 3-8. Function/Command Key Screen

There are two ways of assigning function/command keys:

- You can accept a key and assign a response indicator to it. When a workstation operator presses the key, the indicator is turned on and returned to your program, thus prompting some action to occur within the program.
- You can simply accept a key without assigning a response indicator to it. When a workstation operator presses the key, your program is alerted to take a particular course of action.

The difference between assigning indicators and not assigning indicators depends on how you code your program and the function you're using a key for. The indicators you specify for function keys are response indicators. They're different from any of the other indicators you supply to the SFG. The other indicators (called option indicators) are passed from your program to the SFC. Response indicators, on the other hand, are passed from the SFC to your program. Therefore, response indicators are handled differently in your program. Consult Section 7 for details.

Normally, when an operator presses a function key associated with a response indicator, no data is sent to your program. Instead, your program receives either blanks, zeros, or default constant values. (A bidirectional field that has been conditionally protected or conditionally displayed returns the value it received from your program.) In addition, if an operator presses a function key that hasn't been accepted and which screen format services doesn't use, an error message is sent to the workstation. The workstation operator must then acknowledge the message in system mode.

NOTES:

1. *It is recommended that you use only function keys 2, 3, 4, and 6 through 10. Function keys 1, 5, and 11 through 22 are already defined by the system. If you do reassign a function key that the system uses, you'll lose that system function. Appendix C lists the SFC function keys.*
2. *User-assigned function keys aren't supported in an IMS environment.*
3. *Response indicators shouldn't be specified for RPG II programs. RPG II has a special set of indicators (KA-KN, KP-KW) for the function keys. F1-F14 correspond to KA-KN; F15-22 correspond to KP-KW.*

3.4.8. Nondisplay Screen

The nondisplay screen (Figure 3-9) lets you define a single field that communicates with your program but never appears on the workstation screen. This can be used when there is some value or information that you don't want an operator to have access to.

1. DESCRIBE THE SINGLE NON-DISPLAYED FIELD FOR FORMAT nnnnnnnn
- 2.
3. LENGTH OF THIS FIELD IS (__)
4. VALUE OF FIELD IS (1):
5. 1 - FIRST FIELD OUTPUT BY PROGRAM (FIELD IS BIDIRECTIONAL)
6. 2 - FIELD IS INPUT-ONLY, VALUE IS:
7. -----
- 8.
9. THIS VALUE IS ALWAYS THE FIRST INPUT FIELD WHICH IS NAMED: (FLD000000)
10. IF OPTION #1 IS SELECTED ABOVE, IT IS ALSO THE FIRST OUTPUT FIELD.

Figure 3-9. Nondisplay Screen

The nondisplayed field that you define must be the first field in the input or output work area in your program. It can be an input constant associated with the format, or it may be a variable. In the latter case, this bidirectional field is communication between the output and input logic of the program. The value is not available to the workstation operator to inspect or change.

In the former case, the input-only field is a constant that enables the program to take some action based on information stored in the format rather than in the program.

The first question the nondisplay screen asks is the length of the field. The range of acceptable entries is 1 to 80. The second question the screen asks is whether the field is bidirectional or input-only. If input-only, you enter the constant value. The SFG field name (FLD00000) appears on line 9. As with any other field name, you have the option of renaming it, using an application oriented name.

3.5. DIALOG SCREENS

After you transmit your optional screens (or, if you didn't request optional screens, then after you transmit the characteristics screen), you're ready to set up your format. This involves three passes: in pass 1, you lay your format out on a blank screen; in pass 2, you assign editing attributes to variable data fields; in pass 3, you indicate field I/O direction and initiate dialog screens for the fields that require them. Dialogs, however, are also initiated automatically for any field defined with an exclamation point in pass 2 or defined with edit characters that can be protected. Section 4 explains how to complete passes 1 through 3 and how you initiate dialogs. This subsection introduces and defines what the various dialog functions are.

Dialog screens are used to place additional requirements on how information for certain variable data fields is to be handled at run time. They serve a purpose similar to the optional screens, only they affect individual fields rather than the whole format.

Some dialogs let you set up conditional indicators that control how a field looks or behaves under certain conditions. These indicators are values you define in your program and on the appropriate dialog screen. At run time, they act as a signal to your format, telling it that a field is to be treated differently from the other fields on the screen. These indicators work the same way as the indicators used on the optional screens. Refer to 3.4.1 for a brief explanation of what the indicators do.

Table 3-2 lists and briefly describes the seven dialog screens.

Table 3-2. Dialog Screens and Their Functions

Dialog Screen	Function
1	Always first dialog displayed. Indicates I/O direction, field name, internal field length, edit mask, and insertion characters and asks you if additional dialog screens are required
2	Indicates field response and default and replenishing values. For input fields only
3	Conditional display dialog. Specifies when a field will or will not appear at a workstation screen
4	Special display properties dialog. Controls the display properties of individual fields
5	Conditional retention dialog. Indicates when the contents of a field should be retained on a workstation screen
6	Conditional protection dialog. Indicates when the contents of a field should be protected
7	Range checking. Indicates what the valid range of values is for a field

After transmission of the I/O screen (pass 3), dialog screen 1 is always presented automatically for the first field requesting dialogs.

If more than one field requests dialogs, the SFG handles the fields one at a time. When dialog screen 1 and any subsequent dialogs are completed for the first field, dialog screen 1 is automatically presented for the next field and so on until all the fields requesting dialogs have been satisfied. The creation of your format is then complete, and the home screen is displayed for you to choose another function.

The following are some general rules to keep in mind when you're working with dialog screens:

1. You must correct all errors (indicated by blinking or message) before transmission of the screen can occur. To ignore warning messages, position the cursor to the end of the screen and press the transmit key.
2. You may terminate the session at any point via the F1 function key. No screen format is created, and the home screen appears for you to specify another function.
3. You can initiate the HELP function at any point during dialogs.
4. Positioning of the cursor before transmitting dialog screens is the same as described for the home screen (3.6).

3.5.1. Dialog Screen 1 (General Field Characteristics)

Dialog screen 1 (Figure 3-10) is always the first dialog screen presented for any field requiring dialogs. On it you can specify an alternate field name, the I/O direction, the internal field length, edit mask, and insertion characters. It is also where you request dialog screens 3 through 7.

```

1.  display of the field
2.
3.  (FLDnnnnn) IS FOR FIELD USE OF (1):  1  OUTPUT  2  INPUT  3  BOTH
4.  INTERNAL USAGE IS (1):  1  DISPLAY  2  PACKED  3  BINARY  4  ZONED
5.  INTERNAL LENGTH IS: (nn)
6.  PLEASE INDICATE WHETHER THE FOLLOWING ARE REQUIRED:  1  NO  2  YES
7.      CONDITIONAL DISPLAY             (1)      SPECIAL DISPLAY PROPERTIES (1)
8.      CONDITIONAL RETENTION           (1)      CONDITIONAL PROTECTION   (1)
9.      RANGE CHECKING                  (1)
10. REPLACE !'S IN THE FOLLOWING LINE BY INSERTION CHARACTERS
11.  display of the field

```

Figure 3-10. Dialog Screen 1

Each line of the format can be analyzed as follows:

■ Line 1

This line displays the field requesting dialogs. It appears the way it was defined in pass 2 (the type attribute screen). If the field is the only one on a line, it appears by itself. However, if one or more fields requesting dialogs are located on the same line with other fields, the entire line, up to and including the field that requires dialogs, is displayed.

For example, you could design your format with three fields on a single line like this:

```
NAME_____ SOCIAL SECURITY_____ AMOUNT DUE_____
```

If dialogs are requested for both the SOCIAL SECURITY and AMOUNT DUE fields, then line 1 of dialog screen 1 for the SOCIAL SECURITY field would look like this:

```
NAME A_____ SOCIAL SECURITY: 999!99!9999
```

Line 1 of dialog screen 1 for the AMOUNT DUE field would appear as:

```
NAME A_____ SOCIAL SECURITY: 999!99!9999 AMOUNT DUE: $99.999.99
```


■ Line 3:

(FLDnnnn) IS FOR FIELD USE OF (1): 1 OUTPUT 2 INPUT 3 BOTH

Shows a default name for the field (FLDnnnn) and also lets you change the I/O direction for that field from the choice made on the I/O screen.

The SFG automatically gives a name to each variable data field in the format. This name is kept in a table that can be referenced for future modification (via the modify screen for the CREATE-FROM or MODIFY functions). These names are assigned as 'FLDnnnn' sequentially, where $0 < nnnn < 256$. You can, however, assign your own logical field name by overwriting the default name in the parentheses. The logical name should be unique and should not be a duplicate of an SFG-provided name. It must start with an alphabetic character, must not contain an embedded blank, and must not exceed eight characters. The only acceptable special characters it can contain are: #, @, \$, ?, — (underline), and - (hyphen).

You specify the I/O direction for the field either by accepting the default value within the parentheses or by overwriting it with one of the other acceptable values. (If you indicate that the field is an INPUT field, the next dialog screen presented is dialog screen 2.)

■ Line 4:

INTERNAL USAGE IS (1): 1 DISPLAY 2 PACKED 3 BINARY 4 ZONED

Shows the type of internal usage. The default value for this question is normally 1. You should overwrite this value if the usage of the field within your program is anything other than display. If you specify one of the other usages for an input field, the information entered by the workstation operator is automatically converted to that usage. If you specify another usage for an output field, that information is automatically converted to display form on output. Usages specified for a bidirectional (input/output) field are handled accordingly.

The default internal usage for a field defined in pass 2 with a sign (+, -, CR, DB) is ZONED. A display field can't represent a signed number (internally).

■ Line 5:

INTERNAL LENGTH IS: (nn)

Shows the default internal length calculated by the SFG. The SFG determines internal length based on the picture string defined on the type attribute screen (!'s and other edit characters used for simple and fixed insertion are not counted) and upon the internal usage (packed, binary, zoned, or display) specified.

Depending on the internal usage specified and the internal length calculated, the SFG may generate one of the three following warning messages.

SFG40 INTERNAL LENGTH IS LESS THAN DEFAULT FOR USAGE SPECIFIED

Indicates that the internal length entered for the field is less than it should be based on the internal usage specified.

SFG41 INTERNAL LENGTH IS GREATER THAN DEFAULT FOR USAGE SPECIFIED

Indicates that the internal length entered is greater than it should be based on the internal usage specified.

SFG42 INTERNAL SIZE MAY BE INVALID. CHECK USER PROGRAM

Indicates that the internal length entered may be invalid for the internal usage specified. If, for instance, a binary field is not specified as one, two, or four bytes, this message appears.

If desired, the internal length and/or the internal usage may be changed, or the warning message may be disregarded. To disregard it, place the cursor at the end of the dialog screen and transmit.

In all cases, however, the internal size and usage provided on the dialog screen must be compatible with the data structures (data division in COBOL, input and output specifications in RPG II, etc) within the program; otherwise, an error condition may result during execution of the program.

■ Lines 6 through 9:

```
PLEASE INDICATE WHETHER THE FOLLOWING ARE REQUIRED: 1 NO 2 YES
CONDITIONAL DISPLAY      (1)      SPECIAL DISPLAY PROPERTIES (1)
CONDITIONAL RETENTION    (1)      CONDITIONAL PROTECTION     (1)
RANGE CHECKING           (1)
```

List the other dialog screens you can select for the field. If you enter 2 (YES) for any of the above, the associated dialog screen is displayed after you transmit dialog screen 1 (or after you complete and transmit dialog screen 2 if the field is an input field).

■ Lines 10 and 11

REPLACE !'S IN THE FOLLOWING LINE BY INSERTION CHARACTERS
display of the field

The contents of lines 10 and 11 vary according to the type attributes and I/O direction of the field. If exclamation marks were used to indicate that insertion characters were required, then line 10 would display the message shown in Figure 3-10 (REPLACE !'S IN THE FOLLOWING LINE BY INSERTION CHARACTERS).

For example, suppose after pass 2 you had a field defined like this:

SOCIAL SECURITY: 999!99!999

Lines 10 and 11 of dialog screen 1 for this field would then appear as follows:

REPLACE !'S IN THE FOLLOWING LINE BY INSERTION CHARACTERS
999!99!999

You would then overwrite the exclamation marks with a dash:

REPLACE !'S IN THE FOLLOWING LINE BY INSERTION CHARACTERS
999-99-999

The dashes will be protected. They aren't counted in the internal field length and can't be overwritten when the format is being used with a program at run time.

If dialogs were required for a field requiring protection for edit characters, a different message would appear on line 11. For example, if you had a field defined as:

AMOUNT DUE: \$99,999.99

Then lines 10 and 11 would appear as:

USE A '/' TO MARK PROTECTED EDIT CHARS IN THE FOLLOWING LINE
\$99,999.99

The only characters you can overwrite are the comma (,) and the decimal point (.). (The currency symbol (\$) is automatically protected.) Overwriting these characters with the slash causes that character to be protected. Protected characters act as an edit mask and are not counted in the internal length of the field.

If you choose to overwrite, the field in this example would appear as \$99/999/99. When the completed format is displayed on the workstation during its use with a program, this field appears as:

\$ _ / _ _ _ / _ _ _

NOTE:

If you specify protection for a comma in a numeric field, you must also specify protection for the decimal point and vice versa.

Table 3-3 summarizes how the contents of lines 10 and 11 vary according to the way a field is defined.

Table 3-3. Contents of Lines 10 and 11 of Dialog Screen 1

If Field Requesting Dialog	Line 10 Reads	Line 11 Displays
Is input, output, or both Contains one or more !'s Is one of the following types: 1. Alphabetic (AA!AAA) 2. Alphanumeric (AA!XXX) 3. Numeric (999!999)	REPLACE !'S IN THE FOLLOWING LINE BY INSERTION CHARACTERS	The field in question
1. Is input or both 2. Is numeric edited: \$99,999.99 \$9.99	USE A '/' TO MARK PROTECTED EDIT CHARS IN THE FOLLOWING LINE	The field in question

If the field is other than those described in Table 3-3, nothing appears on lines 10 and 11.

NOTE:

*Numeric edited fields containing any acceptable combination of the floating insertion sign \$, +, or -; the character CR or DB; or the suppression/replacement characters * or Z are always output fields and, like all other output fields, are completely protected. It is for this reason that you aren't asked to specify any edit characters to be protected. The same is true for alphanumeric fields containing B or O.*

Some general rules to keep in mind when completing dialog screen 1 are:

1. You provide responses for dialog screen 1 by overwriting the default values. If no overwriting is done, the defaults are automatically accepted.
2. Any character other than a slash used to indicate protection for a numeric edited (input or bidirectional) field is ignored.
3. Characters that can be protected in a numeric edited input or bidirectional field are the comma and the decimal point. If you specify protection for a comma, you must do so for the decimal point and vice versa.
4. You can't replace any character other than the exclamation point with an insertion character.
5. The exclamation point itself may be used as an insertion character.

3.5.2. Dialog Screen 2 (Input Field Response, Default, and Replenishing)

Dialog screen 2 lets you indicate the field response, default value, and replenishing value for an input field.

Recall from 3.3.4 that the SFC can erase input from a format so that a workstation operator can enter new data. When the input is erased, the area where the input appeared is replenished, or replaced, with underlines. Dialog screen 2 provides you with the option of choosing a character string other than underlines to replenish an input field. You may also indicate how much or how little of the field must be filled in by the operator. This is referred to as the field response. You can also specify a default value for the field. This value is returned to your program when an operator doesn't enter data in the input field. Figure 3-11 shows the format of a sample dialog screen 2 display for a field called the SOCIAL SECURITY field.

```

1. SOCIAL SECURITY: 999-99-9999
2.
3. FIELD RESPONSE (1): 1 OPTIONAL  2 REQUIRED  3 MUST-FILL  4 DEFAULT VALUE IS:
4.  ---^---^---
5.
6. INITIAL REPLENISHING VALUE IS:
7.  ---^---^---
8.
9.
10. NOTE: FOR DEFAULT, ALPHA CHARACTERS WILL BE PASSED TO PROGRAM IN UPPER OR LOWER
11. CASE AS ENTERED. FOR REPLENISH, ALPHA CHARACTERS WILL BE DISPLAYED AS ENTERED.

```

Figure 3-11. Dialog Screen 2

The explanations for the lines in Figure 3-11 are:

■ Line 1:

Simply displays the field for which the dialogs are being performed. Note that any insertion characters (such as the hyphen) that you supplied in dialog screen 1 are shown.

■ Line 3:

Permits you to overwrite the default value (1) for the field response with any of the other values listed.

OPTIONAL

Means that any or no characters may be entered for the field. If no character is entered, zero will be assumed for numeric fields and left-justified with blank padding for alphabetic and alphanumeric fields.

REQUIRED

Means that at least one character must be entered for the field. The value is right-justified with zero padding for numeric fields and left-justified with blank padding for alphabetic and alphanumeric fields.

MUST-FILL

Means that the field must be completely filled in by the workstation operator.

DEFAULT VALUE IS

If no characters are entered for the input field, a value that you supply will be used for the program. The value is defined in the following line (4) and is right-justified with zero padding for numeric fields and left-justified with blank padding for alphabetic and alphanumeric fields.

■ **Line 4**

May only be used if you select **DEFAULT VALUE IS** for the **FIELD RESPONSE**. You overwrite the underlines with the value that you want used in the program if the workstation operator fails to enter anything for that field. As the note at the bottom of the dialog screen indicates (Figure 3-11), alphabetic characters will be passed to the program exactly as they are entered here (in uppercase or lowercase). A numeric value is converted to the appropriate internal usage (display, packed, binary, zoned) just as it would be if the workstation operator entered it. Under certain circumstances, one of the following messages may be issued for an entry on line 4.

SFG51 DEFAULT VALUE HAS AN INVALID CHARACTER, OR IS WRONG TYPE

This is displayed if you enter a value that is not compatible with the external or the internal usage specified (e.g., the default value contains an alphabetic character when the field's internal usage is specified as packed or binary) or one that has an invalid construction (e.g., more than one sign or decimal point, no valid numeric digits for a numeric field). You must correct this value to continue.

SFG50 DEFAULT VALUE IS NON-NUMERIC OR ALL UNDER-LINES (WARNING)

This is displayed if you enter a nonnumeric value for a field defined externally as numeric or numeric-edited and internally as display or zoned. This situation is acceptable but it may not be intentional. If the value is correct, the message can be ignored. To ignore it, position the cursor to the bottom of the screen and then transmit.

SFG52 DEFAULT VALUE IS TOO LARGE TO BE PROCESSED

This is displayed if the value you enter for a numeric field is

1. more than 40 digits for **DISPLAY** or **ZONED** internal usage;

2. more than 15 digits for PACKED DECIMAL internal usage; or
3. not within the range $-2^{31} < \text{value} < (2^{31}-1)$ for BINARY internal usage.

■ Line 7

Displays what the replenished input field looks like. You can accept the underline as the replenishing value or specify another value by overwriting these underlines. Alphabetic characters will be displayed on the format exactly as they are entered here (in uppercase or lowercase).

It's important to note that if you specify a replenish character that doesn't match the field's data type (numeric, alphanumeric, alphabetic), it could result in error. If the operator completely fills in such a field (thus overwriting the replenish character), he can successfully transmit the format. Otherwise, the SFC returns the replenish value as part of the data and the illegal characters are rejected.

In addition, the SFC doesn't translate underlines (—) that it detects within a user-specified replenish value to blanks or zeros. Therefore, you should not specify a replenish string that *contains* an underline for numeric or alphabetic fields. If you do and an operator doesn't key over the underline with a valid character, the SFC rejects the field. Be aware, too, that if your field is alphanumeric, the SFC won't translate the underline to a blank but it will accept it as input. In this case, it could mean returning a meaningless character to your program.

3.5.3. Dialog Screen 3 (Conditional Field Display)

Dialog screen 3 is the conditional display dialog. It appears as shown in Figure 3-12.

```
display of field
SPECIFY THE CONDITIONAL INDICATOR WHICH IS TO CONTROL DISPLAY OF FIELD FLDnnnnn
(Y ___)

THE FIRST CHARACTER SHOULD BE 'Y' IF DISPLAY IS TO OCCUR WHEN THE INDICATOR
IS ON OR 'N' IF DISPLAY IS TO OCCUR WHEN INDICATOR IS OFF.

F-KEY 13 WILL PROVIDE HELP
```

Figure 3-12. Dialog Screen 3

Conditional display lets you specify an indicator that controls when a field appears at a workstation screen. At run time, your program can decide if it needs a particular field displayed. If it does, it can turn the display indicator on. If it doesn't, it turns the indicator off. (If you use the N nnn indicator syntax, then the opposite is true: display occurs when the indicator is off and is suppressed when the indicator is on.)

When a field is not displayed, only the display constant (if one is associated with the field) appears. (For example, if you had a field like

SOCIAL SECURITY _ _ _ _ _

only the SOCIAL SECURITY constant appears.) When your program turns the display of an input field off, a workstation operator is unable to enter any data to it. The value returned to your program in this case is blanks, zeros, or a default value that you specified for the field via dialog screen 2. If you turn off the display of an output or bidirectional field, your program still sends a value for it out to your format, but the SFC doesn't display it on the screen. In the case of bidirectional fields, the value your program sends to the format is returned to your program as part of the input record.

If you have an output field whose value you never want the workstation operator to see, you can unconditionally prevent its display. To do so, use the N 000 syntax for the indicator. This hides the output variable, but saves you the work of rearranging your output record.

NOTE:

If you initially prevent the display of an input field and then on the next output operation set a retain indicator for it, an operator won't be able to enter data to that field.

3.5.4. Dialog Screen 4 (Special Field Display)

Dialog screen 4 is the special display properties dialog. It lets you specify:

- *Default* display properties for a field
- *Conditional* display properties for a field that only take effect when an indicator is set

Instead of only one indicator, however, dialog screen 4 lets you specify as many as four indicators, each associated with a different intensity/emphasis combination. In addition, each display combination has a different priority level.

Figure 3-13 shows a display of dialog screen 4.


```

1.  display of field
2.  DISPLAY PROPERTIES FOR FIELD FLDnnnnn:      INTENSITY:      EMPHASIS:
3.  LOWEST PRECEDENCE: I.DEFAULT PROPERTIES      (A)            (_____)
4.  II.OPTION IND (_ ___)                        ( )            (_____)
5.  III.OPTION IND (_ ___)                       ( )            (_____)
6.  IV.OPTION IND (_ ___)                        ( )            (_____)
7.  HIGHEST PRECEDENCE:V.OPTION IND (_ ___)     ( )            (_____)
8.  INTENSITY:                                    EMPHASIS:
9.  A.NORMAL INTENSITY                          1.UNDERLINED   4.REVERSE VIDEO
10. B.ALTERNATE (LOW) INTENSITY                 2.COLUMN SEPARATORS 5.BLINK FIELD
11. C.NOT DISPLAYED                            3.STRIKE THRU BARS 6.POSITION CURSOR
12. E.SAME AS FIELDS IN ERROR

```

Figure 3-13. Dialog Screen 4

When dialog screen 4 first appears, line 3 shows the default display properties that have already been assigned for the field. This default can be one of two things:

1. The value assigned by the SFG
2. The value you assigned on the special display screen (3.4.4)

In Figure 3-13, the defaults shown are the SFG defaults for a bidirectional field:

```

DISPLAY PROPERTIES FOR FIELD FLDnnnnn:      INTENSITY:      EMPHASIS:
LOWEST PRECEDENCE:  I. DEFAULT PROPERTIES      (A)            (_____)

```

The A refers to the key that appears at the bottom of the dialog screen; it signifies normal intensity. Since the emphasis field is left blank, it means there is no special emphasis for the field. You can specify a different default for the field by overwriting the SFG default with your own choice of display properties.

Every display combination you specify can have as many as six different types of emphasis and one type of intensity. To enter an intensity, refer to the key at the bottom of the screen and then enter the letter representing the intensity you want in the column marked INTENSITY. You do the same for the emphasis by entering the numbers for the different types of emphasis in the field under EMPHASIS.

On lines 4 through 7, you can specify up to four indicators and corresponding intensity/emphasis combinations. At run time, the intensity and emphasis of the field depend on which indicator is on. The indicators are assigned priority levels. This way, if more than one indicator is on at run time, the one with the highest priority level prevails. The default always has the lowest priority, followed by II OPTION IND on up to V OPTION IND, which has the highest priority level.

Preceding each indicator number you must supply a Y or N, depending on how you want the indicator to trigger a particular display. The (Y _ _ _) syntax implies that when an indicator is on, the field appears with the intensity/emphasis associated with it. The (N _ _ _) syntax implies that a field appears with a particular

intensity/emphasis only when the indicator associated with it is off. (Unconditional entries are not permitted for this dialog.)

Lines 8 through 12 list the various intensities and types of emphasis you can choose from. Table 3-4 explains each one.

Table 3-4. Intensity/Emphasis Selections

Display Property	Function/Appearance
INTENSITY: A. NORMAL INTENSITY B. ALTERNATE (LOW) INTENSITY C. NOT DISPLAYED E. SAME AS FIELDS IN ERROR	Fields appear with normal screen intensity. Fields appear in low intensity, like display constants. Characters keyed in won't appear on the workstation screen. For example, if a field was meant for a password, an operator could enter the password without the characters actually appearing on the screen. A field appears with the same intensity as fields entered in error. This intensity is not permitted for output fields.
EMPHASIS: 1. UNDERLINED 2. COLUMN SEPARATORS 3. STRIKE THRU BARS 4. REVERSE VIDEO 5. BLINK FIELD 6. POSITION CURSOR	The underline is used as a display attribute. This is available only on UTS40D (model 2) workstations and applies only to input fields. Fields are highlighted with the column separator character (). Fields appear with strike-thru bars through the characters. Characters in a field appear black on a green video background. Field is highlighted by blinking. You can set up your format (using an indicator) so that the cursor is automatically positioned at a particular field. Normally, the cursor is positioned at the first input or bidirectional field. If the POSITION CURSOR indicator for more than one field is set, the cursor is positioned at the first of those fields. You can't specify POSITION CURSOR for output fields.

NOTE:

Not all devices support every display attribute. See Appendix E for a list of what devices support what attributes.

3.5.5. Dialog Screen 5 (Conditional Field Retention)

This is the conditional retention dialog screen. It let's you specify an indicator that controls when the contents of a field are retained on the workstation screen. When a field is retained, only the field's display attributes (intensity and emphasis) are rewritten to the workstation screen. Figure 3-14 is a display of dialog screen 5.

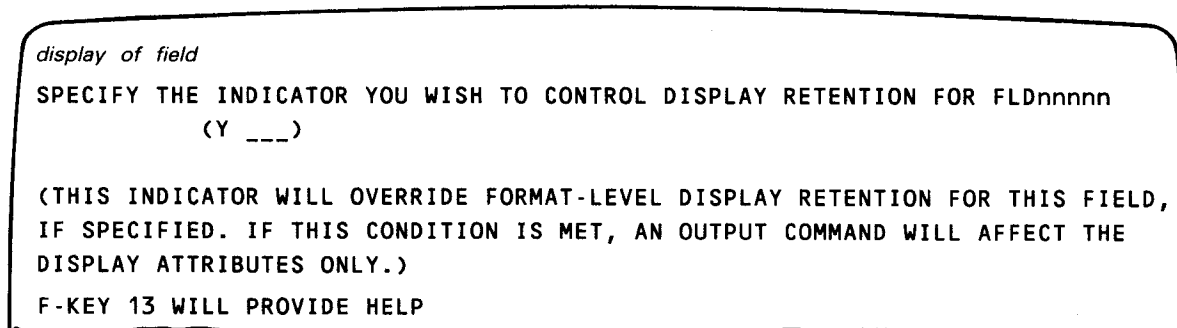


Figure 3-14. Dialog Screen 5

If your program sets a retain indicator for an output field, the field retains the value your program sent to it. A bidirectional field retains either the value your program supplied or the value an operator keyed in over your program value. An input field retains the data a workstation operator entered. The value retained in either a bidirectional or input field is the value that is ultimately returned to your program.

Since the retain operation allows the SFC to rewrite the field display attributes, you can highlight any field you retain. For example, when your program sets a field retain indicator, it can also set a special display indicator from dialog screen 4.

The retain option can also be specified at the format-level (3.4.6). However, the field-level retain indicator you specify on dialog screen 5 overrides the format-level retain indicator.

If you use the (Y _ _ _) indicator syntax, the contents of a field are retained on the workstation screen when the indicator is on. If you use the (N _ _ _) indicator syntax, the field is retained only when the indicator is off. (Use of the unconditional syntax is permitted, but not recommended.)

NOTE:

When a format is initially written to a screen, any retain indicators are ignored.

3.5.6. Dialog Screen 6 (Conditional Field Protection)

Dialog screen 6 is the conditional protection dialog. Conditional protection prevents an operator from entering data in an input or bidirectional field, if a protection indicator is set.

When a protection indicator is set, a workstation operator is unable to position the cursor at that field. The cursor simply skips over it. You control protection via the indicator you enter on dialog screen 6, shown in Figure 3-15.

display of field

SPECIFY THE INDICATOR YOU WISH TO CONTROL PROTECTION OF FIELD FLDnnnnn
(Y _ _)

IF THE FIRST CHARACTER IS 'Y', THE FIELD WILL BE PROTECTED WHEN THE INDICATOR IS ON. IF 'N' IS SPECIFIED, PROTECTION OCCURS WHEN THE INDICATOR IS OFF.

F-KEY 13 WILL PROVIDE HELP

Figure 3-15. Dialog Screen 6

You can use the Y _ _ _ indicator syntax where a field is protected when the indicator is on; or you can use the N _ _ _ indicator syntax where the field is protected when the indicator is off. (Use of the Y 000 unconditional syntax is permitted, but not the unconditional N 000 syntax.)

A protected bidirectional field returns either a program-supplied or user-supplied value to your program. A protected input field returns blanks, zeros, or a default constant (supplied on dialog screen 2). If your program sets both a retain and a protection indicator on, the value returned to your program is the retained value.

NOTES:

1. If you set a protection indicator on when a format is first displayed and on a subsequent output operation you reset the protection indicator, an SF08 error message is displayed.
2. On UNISCOPE 100/200 devices, a field can't be conditionally protected and retained.

3.5.7. Dialog Screen 7 (Input Field Range Checking)

Dialog screen 7 lets you set range check values for input and bidirectional fields. Figure 3-16 shows a dialog screen 7 display for a sample field called the SOCIAL SECURITY FIELD.

1.	SOCIAL SECURITY: 999-99-9999
2.	
3.	SPECIFY THE TYPE OF CHECK AND RANGE VALUE(S) FOR FIELD FLD000000:
4.	
5.	RANGE CHECK (1): 1 BETWEEN 2 LESS THAN 3 GREATER THAN 4 EQUAL
6.	5 NOT EQUAL 6 LESS/EQUAL 7 GREATER/EQUAL
7.	RANGE VALUE(S):
8.	---'---'----
9.	---'---'----
10.	
11.	MORE? (1): 1 NO 2 YES

Figure 3-16. Dialog Screen 7

Range checking simply means that data an operator enters in an input or bidirectional field is checked to see if it is within range (between, greater than, less than, etc) of a set of values that you specify. You can specify range checking for numeric, alphanumeric, and alphabetic fields. The range check values themselves can be numeric or alphabetic. You select the type of range check on line 5 and supply the range value on line 8 (and line 9 if you specified BETWEEN).

NOTE:

For BETWEEN range checks, you must enter two values, and the first value must be less than the second value.

The field in our example is a numeric input field defined as 999-99-9999. If BETWEEN is specified, the input value must either fall between the two values or be equal to one of the two values that you supply (overwrite) on lines 8 and 9.

You select	1 BETWEEN	on line 5.
You supply RANGE VALUES	195-44-8867	on lines 8 and 9.
	295-67-3563	

If you choose any of the other range checks, only line 8 should be overwritten with a value.

You select	LESS/EQUAL	on line 5.
You supply	195-44-8867	on line 8.

You can specify, in any combination, a maximum of 10 range checks for one field by overwriting the default value on line 11 with the value 2 (YES). Upon transmission, the screen is displayed again for the same field. You may then select another range check and supply the corresponding value.

When multiple range checks are requested, the SFG automatically performs an OR operation, which logically unites the combinations for you. Suppose you select 3 (GREATER THAN) and 5 (NOT EQUAL) as the range checks for a field. The run-time value for that field is checked, ensuring that it is either greater than or not equal to the range values you specified on line 8.

An error condition is indicated at run time if the value for a field doesn't pass the range check specified. You may make provisions within your program in the event that a workstation operator is unable to correct the error. Range checking is suppressed under the following conditions:

- An operator doesn't enter data in a field. (Even though the input value returned to your program in this case is zero or blanks and thus might not meet the range check values specified, they aren't checked.)
- An operator doesn't enter data in an input field that has a default value. The default value isn't checked.
- Data that your program sends out to a bidirectional field isn't checked.

The SFC performs range checking only after it completes other lower-level validation of input. For example, if an operator enters numeric data in an alphabetic field, the SFC detects the error before it subjects the data to a range check.

NOTE:

Lowercase to uppercase translation occurs before range checking is applied. Therefore, if you enter alphabetic range values on dialog screen 7 in lowercase and you specified lower to uppercase translation on the characteristics screen, the data an operator enters always fails.

When you transmit dialog screen 7 after specifying the tenth range check or after answering 1 (NO) to the question on line 11, dialog screen 1 is automatically redisplayed for the next field requesting dialogs. If, however, no other fields request dialog, the home screen is automatically presented for you to select another function. The creation of the format is complete.

3.5.8. Default Values for Dialog Screens

If dialogs aren't requested, certain default characteristics are automatically assumed for the field. These defaults are:

- For numeric and numeric-edited fields
 1. Field ID: SFG-provided ID is accepted.
 2. Insertion characters: None
 3. Protected edit characters: None
 4. Internal usage: Display (zoned for those fields defined with a sign (+, -, CR, DB) on the type attribute screen)
 5. Internal length: SFG-computed length is accepted.
 6. Range checking: None
 7. Conditional display: No (Field always appears.)
 8. Conditional retention: No
 9. Special display properties: No
 10. Conditional protection: No
 11. Replenishing value for I or B fields: Underlines
 12. Field response: OPTIONAL (zero for numeric fields)
 13. Fields are right-justified with zero padding.
- For alphabetic and alphanumeric edited or nonedited fields
 1. Defaults 1 - 11 for numeric fields apply to alphabetic and alphanumeric fields.
 2. Field response: OPTIONAL (blanks)
 3. Fields are left-justified with blank padding.

3.6. COMPLETING SFG SCREENS – CURSOR POSITIONING AND OVERWRITING

As soon as the SFG is activated, you get a workstation display of the home screen:

```

1.  FUNCTION (1): 1 CREATE      2 CREATE-FROM    3 MODIFY      4 DELETE
2.  5 SHOW          6 LIST          7 SPOOL       8 TERMINATE
3.  OLD FORMAT NAME (_____) IS ON THE FOLLOWING LIBRARY:
4.  FILE NAME: ($Y$FMT          ) VOLUME: (RES      )
5.
6.  NEW FORMAT NAME (_____) IS TO BE STORED ON THE FOLLOWING LIBRARY:
7.  FILE NAME: ($Y$FMT          ) VOLUME: (RES      )
8.  IF THIS FILE DOES NOT EXIST, ALLOCATE (002) CYLINDERS. INCREMENT IS (01) CYL.
9.
10. **FUNCTION KEYS ARE:F1-GO TO HOME SCREEN, F5-BREAKPOINT SPOOL FILE,
11.    F13-HELP, F14-EXIT HELP, F20-RESTORE SCREEN

```

In general, the unprotected value appearing within parentheses following any question is the default value (response) for that question. For example, on lines 4 and 7 of the home screen, the default for FILE NAME is \$Y\$FMT. You can either accept that default or overwrite it with another acceptable value.

When overwriting, you must first position the cursor over the default value. When the home screen is initially presented, the cursor appears in the upper left-hand corner of the screen. You can control its movement by using the following keys:

Cursor to HOME key

Cursor scan keys

Tab FWD or tab back keys

The results produced by using any of these keys are essentially the same. The cursor moves, but automatically stops and positions itself over the first default value. Refer to the interactive services command and facilities user guide/programmer reference for the specifics concerning control of the cursor's movement.

You can enter another acceptable value in the parentheses or keep the default value and simply move the cursor to another position. Keep in mind that the action of overwriting causes the cursor to move automatically to the next unprotected character.

NOTE:

In this manual, sample user entries on SFG screens are shown in reverse lettering (white characters on black background):

Once the home screen is completed, you transmit it by pressing the XMIT key. If, for example, the function you chose on line 1 was CREATE, the characteristics screen is the next screen that appears:

- ```

1. GLOBAL CHARACTERISTICS FOR FORMAT nnnnnnnn
2. LOWER CASE TRANSLATION? (1): 1 YES 2 NO
3. ALPHABET: (ENGLISH_) SCREEN FORMAT IS (1): 1 ORIGINAL 2 OVERLAY
4. ERASE/UNLOCK OPTION (1): 1 NONE 2 REPLENISH SCREEN 3 ERASE SCREEN
5. 4 UNLOCK KEYBOARD 5 CONDITIONAL INDICATOR IN USER PROGRAM
6. ERROR RETRY COUNT: (02) SPECIAL EDITING CHARACTERS? (1): 1 NO 2 YES
7. SPECIAL DISPLAY CONTROL? (1): 1 NO 2 YES
8. DO YOU WISH TO SPECIFY AN ERROR MESSAGE FIELD? (1): 1 NO 2 YES
9. DISPLAY RETENTION ON ALL FIELDS? (1): 1 NO 2 YES
10. FUNCTION OR COMMAND KEYS TO BE DEFINED? (1): 1 NO 2 YES
11. DOES THIS FORMAT HAVE A NON-DISPLAYED CONSTANT? (1): 1 NO 2 YES

```

The rules for positioning the cursor on the home screen also apply to the characteristics screen. You can transmit the characteristics screen whenever you finish all the necessary overwriting. For example, if you only want to change a value on line 4,

```
ERASE/UNLOCK OPTION (2): 1 NONE 2 REPLENISH SCREEN 3 ERASE SCREEN
```

you may transmit the characteristics screen immediately after you overwrite the 1 on that line. The default values for the following lines are automatically accepted. If you change all these values, the cursor must be at the bottom of the home screen before transmission. Remember, if you don't change any of the values, you simply transmit the characteristics screen after it appears.

If you make mistakes while completing any SFG screen, transmission doesn't occur. In general, the screen is redisplayed and any errors are highlighted.

**NOTE:**

*If you overwrite any of the values on a screen, before transmitting make sure that you position the cursor past the last default value you overwrote on the screen.*

These general rules for completing the home screen and characteristics screen apply to other SFG displays, such as optional and dialog screens.

If you have difficulty completing a screen at any point, you can initiate the HELP function (3.8)

For more information concerning workstation operation, refer to the interactive services commands and facilities user guide/programmer reference.



### 3.7. ERROR AND WARNING MESSAGES

There are certain instances when an entry on the home screen may be syntactically correct but violates an SFG logical rule. In these instances, blinking does not occur. Instead, one of the following messages is displayed on the workstation screen.

- SFG20 OLD FORMAT \_\_\_\_\_ DOES NOT EXIST.

The format specified as OLD FORMAT cannot be found. You should check to see if the spelling of the format's name is correct, or you might check the contents of the format library (\$Y\$FMT or a user-assigned library), via the librarian or the FSTATUS system command, to see if that format actually exists.

- SFG11 FORMAT \_\_\_\_\_ IS LARGER THAN CURRENT SCREEN.

This message indicates the format is larger than the screen size. For example, a format of 24 lines cannot be displayed and consequently cannot be modified on a workstation screen with a 12-line capacity.

These messages (SFG20 and SFG11) require corrective action. You must enter another format name before you can transmit the home screen. Messages SFG21 and SFG23 are only warning messages. Corrective action may be taken; however, no action is required.

- SFG21 NEW FORMAT \_\_\_\_\_ ALREADY EXISTS. TO REPLACE, TRANSMIT.

This message indicates that the current format name is the same as an existing format name. If you don't want the old format name replaced, change the name of the new format. If you do want the old format replaced, position the cursor to the end of the format and transmit.

- SFG22 FC TYPE ELEMENT NOT FOUND FOR FORMAT \_\_\_\_\_

This message indicates the format element type "F" exists but the "FC" type does not exist. Use the MODIFY option to create your new format using the old format name.

- SFG23 (WARNING) \_\_\_\_\_ LANGUAGE TABLE NOT FOUND IN FRMTOUT.

This message indicates that a language other than English was specified on the home screen and that the output file into which the SFG is to write the format does not have a character set translation for that language.

Again, this is a warning message only. You may go ahead and build the format (by positioning the cursor to the bottom of the screen and transmitting); however, note that a translation table must be supplied for the file before the SFC can use the format.

**■ SFG90 FUNCTION IS NOT SUPPORTED**

This message indicates the function requested on the home screen, the function key, or some other specification requests a nonexistent function or feature. Correct and rerun.

The system messages programmer/operator reference lists these messages, as well as other error messages you may get while using the SFG or the SFC.

**3.8. THE HELP FUNCTION**

In 1.3, we mentioned that an ease-of-use feature accompanying the SFG is its capability to provide prompting for any procedure with which you might be having difficulty. This capability is the HELP function.

You may request help once the home screen is displayed and anytime during the SFG session simply by pressing the function shift key in combination with the F13 function key. You then get the appropriate screen that tells you how to complete the operation for which help was requested. You can return to the point in the SFG session where you originally left off by using the F14 function key.

Let's say that you've been presented with the home screen and selected the CREATE function, but you're not quite sure about completing the rest of the screen. If you request help, the appropriate set of instructions for completing the home screen will be displayed.

The HELP function can be initiated in this manner at any point in the SFG session. Whether you're in the middle of creating a format or modifying one, you can ask the SFG for help whenever it's necessary. Appendix B contains examples of the various help screens.

## 4. Creating Screen Formats

### 4.1. CREATING A SIMPLE SCREEN FORMAT

A simple format is one where all the SFG defaults are accepted and no optional or dialog screens are required. Such a format is included here to demonstrate the basic steps needed to create a usable screen format. Once you're familiar with this procedure, it will be easier to go on to the discussion of more complex formats later in this section.

Table 4-1 shows the sequence of steps involved in creating a simple format.

Table 4-1. Basic CREATE Function Steps

| Step                                                            | Result                                                                                      |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| 1. Key in LOGON with user ID.                                   | Activates the SFG and displays the home screen                                              |
| 2. Press XMIT key.                                              |                                                                                             |
| 3. Key in RV SFGEN.                                             |                                                                                             |
| 4. Press XMIT key.                                              |                                                                                             |
| 5. Accept 1 (CREATE) from home screen.                          | Provides name of format and where it will be stored                                         |
| 6. Key in name of format.                                       |                                                                                             |
| 7. Accept library defaults.                                     |                                                                                             |
| 8. Press XMIT key.                                              |                                                                                             |
| 9. Complete pass 1 - template screen.                           | Provides actual format layout with display constants and variable data field representation |
| 10. Press XMIT key                                              |                                                                                             |
| 11. Complete pass 2 - type attribute screen.                    | Provides type attributes and editing for variable data                                      |
| 12. Press XMIT key.                                             |                                                                                             |
| 13. Complete pass 3 - I/O screen                                | Indicates direction of data flow between the program and the format                         |
| 14. Press XMIT key.                                             |                                                                                             |
| 15. Terminate (enter 8 on line 1 of) the resulting home screen. | Terminates the SFG session                                                                  |

Recall from Section 3 that the home screen appears after you activate the SFG. You complete the home screen as follows:

- Accept the default value (1) CREATE for the function on line 1.
- Provide a name for your format on line 6.
- Accept the defaults for the library on line 7.
- Transmit the home screen.

As an example, suppose you were creating a format called MYSCREEN. After you complete the home screen, but before you transmit it, it should look like this:

```

1. FUNCTION (1): 1 CREATE 2 CREATE-FROM 3 MODIFY 4 DELETE
2. 5 SHOW 6 LIST 7 SPOOL 8 TERMINATE
3. OLD FORMAT NAME (_____) IS ON THE FOLLOWING LIBRARY:
4. FILE NAME: (YFMT) VOLUME: (RES)
5.
6. NEW FORMAT NAME (MYSCREEN) IS TO BE STORED ON THE FOLLOWING LIBRARY:
7. FILE NAME: (YFMT) VOLUME: (RES)
8. IF THIS FILE DOES NOT EXIST, ALLOCATE (002) CYLINDERS. INCREMENT IS (01) CYL.
9.
10. **FUNCTION KEYS ARE:F1-GO TO HOME SCREEN, F5-BREAKPOINT SPOOL FILE,
11. F13-HELP, F14-EXIT HELP, F20-RESTORE SCREEN

```

After you transmit the home screen, the characteristics screen is automatically displayed:

```

1. GLOBAL CHARACTERISTICS FOR FORMAT MYSCREEN:
2. LOWER CASE TRANSLATION? (1): 1 YES 2 NO
3. ALPHABET: (ENGLISH_) SCREEN FORMAT IS (1): 1 ORIGINAL 2 OVERLAY
4. ERASE/UNLOCK OPTION (1): 1 NONE 2 REPLENISH SCREEN 3 ERASE SCREEN
5. 4 UNLOCK KEYBOARD 5 CONDITIONAL INDICATOR IN USER PROGRAM
6. ERROR RETRY COUNT: (02) SPECIAL EDITING CHARACTERS? (1): 1 NO 2 YES
7. SPECIAL DISPLAY CONTROL? (1): 1 NO 2 YES
8. DO YOU WISH TO SPECIFY AN ERROR MESSAGE FIELD? (1): 1 NO 2 YES
9. DISPLAY RETENTION ON ALL FIELDS? (1): 1 NO 2 YES
10. FUNCTION OR COMMAND KEYS TO BE DEFINED? (1): 1 NO 2 YES
11. DOES THIS FORMAT HAVE A NON-DISPLAYED CONSTANT? (1): 1 NO 2 YES

```

Since this is a basic format, you accept all the defaults by simply transmitting the screen.

This gives your format the following characteristics:

- Lowercase translation is used.
- English alphabet is used.
- The format is an original format, not an overlay.
- Erase/unlock option is NONE.
- The error retry count is 2.
- Your format will not use special editing characters, special display control, or an error message field.
- Your format will not retain the contents of variable data fields on the workstation screen.
- Your format will not use function keys, nor will it have a nondisplayed field.

After you transmit the characteristics screen, you're presented with a blank workstation screen to key in the format layout. This step is called pass 1.

#### 4.1.1. Pass 1 – Providing the Layout

When the blank screen is presented, you simply lay out your format by positioning the cursor to points on the screen that you choose and by keying in any display constants and underlines representing the external length of the variable data fields. The number of lines that a format can contain, as well as the maximum length of its longest line, is limited to the physical size of the workstation screen. Also, the length of a variable data field is limited to the length of one line. Continuation of a variable data field on the following line is not permitted.

Figure 4-1 illustrates what a completed template screen might look like:

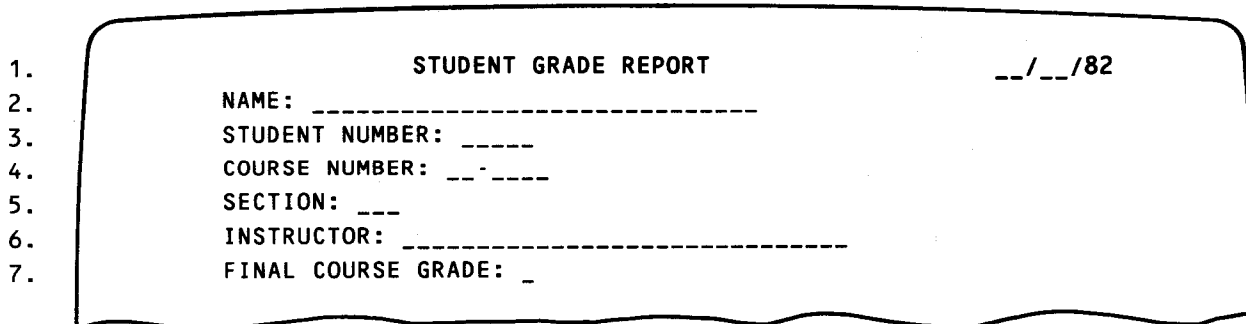


Figure 4-1. Typical Template (Pass 1) Screen

This screen can be analyzed as follows:

■ Line 1

Contains three display constants and two variable data fields. The display constants are STUDENT GRADE REPORT, the slash (/), and the /82. Both data fields are represented by two underlines (— —), indicating an external length of two character positions.

■ Line 2

Contains one display constant and one field. The display constant is NAME:, and the field is represented by 30 consecutive underlines.

■ Line 3

Contains one display constant (STUDENT NUMBER) and one variable data field (five underlines).

■ Line 4

Contains two display constants (COURSE NUMBER: and one hyphen) and two fields (two and four consecutive underlines, respectively).

■ Line 5

Contains one display constant (SECTION:) and one field (three consecutive underlines).

■ Line 6

Contains one display constant (INSTRUCTOR:) and one field (30 consecutive underlines).

■ Line 7

Contains one display constant (FINAL COURSE GRADE:) and one field (one underline).

Although completion of pass 1 is simple, there are some general notes to keep in mind.

1. The external length of a variable data field is always expressed by consecutive underlines.
2. Any character other than an underline is considered a display constant.
3. Display constants are always protected.

4. Display constants will appear on the completed format exactly the way you enter them. Thus, lowercase display constants remain in lowercase and uppercase display constants remain in uppercase. This doesn't apply to Katakana characters.
5. If you specify at least one display constant but no variable data fields, the format's creation is complete as soon as you transmit the template screen. After transmission, the home screen appears again for you to select another function.
6. If you specify at least one variable data field and press the XMIT key, that field is redisplayed for you to complete pass 2.
7. You can use the F13 function key to initiate the HELP function at any point during the completion of pass 1.
8. You can terminate pass 1 at any point by pressing the function shift in combination with the F1 key. No format is created, and the home screen is displayed for you to select another function.
9. The cursor must be at the end of the screen format you're defining before you press the XMIT key.
10. If you press XMIT before moving the cursor (before making any entries on the blank screen), the following message appears:

SFG12 NO DATA ENTERED - POSITION CURSOR TO END OF SCREEN, XMIT

If this happens, simply make the desired entries and transmit.

11. The limit for the number of variable data fields on a screen format is 255. If a greater number of variables (group of underscores) is entered, or if the format specified is too large for the generator to handle for any reason, the following message is displayed:

SFG10 NO. OF VARIABLES OR CHARACTERS IN FORMAT IS TOO LARGE.

Processing returns to the home screen.

12. If you press function key 20 during this pass, the following message appears:

SFG90 FUNCTION NOT SUPPORTED

#### **4.1.2. Pass 2 – Specifying Type Attributes and Editing**

When you transmit the finished template screen, it is automatically redisplayed for you to specify the type attributes and edit characters for all the variable data fields. The type attributes specified indicate the kind of data (numeric, alphanumeric, alphabetic) this field will receive when the format is being used to input and output data. Basically, you use COBOL-like picture clauses to define field type attributes and editing characters. Refer to Section 2 for rules governing type attributes and editing. Figure 4-2 illustrates what a completed type attribute screen looks like.

```
1. STUDENT GRADE REPORT 99/99/82
2. NAME:A _____
3. STUDENT NUMBER: 99999
4. COURSE NUMBER: XX-9999
5. SECTION: 999
6. INSTRUCTOR: A _____
7. FINAL COURSE GRADE: A
```

Figure 4-2. Typical Type Attribute (Pass 2) Screen

This screen can be analyzed as follows:

- Line 1  
Contains two fields that are defined as numeric: 99 and 99.
- Line 2  
Contains one alphabetic field. The A in the first position indicates that the entire field is alphabetic.
- Line 3  
Contains one numeric field (99999).
- Line 4  
Contains one alphanumeric field (XX) and one numeric field (9999).
- Line 5  
Contains one numeric field (999).
- Line 6  
Contains one alphabetic field. The A in the first position indicates that the entire field is alphabetic.
- Line 7  
Contains one alphabetic field (A).

Some rules that apply to the completion of the type attribute screen are:

1. The last type attribute character specified for a field is assumed for the rest of that field. For example, A\_ \_ \_ \_ denotes an entire alphabetic field.
2. An exclamation point (!) may be used in alphabetic, alphanumeric, and numeric fields to indicate that an insertion character is required. The exclamation point may not be used if any other edit characters are specified.



3. If you specified one or more special edit characters for numeric-edited fields, via the edit screen, you must use these characters on the type attribute screen. The edit screen is discussed in 3.4.3.
4. A numeric-edited field that contains any acceptable combination of the floating sign \$, +, or -; the character CR or DB; or the suppression/replacement character \* or Z is considered as an output-only field by the SFG. It is always protected and is used only to display output data on the format at run time.
5. The preceding rule also applies to alphanumeric fields containing the edit character B or zero (0).
6. You can initiate the HELP function at any time during pass 2. See Appendix B for the pass 2 help screen.
7. You can terminate pass 2 at any point by pressing the function shift key and the F1 function key. All entries for the type attribute screen and the preceding template screen are ignored, and no screen format is created. The home screen is automatically displayed for you to select another function.
8. If there are any errors on the type attribute screen, it will be redisplayed when you press the XMIT key, and the errors will be blinked.
9. Errors must be corrected before transmission of the screen can occur.
10. The cursor must be beyond the last character before you press the XMIT key.

After you transmit the type attribute screen, you are ready to proceed to pass 3.

#### 4.1.3. Pass 3 – Specifying I/O Directions and Initiating Dialogs

When you transmit the type attribute screen, the template screen is displayed for you to specify the input and output directions for each field in the format. (This is also the step where you initiate dialog screens if they're required.) Figure 4-3 shows a typical initial I/O screen display.

```

 STUDENT GRADE REPORT 0_/0_/82
NAME:0 -----
STUDENT NUMBER: 0 ----
COURSE NUMBER: 0 _-0 ---
SECTION: 0 --
INSTRUCTOR: 0 -----
FINAL COURSE GRADE: 0
```

Figure 4-3. Typical Initial I/O (Pass 3) Screen

The only difference between this and the template screen is that the letter O is displayed in the first position of each field.

The letter O signifies that the default I/O direction for the fields is output. You can overwrite this default with either an I for input or a B for both input and output, depending, of course, on the specific application of the format in a program. If you want to accept all of the defaults, simply transmit the I/O screen.

The effects of these I/O directions on the fields of the format are:

■ O – For Output

Indicates that the field will only be used to display variable data from a program at run time. Any data displayed in an output field is always protected and cannot be altered until new output data from your program overwrites it or until the entire format is erased.

■ I – For Input

Indicates that the field will only be used for variable data entered by the workstation operator. This input will be passed to the program at run time. Input fields are unprotected (except possibly for certain edit characters).

■ B – For Both (input and output, or bidirectional, fields)

Indicates that the field will be used to display variable output or to enter variable input. Bidirectional fields are also unprotected. A bidirectional field is always expected to display output the first time the format from your program is displayed at the workstation (see 1.3.2.2).

Some general rules pertaining to the completion of the I/O screen are:

1. A numeric-edited field that contains any acceptable combination of the floating sign \$, +, or -; the character CR or DB; or the suppression/replacement character \* or Z is considered a protected output field. Therefore, any numeric-edited field that was represented in this manner in pass 2 must be defined as an output field in pass 3. Specifying anything other than an O (for output), an exclamation point (for dialogs), or an underline (for dialogs) results in an error condition (blinking).
2. The preceding rule applies to alphanumeric fields containing the characters B or 0 (zero).
3. You can initiate the HELP function at any time during the completion of pass 3. See Appendix B for the pass 3 help screen.
4. You can terminate the I/O screen at any time by pressing the F1 function key. All entries for this and preceding passes are ignored, and no screen format is created. The home screen is automatically displayed for you to select another function.
5. If there are any errors in the I/O screen when you press the XMIT key, the screen is redisplayed, and the errors are blinked.

6. Any errors must be corrected before transmission of the screen can occur.
7. The cursor must be positioned after the last specification entered on the I/O screen before you press XMIT.
8. If dialogs are not requested, format generation is complete. The format is stored on the library specified on the home screen, and the home screen is redisplayed.

In many cases, completion of the I/O screen is the last step in creating a format. Using our simple example, we can review all three passes.

- Pass 1: You provide the format layout and transmit.

```

 STUDENT GRADE REPORT __/__/82
NAME: _____
STUDENT NUMBER: _____
COURSE NUMBER: ____-____
SECTION: ____
INSTRUCTOR: _____
FINAL COURSE GRADE: _

```

- Pass 2: You specify type attributes and editing and then transmit.

```

 STUDENT GRADE REPORT 99/99/82
NAME:A _____
STUDENT NUMBER: 99999
COURSE NUMBER: XX-9999
SECTION: 999
INSTRUCTOR: A _____
FINAL COURSE GRADE: A

```

- Pass 3: You specify input/output directions and transmit. For the example format, suppose all the fields are input fields. Pass 3 would look like this:

```

 STUDENT GRADE REPORT I_/I_/82
NAME:I _____
STUDENT NUMBER: I _____
COURSE NUMBER: I _-I ____
SECTION: I ____
INSTRUCTOR: I _____
FINAL COURSE GRADE: I

```

When the I/O screen is transmitted, the format is complete and the home screen is redisplayed for you to choose another function.

## 4.2. CREATING MORE COMPLEX FORMATS (USING OPTIONAL AND DIALOG SCREENS)

The format created in 4.1 is a basic format. It doesn't require additional controls over any run-time characteristics, nor does it contain any edited fields. Not all formats are that simple, however. Many formats do require edited fields and are more effective when certain features of the format are altered. This subsection illustrates how formats are created with dialog and optional screens. These additional screens provide more stringent controls over a format's run-time features.

### 4.2.1. A Sample Format

Suppose you needed a format for a program that gathers information on purchase orders. The purchase orders are from companies that regularly place an order every month. Though the companies are the same every month, the items they order and their quantities do vary.

The user program is to display a format, outputting the name and address of the company placing an order. The order entry clerk enters the date of the order, the item name, quantity, stock number, price, and the date required. These are the format's input fields.

The program must ensure that the stock number is a valid one. If valid, the ordering information from the input fields is passed back to the program and added to a file. The input fields are then replenished to permit entry of the next order date, item, stock number, etc, for that company. If the stock number is invalid, the program blinks the number and outputs an error message to the format.

When the order entry clerk has entered all the items for a particular company, he enters an item name of all zeros (0). This directs the program to display the name and address of the next company in the file.

To meet all these specifications, the format requires these optional screens:

- The conditional erase/replenish screen. A conditional indicator supplied on this screen controls when the format's input fields are replenished. Remember that if one customer orders more than one item, only the input-only fields are replenished, not the output fields that contain the customer name and address.
- The error message screen. The format needs an error message field to notify the order entry clerk that he has entered an invalid stock number.

The dialog screens needed and the fields that need them are:

■ **STOCK NUMBER**

- Dialog Screen 1

Always the first dialog screen presented. Needed to prompt subsequent dialogs, specify field direction, and provide information about the field's internal usage.

- Dialog Screen 2

Always presented for input fields that require dialogs. Needed to specify operator response and any replenishing or default values.

- Dialog Screen 4

Needed to specify special display control for this field. When an order entry clerk enters an invalid stock number, the field blinks.

■ **UNIT PRICE**

- Dialog Screen 1

This dialog is especially important for the UNIT PRICE field, because it will be an edited field. In addition to the functions noted for the STOCK NUMBER, this dialog also lets you specify protection for a decimal point and comma associated with this field.

- Dialog Screen 2

Serves the same function as dialog screen 2 for STOCK NUMBER.

The steps needed to create the format described are summarized in Table 4-2.

Table 4-2. CREATE Function Steps Using Optional and Dialog Screens (Part 1 of 2)

| Step                                   | Result                                                     |
|----------------------------------------|------------------------------------------------------------|
| 1. Key in LOGON with user ID.          | Activates the SFG and displays the home screen.            |
| 2. Press XMIT key.                     |                                                            |
| 3. Key in RV SFGEN.                    |                                                            |
| 4. Press XMIT key.                     |                                                            |
| 5. Select 1 (CREATE) from home screen. | Provides information about where the format will be stored |
| 6. Complete home screen.               |                                                            |
| 7. Press XMIT key.                     |                                                            |

Table 4-2. CREATE Function Steps Using Optional and Dialog Screens (Part 2 of 2)

| Step                                                                                              | Result                                                                                                                  |
|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 8. Complete characteristics screen                                                                | Provides information related to format as a whole; prompts optional screens                                             |
| 9. Press XMIT key.                                                                                |                                                                                                                         |
| 10. Optional: Complete any optional screens.                                                      | Provides more detailed information about format as a whole                                                              |
| 11. Press XMIT key.                                                                               |                                                                                                                         |
| 12. Complete pass 1 - template screen.                                                            | Provides actual format layout with display constants and variable data field representation                             |
| 13. Press XMIT key.                                                                               |                                                                                                                         |
| 14. Complete pass 2 - type attribute screen.                                                      | Provides type attributes and editing for variable data                                                                  |
| 15. Press XMIT key.                                                                               |                                                                                                                         |
| 16. Complete pass 3 - I/O screen.                                                                 | Indicates direction of data flow between the program and the format. May initiate dialog screens for one or more fields |
| 17. Press XMIT key.                                                                               |                                                                                                                         |
| 18. Optional: complete any dialog screens.                                                        | Provides further information about specific fields                                                                      |
| 19. Press XMIT key.                                                                               |                                                                                                                         |
| 20. Terminate (enter 8 on line 1) the resulting home screen if no further functions are required. | Terminates the SFG session                                                                                              |

#### 4.2.1.1. The Sample Home Screen and Characteristics Screen

After you activate the SFG, the home screen is displayed for you to choose a function, name your format, and indicate its library. The completed home screen for the example format is:

```

1. FUNCTION (1): 1 CREATE 2 CREATE-FROM 3 MODIFY 4 DELETE
2. 5 SHOW 6 LIST 7 SPOOL 8 TERMINATE
3. OLD FORMAT NAME (_____) IS ON THE FOLLOWING LIBRARY:
4. FILE NAME: (YFMT) VOLUME: (RES)
5.
6. NEW FORMAT NAME (MYFORMAT) IS TO BE STORED ON THE FOLLOWING LIBRARY:
7. FILE NAME: (MYFILE) VOLUME: (MYVOL1)
8. IF THIS FILE DOES NOT EXIST, ALLOCATE (5) CYLINDERS. INCREMENT IS (01) CYL.
9.
10. **FUNCTION KEYS ARE: F1-GO TO HOME SCREEN, F5-BREAKPOINT SPOOL FILE,
11. F13-HELP, F14-EXIT HELP, F20-RESTORE SCREEN

```

On line 1, you accept the default (1 CREATE). The format name (MYFORMAT) is supplied on line 6. This format won't be stored on the \$Y\$FMT file, however. Instead, it will be stored in the file MYFILE on the volume MYVOL1, as shown on line 7. Since space for MYFILE was not previously allocated, five cylinders are specified on line 8, and the INCREMENT default of 1 is accepted.

Once you transmit the home screen, the characteristics screen appears. It is completed as shown:

```

1. GLOBAL CHARACTERISTICS FOR FORMAT MYFORMAT
2. LOWER CASE TRANSLATION? (1): 1 YES 2 NO
3. ALPHABET: (ENGLISH_) SCREEN FORMAT IS (1): 1 ORIGINAL 2 OVERLAY
4. ERASE/UNLOCK OPTION (5): 1 NONE 2 REPLENISH SCREEN 3 ERASE SCREEN
5. 4 UNLOCK KEYBOARD 5 CONDITIONAL INDICATOR IN USER PROGRAM
6. ERROR RETRY COUNT: (02) SPECIAL EDITING CHARACTERS? (1): 1 NO 2 YES
7. SPECIAL DISPLAY CONTROL? (1): 1 NO 2 YES
8. DO YOU WISH TO SPECIFY AN ERROR MESSAGE FIELD? (2): 1 NO 2 YES
9. DISPLAY RETENTION ON ALL FIELDS? (1): 1 NO 2 YES
10. FUNCTION OR COMMAND KEYS TO BE DEFINED? (1): 1 NO 2 YES
11. DOES THIS FORMAT HAVE A NON-DISPLAYED CONSTANT? (1): 1 NO 2 YES

```

All the SFG defaults are accepted with two exceptions:

- Erase/unlock is by CONDITIONAL INDICATOR (line 4)
- An error message field is requested on line 8.

By specifying CONDITIONAL INDICATOR for the ERASE/UNLOCK option (line 4), you prompt the display of the conditional erase/replenish screen. Similarly, by responding with 2 for YES on line 8, you prompt the error message screen.

After you transmit the characteristics screen, the conditional erase/replenish screen is the first of the optional screens to appear.

#### 4.2.1.2. The Conditional Erase/Replenish Screen for MYFORMAT

Recall from 3.4.2 that you can specify a conditional indicator that controls when input fields are replenished. For the sample format, you want input fields replenished after an order entry clerk has filled them in. To this condition, suppose you assign an indicator number of 1. When you complete the conditional erase/replenish screen, it should appear as:

```

1. ENTER THE CONDITIONAL ACTION DESIRED AFTER INPUT FOR FORMAT MYFORMAT
2.
3. DO YOU WISH ALL INPUT-ONLY FIELDS TO BE REPLENISHED AFTER INPUT
4. BASED UPON A CONDITIONAL INDICATOR? (2): 1 NO 2 YES INDICATOR (Y 1__)
5.
6. DO YOU WISH TO HAVE THE SCREEN ERASED AFTER INPUT
7. BASED UPON A CONDITIONAL INDICATOR? (1): 1 NO 2 YES INDICATOR (Y ___)

```

### 4.2.1.3. The Error Message Screen for MYFORMAT

After you transmit the conditional erase/replenish screen, the error message screen automatically appears. (For a complete discussion of the error message screen, see 3.4.5.) To meet the needs of the sample format, you complete the error message as shown here:

- |     |                                                                            |                           |                            |
|-----|----------------------------------------------------------------------------|---------------------------|----------------------------|
| 1.  | ENTER THE FOLLOWING INFORMATION FOR YOUR ERROR MESSAGE FOR FORMAT MYFORMAT |                           |                            |
| 2.  | MESSAGE FIELD NAME IS:                                                     | (ERRORMSG)                |                            |
| 3.  | NUMBER OF LINES IN ERROR MESSAGE:                                          | (1)                       | ENTER 1 OR 2               |
| 4.  | DISPLAY ERROR MESSAGE ON LINE NUMBER:                                      | (LAST)                    | ENTER NUMBER OR 'LAST'     |
| 5.  | ERROR MESSAGES CONDITIONALLY DISPLAYED:                                    | INDICATOR (Y <u>2</u> __) |                            |
| 6.  | DISPLAY ATTRIBUTES: INTENSITY:                                             | (A)                       | EMPHASIS: ( <u>4</u> ____) |
| 7.  |                                                                            |                           |                            |
| 8.  | INTENSITY:                                                                 |                           | EMPHASIS:                  |
| 9.  | A. NORMAL INTENSITY                                                        | 1. UNDERLINED             | 4. REVERSE VIDEO           |
| 10. | B. ALTERNATE (LOW) INTENSITY                                               | 2. COLUMN SEPARATORS      | 5. BLINK FIELD             |
| 11. |                                                                            | 3. STRIKE THRU BARS       |                            |

The name of the error message field is ERRORMSG (the default). The number of lines for the message is 1 and when it appears, it is displayed on the last line of the format. The indicator number specified is 2. Therefore, when that indicator is on, the error message is displayed. (Note that the actual contents of the error message must be defined in your program.) When the error message appears, it is displayed in reverse video.

Once you complete and transmit the error message screen, you're ready to lay the format out.

### 4.2.1.4. Completing Pass 1 for MYFORMAT

The rules concerning the layout (pass 1) of the sample format are the same as those discussed in 4.1.1 for the simple format. You simply key in the display constants and indicate variable data fields with underlines. The template for MYFORMAT is:

```

 PURCHASE ORDER REPORT
CUSTOMER NAME: _____
ADDRESS: _____ STATE: __ ZIP: _____
DATE OF ORDER: __/__/__
ITEM NAME: _____
STOCK NUMBER: _____ QUANTITY: _____
UNIT PRICE: _____ DATE REQUIRED: __/__/__

```



### 4.2.1.5. Completing Pass 2 for MYFORMAT

After you transmit the template (or pass 1) screen, it is redisplayed for you to complete pass 2. In pass 2, you fill in the type attributes for the various fields. (Refer to 4.1.2 for the rules concerning pass 2.) The type attribute screen for MYFORMAT is:

1. **PURCHASE ORDER REPORT**  
2.  
3.       CUSTOMER NAME: X \_\_\_\_\_  
4.       ADDRESS: X \_\_\_\_\_ STATE: A\_ ZIP: 9 \_\_\_\_  
5.       DATE OF ORDER: 99/99/99  
6.       ITEM NAME: X \_\_\_\_\_  
7.       STOCK NUMBER: 999999               QUANTITY: 999  
8.       UNIT PRICE: \$9,999.99           DATE REQUIRED: 99/99/99

The screen can be analyzed as follows:

■ Line 3

Contains one alphanumeric field (CUSTOMER NAME).

■ Line 4

Contains three fields: one alphanumeric field (ADDRESS), one alphabetic field (STATE), and one numeric field (ZIP).

■ Line 5

Contains three numeric fields separated by slashes (DATE OF ORDER)

■ Line 6

Contains one alphanumeric field (ITEM NAME).

■ Line 7

Contains two numeric fields (STOCK NUMBER and QUANTITY).

■ Line 8

Contains four numeric fields (one for UNIT PRICE and three, separated by slashes, for DATE REQUIRED). The UNIT PRICE field is edited (\$9,999.99). Later in the creation process (dialog screen 1) you can specify protection for the comma (,) and the decimal point (.). You may do this only for fields that are going to be used for input or both input and output. Output fields are always protected.

#### 4.2.1.6. Completing Pass 3 for MYFORMAT

After you transmit the type attribute screen, the after image of pass 1 (the template screen) is again displayed for pass 3. In pass 3, you indicate the I/O direction for each field (see 4.1.3) and you initiate dialog screens for the fields that require them.

When the template screen first appears, the letter O (for output) is displayed in the first position of each field. Either you overwrite the O with the direction you want, or you accept it.

If you require dialogs, initiate them for a particular field by:

- Overwriting the O with an exclamation point (!)
- Overwriting the O with an underline

For certain fields, the SFG initiates dialogs for you automatically. These are:

- Fields containing the edit character !
- Fields containing the edit characters , or . that can be protected.

The pass 3 screen for MYFORMAT is completed as follows:

```

 PURCHASE ORDER REPORT
CUSTOMER NAME: O _____
ADDRESS: O _____ STATE: O_ ZIP: O____
DATE OF ORDER: I_/I_/I_
ITEM NAME: I _____
STOCK NUMBER: !_____ QUANTITY: I__
UNIT PRICE: I_____ DATE REQUIRED: I_/I_/I_

```

The fields CUSTOMER NAME, ADDRESS, STATE, and ZIP accept the O default because they are all output fields. The DATE OF ORDER, ITEM NAME, QUANTITY, and DATE REQUIRED fields are all defined as input fields and do not require dialogs. An exclamation point (!) in the first character position of the STOCK NUMBER field indicates that dialog screens are required for this field. Although the UNIT PRICE field is defined as an input field on the screen, dialog screens are automatically initiated for this field because it was defined as an edited field in pass 2.

Once you transmit the I/O screen, the dialog screen 1 for the first field requiring dialogs is presented. In the case of MYFORMAT, this field is the STOCK NUMBER field.

#### 4.2.1.7. Completing Dialog Screens for MYFORMAT

Recall from 3.5.1 that on dialog screen 1 you can specify

- An alternate field name
- The I/O direction of the field
- The internal field length
- An edit mask (for edited fields)
- Insertion characters
- Dialog screens 3 through 7

For the STOCK NUMBER field, dialog screen 1 is completed as follows:

|    |                                                                |
|----|----------------------------------------------------------------|
| 1. | STOCK NUMBER: 999999                                           |
| 2. |                                                                |
| 3. | (FLD00009) IS FOR FIELD USE OF (2): 1 OUTPUT 2 INPUT 3 BOTH    |
| 4. | INTERNAL USAGE IS (1): 1 DISPLAY 2 PACKED 3 BINARY 6 ZONED     |
| 5. | INTERNAL LENGTH IS (06)                                        |
| 6. | PLEASE INDICATE WHETHER THE FOLLOWING ARE REQUIRED: 1 NO 2 YES |
| 7. | CONDITIONAL DISPLAY (1) SPECIAL DISPLAY PROPERTIES (2)         |
| 8. | CONDITIONAL RETENTION (1) CONDITIONAL PROTECTION (1)           |
| 9. | RANGE CHECKING (1)                                             |

The SFG default for field name is accepted, and the I/O direction is specified as input. You accept the values displayed for internal usage and internal length. Since the specifications for MYFORMAT call for blinking the STOCK NUMBER field when an entry is invalid, SPECIAL DISPLAY PROPERTIES is selected. The STOCK NUMBER field is not an edited field, nor does it require insertion characters. Therefore, lines 10 and 11 are blank.

Since the STOCK NUMBER field is defined as an input field, dialog screen 2 automatically appears after you transmit dialog screen 1. (Refer to 3.5.2 for a complete explanation of dialog screen 2's function.)

- ```

1. STOCK NUMBER: 999999
2.
3. FIELD RESPONSE (1): 1 OPTIONAL 2 REQUIRED 3 MUST-FILL 4 DEFAULT VALUE IS:
4. -----
5.
6. INITIAL REPLENISHING VALUE IS:
7. -----
8.
9.
10. NOTE: FOR DEFAULT, ALPHA CHARACTERS WILL BE PASSED TO PROGRAM IN UPPER OR LOWER
11. CASE AS ENTERED. FOR REPLENISH, ALPHA CHARACTERS WILL BE DISPLAYED AS ENTERED.

```

For the purposes of MYFORMAT, FIELD RESPONSE must be optional. (Remember that when all the items for a particular customer have been entered, all 0's are entered in the ITEM NAME field. This signals the program to display the next customer name and address. Therefore, the field can't be a MUST-FILL field or a REQUIRED field, nor does it need a default value.) No special replenishing value is needed for this field; underlines are perfectly suitable for the application. Since all the dialog screen 2 defaults meet the field's requirements, simply transmit the screen as is.

The final dialog screen that appears for the STOCK NUMBER field is dialog screen 4, the special display properties dialog. (Refer to 3.5.4 for a discussion of its function.)

It should be completed as follows:

- ```

1. STOCK NUMBER: 999999
2. DISPLAY PROPERTIES FOR FIELD FLD00009 INTENSITY: EMPHASIS:
3. LOWEST PRECEDENCE: I. DEFAULT PROPERTIES (A) (_____)
4. II. OPTION IND (Y 2_) (B) (_____)
5. III. OPTION IND (_ _ _ _) () (_____)
6. IV. OPTION IND (_ _ _ _) () (_____)
7. HIGHEST PRECEDENCE: V. OPTION IND (_ _ _ _) () (_____)
8. INTENSITY: EMPHASIS:
9. A. NORMAL INTENSITY 1. UNDERLINED 4. REVERSE VIDEO
10. B. ALTERNATE (LOW) INTENSITY 2. COLUMN SEPARATORS 5. BLINK FIELD
11. C. NOT DISPLAYED 3. STRIKE THRU BARS 6. POSITION CURSOR
12. E. SAME AS FIELDS IN ERROR

```

On line 4, you supply the indicator number that controls when the STOCK NUMBER field blinks. This number is the same as the indicator number specified for the error message field. This means that when indicator number 2 is on, it triggers two things: the display of an error message on the error message field and blinking of the invalid entry for the STOCK NUMBER. Blinking occurs when the indicator is on because the intensity chosen for this condition is SAME AS FIELDS IN ERROR (the SFG's method of highlighting a field entered in error).

Once you transmit dialog screen 4 for the STOCK NUMBER field, dialog screen 1 automatically appears for the next field requiring dialogs. This field is the UNIT PRICE field. Dialog screen 1 is automatically initiated for this field because it is an edited field and requires an edit mask for the decimal point and comma within the field. UNIT PRICE also prompts dialog screen 2 because it is an input field. The sequence of dialogs follows:

■ Dialog Screen 1

```

1. UNIT PRICE: $9,999.99
2.
3. (FLD00011) IS FOR FIELD USE OF (2): 1 OUTPUT 2 INPUT 3 BOTH
4. INTERNAL USAGE IS (1): 1 DISPLAY 2 PACKED 3 BINARY 6 ZONED
5. INTERNAL LENGTH IS (06)
6. PLEASE INDICATE WHETHER THE FOLLOWING ARE REQUIRED: 1 NO 2 YES
7. CONDITIONAL DISPLAY (1) SPECIAL DISPLAY PROPERTIES (1)
8. CONDITIONAL RETENTION (1) CONDITIONAL PROTECTION (1)
9. RANGE CHECKING (1)
10. USE A '/' TO MARK PROTECTED EDIT CHARS IN THE FOLLOWING LINE
11. $9/999/99

```

Note that on line 11, you overwrite the comma and decimal point with a slash, indicating that these characters are to be protected.

■ Dialog Screen 2

```

1. UNIT PRICE: $9,999.99
2.
3. FIELD RESPONSE (1): 1 OPTIONAL 2 REQUIRED 3 MUST-FILL 4 DEFAULT VALUE IS:
4. $,____.____
5.
6. INITIAL REPLENISHING VALUE IS:
7. $,____.____
8.
9.
10. NOTE: FOR DEFAULT, ALPHA CHARACTERS WILL BE PASSED TO PROGRAM IN UPPER OR LOWER
11. CASE AS ENTERED. FOR REPLENISH, ALPHA CHARACTERS WILL BE DISPLAYED AS ENTERED.

```

Since the SFG defaults are appropriate for the UNIT PRICE field, dialog screen 2 is transmitted without any modifications.

When you transmit dialog screen 2 for the UNIT PRICE field, creation of MYFORMAT is complete. At this point, the home screen is displayed for you to select another function. When you run your program, the format the order entry clerk sees is:

PURCHASE ORDER REPORT

CUSTOMER NAME: \_\_\_\_\_  
ADDRESS: \_\_\_\_\_ STATE: \_\_ ZIP: \_\_\_\_\_  
DATE OF ORDER: \_\_/\_\_/\_\_  
ITEM NAME: \_\_\_\_\_  
STOCK NUMBER: \_\_\_\_\_ QUANTITY: \_\_\_\_\_  
UNIT PRICE: \$,\_\_\_\_.\_\_\_\_ DATE REQUIRED: \_\_/\_\_/\_\_

### 4.3. OVERLAY FORMATS

Recall from 3.3.3 that an overlay format physically overlays a portion of your original format. When your program requests the display of an overlay, the overlay acts as an addition or extension to your original format. You can use it to input information to or output information from your program. Overlays can cover the entire workstation screen or just a portion of it.

#### 4.3.1. The Need for an Overlay Format

The use and subsequent advantage of an overlay format are best illustrated with an example. Suppose that your program uses an original format that is displayed as:

| SALES ORDER    |                            | DATE: __/__/82 |
|----------------|----------------------------|----------------|
| CUSTOMER NAME: | _____                      |                |
| ADDRESS:       | _____                      |                |
| CITY:          | _____ STATE: __ ZIP: _____ |                |
| ITEM:          | _____                      |                |
| STOCK NUMBER:  | _____                      |                |
| COST: \$       | _,____.____                |                |
| QUANTITY:      | ____                       |                |
| SALESMAN:      | _____                      |                |

All the fields are input fields a salesman at a workstation fills in. Every time the salesman transmits a screen, your program checks the inventory levels to make sure there's enough of the ordered item in stock. If there is, the screen format is replenished with underlines and the salesman can enter the next order. If a stock item is unavailable, your program displays an overlay format. This overlay offers the salesman alternatives to the item originally requested.

For example, a salesman enters the following order:

| SALES ORDER    |                                      | DATE: 15/05/82 |
|----------------|--------------------------------------|----------------|
| CUSTOMER NAME: | John Doe Associates_                 |                |
| ADDRESS:       | 123 Main Street_____                 |                |
| CITY:          | Philadelphia___ STATE: PA ZIP: 19108 |                |
| ITEM:          | R-Widgets__                          |                |
| STOCK NUMBER:  | 1234567890                           |                |
| COST:\$        | _, 50.00                             |                |
| QUANTITY:      | 10_                                  |                |
| SALESMAN:      | John Smith_____                      |                |





When your program checks the inventory level of R-Widgets, it finds only five in stock. The quantity requested is 10. Under the circumstances, your program overlays the lower half of the screen format with an overlay format. The overlay offers the salesman a choice of other similar stock items or the option of backordering the items that are not available. The display the salesman sees is:

```

1. SALES ORDER DATE: 15/05/82
2. CUSTOMER NAME: John Doe Associates_
3. ADDRESS: 123 Main Street_____
4. CITY: Philadelphia___ STATE: PA ZIP: 19108
5. ITEM: R-Widgets__
6. ► ITEM IS NOT AVAILABLE
7. YOUR ALTERNATIVES ARE:
8. 1. A-WIDGETS 2. C-WIDGETS
9. 3. E-WIDGETS 4. BACKORDER ITEM REQUESTED
10. ENTER ALTERNATE SELECTION NUMBER_

```

Lines 6 through 10 constitute the overlay format. On the overlay, the salesman makes his choice and transmits the screen. Your program then writes the original format and it is ready for the next order.

It is important to note that only one format can be active on a workstation screen at a time. What this means is that when the overlay is displayed in the example above, a salesman can enter information only on lines 6 through 10. He can't go back and make a correction to any of the information from the original format.

Now consider another example in which two overlay formats are used.

- An input format is displayed at the workstation.

```

NAME: _____
EMPLOYEE NO.: _____

```

- The workstation operator enters information and transmits.

```

NAME: JOHN SMITH
EMPLOYEE NO.: 54123

```

- An overlay format with corresponding employee information is selected by the program.

```
NAME:JOHN SMITH
EMPLOYEE NO.:54123
DEPT #:103 DEPT. NAME:BILLING
DATE OF HIRE:72/06/10
DO YOU WISH SALARY INFO (YES, NO):_---
```

- The operator answers YES to the question on the last line and transmits. A second overlay format displays the requested information.

```
NAME:JOHN SMITH
EMPLOYEE NO.:54123
▶ CURRENT SALARY:$20000
PREVIOUS INCREASE:8% DATE:790601
GRADE:D
MIN:$15000 MID:$19000 MAX:$23000
```

Use of an overlay can be initiated or terminated, depending on the application of your program. The important thing to remember is that an overlay format, although it can look the same as the original, is a completely separate format. It must be created separately and as a result has its own format name as well as field ID's. Logically speaking, an overlay format would be created after an original format so that its eventual positioning on the workstation screen, as it relates to the original format, can be determined.

#### 4.3.2. Creation of an Overlay Format

An overlay format is created in much the same way as an original format. The steps for creation are:

1. Activate the SFG via LOGON and RV SFGEN.
2. Complete home screen. On the characteristics screen, use answer 2 (OVERLAY) on line 3 for SCREEN FORMAT IS.
3. Complete any requested optional screens.
4. Complete template, type attribute, and I/O screens.
5. Complete any requested dialogs.

The only difference between the creation of an overlay and an original format occurs in pass 1 – the template screen. If you want the overlay to replace, at some point, lines 6 to 10 of an original format, you must begin the overlay on line 6 of the template screen. You must also place the start-of-entry character (SOE ►) in the first position of that line. It indicates that the format is an overlay whose point of entry on an original format will be on line 6. The position of the cursor at transmission marks the end of the overlay format.

For example, to create an overlay for lines 6 through 10 of the following sample original format:

|    |                |             |                      |
|----|----------------|-------------|----------------------|
| 1. |                | SALES ORDER | DATE: __/__/82       |
| 2. | CUSTOMER NAME: | -----       |                      |
| 3. | ADDRESS:       | -----       |                      |
| 4. | CITY:          | -----       | STATE: __ ZIP: ----- |
| 5. | ITEM:          | -----       |                      |
| 6. | STOCK NUMBER:  | -----       |                      |
| 7. | COST: \$,      | -----       |                      |
| 8. | QUANTITY:      | ---         |                      |
| 9. | SALESMAN:      | -----       |                      |

you must complete pass 1 for the overlay format as follows:

|     |                                             |
|-----|---------------------------------------------|
| 6.  | ► ITEM IS NOT AVAILABLE                     |
| 7.  | YOUR ALTERNATIVES ARE:                      |
| 8.  | 1. A-WIDGETS    2. C-WIDGETS                |
| 9.  | 3. E-WIDGETS    4. BACKORDER ITEM REQUESTED |
| 10. | ENTER ALTERNATE SELECTION NUMBER _          |

The start-of-entry character marks the beginning of the overlay, and the cursor marks its end with respect to the original format. When the template screen is transmitted, the type attribute screen is displayed for this 2-line format. Since it is a completely independent format, the type attributes that you specify may be different from those specified for the original. The same is true of I/O directions. You may also initiate dialogs for one or both of these fields.

All overlay screens must begin with the start-of-entry character. Remember though, when you place a start-of-entry character in the first position on the screen (1,1), that position is retained even when the start-of-entry character disappears in passes 2 and 3. The first line of the generated screen always begins in position (1,2). No attempt should be made to shift back the incorrect line.

The flexibility you are afforded when creating a new or using an existing overlay format parallels that of an original. The use of one or more overlay formats for inputting and outputting can be initiated by your program. An overlay format is incorrect only if it does not physically and logically fit in with the original format to accurately reflect the requirements of the program.

#### 4.4. CREATING A FORMAT WITH DUPLICATE LINES

When creating a screen format, you may specify that certain lines be duplicated without actually entering a line twice on the template screen.

The variable data fields on any duplicate line are totally independent from the fields on the original line. Consequently, the characteristics for these fields, such as type attributes and I/O directions, may differ.

To specify duplicate lines during creation, you enter the number of the line to be duplicated in the first position (or first and second positions for a 2-digit number) of the line where the duplicate is to appear. You could, for example, lay out the template screen in pass 1 as:

|    |       |                        |                       |
|----|-------|------------------------|-----------------------|
| 1. |       | PERSONAL CREDIT REPORT | __/__/82              |
| 2. | NAME: | -----                  |                       |
| 3. | ADDR: | -----                  | STATE: ___ ZIP: ----- |
| 4. | 2     |                        |                       |
| 5. | 3     |                        |                       |
| 6. |       |                        |                       |

This screen illustrates that lines 4 and 5 are duplicates of lines 2 and 3. Upon transmission, the type attribute screen is displayed as:

|    |       |                        |                       |
|----|-------|------------------------|-----------------------|
| 1. |       | PERSONAL CREDIT REPORT | __/__/82              |
| 2. | NAME: | -----                  |                       |
| 3. | ADDR: | -----                  | STATE: ___ ZIP: ----- |
| 4. | NAME: | -----                  |                       |
| 5. | ADDR: | -----                  | STATE: ___ ZIP: ----- |
| 6. |       |                        |                       |

Since duplicate lines are independent from the originals, the type attributes selected for the fields on these lines may be the same or different from the characters selected for the originals. You could, for example, complete pass 2 like this:

|    |       |                        |                         |
|----|-------|------------------------|-------------------------|
| 1. |       | PERSONAL CREDIT REPORT | 99/99/82                |
| 2. | NAME: | A_-----                |                         |
| 3. | ADDR: | X_-----                | STATE: A__ ZIP: 9_----- |
| 4. | NAME: | A_-----                |                         |
| 5. | ADDR: | A_-----                | STATE: A__ ZIP: 9_----- |
| 6. |       |                        |                         |

Although the first field on line 3 is specified as alphanumeric, its duplicate on line 5 is alphabetic.

This same concept can be applied when completing pass 3. I/O directions for the duplicate fields can be the same as or different from those specified for the originals. It follows that you can initiate dialogs for a duplicate field but not its original and vice versa, or you can initiate dialogs for both. You might complete pass 3 in this way:

|    |         |                        |                       |
|----|---------|------------------------|-----------------------|
| 1. |         | PERSONAL CREDIT REPORT | 0_/0_/82              |
| 2. | NAME: I | -----                  |                       |
| 3. | ADDR: B | -----                  | STATE: B__ ZIP: !____ |
| 4. | NAME: O | -----                  |                       |
| 5. | ADDR: ! | -----                  | STATE: B__ ZIP: !____ |
| 6. |         |                        |                       |

The rules for specifying duplicate lines are:

1. A duplicate line number must not be entered on the first line of a format.
2. A duplicate line number entered in the first position of any line must be less than the number of the line on which it is entered.
3. A duplicate line number specified in the first position (first and second positions for a 2-digit line number) must be followed by blanks for the rest of the line.

If you violate any of these rules, the number is treated as part of the format (a display constant) when the template screen is transmitted. No blinking occurs as in other error conditions.

Assume that you lay out the template screen in pass 1 as:

|    |                    |                        |                       |
|----|--------------------|------------------------|-----------------------|
| 1. |                    | PERSONAL CREDIT REPORT | ___/___/82            |
| 2. | NAME:              | -----                  |                       |
| 3. | ADDR:              | -----                  | STATE: ___ ZIP: ----- |
| 4. | 2                  |                        |                       |
| 5. | 3                  |                        |                       |
| 6. | 2 PAST DUE AMOUNT: |                        |                       |
| 7. | 10                 |                        |                       |
| 8. | 3                  |                        |                       |
| 9. |                    |                        |                       |

Analyzing this screen, we see:

- Lines 4 and 5

Indicate that lines 2 and 3 are duplicated.

## ■ Line 6

Indicates that line 2 is not duplicated because blanks don't follow the number. The SFG recognizes this line as a display constant.

## ■ Line 7

Incorrectly specified because the number 10 is greater than the number of the line on which it appears. The number 10 is considered a display constant.

## ■ Line 8

Incorrectly specified because the first position contains a space. The number 3 is a display constant.

If these errors are not corrected before you transmit this screen, the type attribute screen is displayed as:

```
1. PERSONAL CREDIT REPORT __/__/82
2. NAME: _____
3. ADDR: _____ STATE: ___ ZIP: _____
4. NAME: _____
5. ADDR: _____ STATE: ___ ZIP: _____
6. 2 PAST DUE AMOUNT:
7. 10
8. 3
9.
```

To make corrections, you can do one of two things:

1. Return to the home screen (function key F1) and start the format over again.
2. Complete passes 2 and 3 and any dialog screens. When the home screen is displayed, select the MODIFY function (Section 5) to correct the format.

## 5. Modifying Formats

### 5.1. CREATE-FROM AND MODIFY FUNCTIONS

The CREATE-FROM and MODIFY functions are essentially the same. CREATE-FROM allows you to create a new format by changing an existing format. The old format is not altered and is saved along with the new format in `Y$FMT` or a user-assigned library file for future use.

MODIFY also allows you to create a new format by changing an existing one; however, the old format is not saved. After modification, it becomes the new format.

Among the changes you can make via the CREATE-FROM and MODIFY functions are:

- Change any format level specification from the characteristics screen.
- Change features that were defined through optional screens.
- Change the format's layout: Add fields, delete fields, and change field lengths.
- Change field type attributes.
- Change field I/O directions.
- Change field characteristics that were defined through dialogs.
- Replace lines.
- Add or insert new lines.
- Delete lines.
- Change only the format's layout (without adding or deleting fields or changing their length).

If you understand all of the steps involved in the CREATE function, the functions discussed here should present little or no difficulty. To initiate either one, you must perform the following operations:

1. Activate the SFG, using LOGON and RV SFGEN. This results in a display of the home screen:

```

1. FUNCTION (1): 1 CREATE 2 CREATE-FROM 3 MODIFY 4 DELETE
2. 5 SHOW 6 LIST 7 SPOOL 8 TERMINATE
3. OLD FORMAT NAME (_____) IS ON THE FOLLOWING LIBRARY:
4. FILE NAME: (YFMT) VOLUME: (RES)
5.
6. NEW FORMAT NAME (_____) IS TO BE STORED ON THE FOLLOWING LIBRARY:
7. FILE NAME: (YFMT) VOLUME: (RES)
8. IF THIS FILE DOES NOT EXIST, ALLOCATE (002) CYLINDERS. INCREMENT IS (01) CYL.
9.
10. **FUNCTION KEYS ARE:F1-GO TO HOME SCREEN, F5-BREAKPOINT SPOOL FILE,
11. F13-HELP, F14-EXIT HELP, F20-RESTORE SCREEN

```

2. Select the appropriate function code and overwrite the default on line 1.
3. Provide the old and new format names for CREATE-FROM and old name for MODIFY.

You can then make the appropriate changes to the old format.

### 5.1.1. Modify Screen

Specific changes to an old format are initiated through the modify screen, which is displayed as in Figure 5-1.

```

1. MODIFY OPTION (1): 1 CHANGE TEMPLATE 2 CHANGE TYPE 3 CHANGE I/O
2. 7 CHANGE TEMPLATE ONLY - NO INTERNAL CHANGES

```

Figure 5-1. Modify Screen

The modify screen (Figure 5-1) lists four options that can be used to change an existing format. You specify your selection by overwriting the default value on line 1 with any of the other acceptable values. Completing and transmitting the modify screen produces results characteristic of the option that you select.



#### 5.1.1.1. Modify Screen – Option 1 (CHANGE TEMPLATE)

The sequence of steps for this option is quite similar to those for the CREATE function. After you transmit the modify screen, the characteristics screen automatically appears for you to make changes to your format. The values that appear on the characteristics screen are the ones you specified when you created the format. You can overwrite any of the values on the screen with new values. If you originally selected options that prompt optional screens or if you initiate new optional screens, these too will appear automatically after you transmit the characteristics screen. You can change any of the values that appear on the optional screens.

After you transmit the last optional screen (or after you transmit the characteristics screen if no optional screens were requested), the after image of the template screen for the old format is displayed without protection. You can rearrange the format or change any portion of the format layout, including display constants or the length of any field. You may also delete and add fields.

When the template screen is transmitted, the type attribute screen is displayed for you to define type attributes for every field in the format – changed as well as unchanged ones. In other words, you must complete this screen as if the format were being created for the first time whether the type attribute characters are to be the same as when the format was originally created or not.

When the type attribute screen is transmitted, the I/O screen is displayed for you to complete in the same manner by specifying I/O directions for all the fields in the format, as well as initiating dialogs for any fields that require them. If one or more dialog screens were used during creation to provide certain characteristics (such as range checking, and replenishing values) for a field, you must initiate dialogs again for that field if you want these characteristics to be retained.

When all three passes and any dialog screens are completed, control returns to the home screen for you to select another function.

**NOTE:**

*If function key 20 is pressed during this option, an SFG90 (FUNCTION NOT SUPPORTED) message appears.*

#### 5.1.1.2. Modify Screen – Option 2 (CHANGE TYPE)

When you select option 2, you're first presented with the characteristics screen, where you can specify any format-level changes you wish to make. If any optional screens were requested (either when the format was created or as a result of changes to the characteristics screen), they appear after you transmit the characteristics screen. Once you've transmitted the optional screens (or the characteristics screen if no optional screens were requested), the after image of the type attribute screen for the old format is displayed.

On this display, you may only change type attribute and edit characters for any field. You may not change any display constants or delete, add, or change the length of any field. You may not rearrange the format in any way.

When this screen is transmitted, the I/O screen is displayed for you to define the I/O directions for every field in the format and to initiate dialogs as necessary. When all dialogs are completed, control returns to the home screen.

#### 5.1.1.3. Modify Screen – Option 3 (CHANGE I/O)

If you select this option, the characteristics screen appears first, just as it does for options 1 and 2. After you transmit the characteristics screen and, if requested, any optional screens, the after image of the I/O screen for the old format is displayed. You may only change the I/O directions and initiate dialogs for the fields as necessary.

#### 5.1.1.4. Modify Screen – Option 7 (CHANGE TEMPLATE ONLY – NO INTERNAL CHANGES)

If you select this option, the characteristics screen is the first to appear. Unlike options 1 through 3, however, there are certain changes you *cannot* make. This option only lets you alter the appearance of a format. It doesn't let you make any changes that would affect the program interface. The changes you can make to the characteristics screen and any optional screens that appear as a result are:

- Alphabet
- Error retry count
- Edit screen: any character
- Special display screen: any parameter
- Error message screen: number of lines in error message and any parameter *except* the indicator

You may change the format's layout by doing one or both of the following:

- Change one, all, or none of the display constants and/or their location on the template screen.
- Change the location of fields on the template screen.

In either case, you may not change the *order* of the fields, nor may you add, delete, or change the length of any field. The following is the template screen for an old format.

```

 PERSONAL CREDIT REPORT
NAME: _____
ADDR: _____
ACCOUNT NUMBER: _____
PAST DUE AMOUNT: _____
NEW BALANCE: _____ DUE DATE: __/__/__
MINIMUM PAYMENT: _____

```

Modifications might result in:

```

 PERSONAL CREDIT REPORT
NAME: _____
ADDRESS: _____
ACCOUNT NUMBER: _____ PAST DUE AMOUNT: _____
NEW BALANCE: _____ DUE DATE: __/__/__ MIN PAYMENT: _____

```

Notice that two display constants have been changed: ADDR to ADDRESS and MINIMUM PAYMENT to MIN PAYMENT. The corresponding field lengths have not been altered, nor have any other field lengths. Fields have not been added or deleted, and the order of the fields is unchanged: PAST DUE AMOUNT still follows ACCOUNT NUMBER, NEW BALANCE still follows PAST DUE AMOUNT, and so on. Each field retains the same type attributes, I/O directions, and dialog-provided characteristics that were specified when the format was originally created. In other words, there are no internal field changes.

If you inadvertently change the length of a field, the following error message is displayed:

```

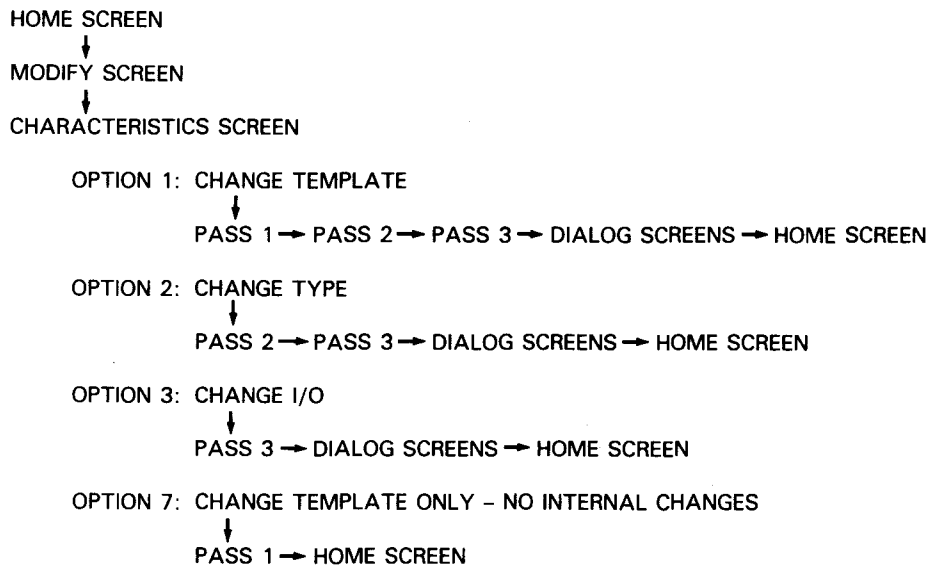
SFG30 INTERNAL FIELD HAS BEEN CHANGED.
UPDATE NOT COMPLETED.

```

You must correct the field length. Upon transmission of the template screen, whether you've rearranged it or not, modification is complete and the home screen is displayed for you to select another function.

## 5.2. MODIFICATION SUMMARY

The CREATE-FROM and MODIFY functions allow you to create a new format by changing an existing one. The method you use depends on the changes that you want to make and which of the previously described options affords the simplest means for making them. The following diagram provides a general summary of these options.



Keep in mind that you can terminate the CREATE-FROM or MODIFY functions (regardless of the option you're using to accomplish the changes) by using the F1 function key at any point before completion of the actual changes. You may also initiate the HELP function at any point except pass 1 during the CREATE-FROM or MODIFY function.

### NOTE:

*If you press XMIT before moving the cursor in pass 1 of the CREATE-FROM function, modify option 1, or modify option 7, the following message appears:*

*SFG12 NO DATA ENTERED-POSITION CURSOR TO END OF SCREEN, XMIT*

*If this happens, simply make the desired entries and press XMIT.*

## 6. Additional Functions

### 6.1. DISPLAY FUNCTIONS

Each of the three display functions – SHOW, LIST, and SPOOL – gives you a display of information about an existing format.

When you want to change a format via the MODIFY function, for example, a display of the old format is useful in determining what changes you want to make. Depending on which display function you choose, you can obtain a general format display at the workstation, a listing of the detailed field characteristics at the workstation, or a printed listing of both.

The DELETE function deletes a chosen format from the library (\$Y\$FMT or a user-assigned library, see Appendix A) and the TERMINATE function ends the SFG session.

#### 6.1.1. SHOW Function

You initiate this function by entering function code 5 on line 1 of the home screen and by entering the old format name on line 3. When the home screen is transmitted, you get a workstation display of the existing format with its field type attributes. Input-only fields will contain their replenish values to distinguish them from bidirectional fields.

You may use the TAB key to help you identify the protected fields from unprotected fields. The cursor automatically stops at the beginning of each unprotected field each time the TAB key is pressed. The low-intensity display of protected fields also helps you to make the distinction.

#### 6.1.2. LIST Function

This function is initiated by entering function code 6 on line 1 of the home screen. Its purpose is to provide comprehensive information about a format as a whole and about the individual fields in a format. The information the LIST function provides includes:

- A summary of format-oriented information from the characteristics screen
- Details from any optional screens originally requested for the format

- The type attributes and characteristics of all fields
- Details from any dialog screens originally requested for any field
- Details about the input and output data structures showing the data
- Interface to the user program

The LIST function provides all this information by displaying a series of screens at your workstation.

When you transmit the home screen, a summary screen is displayed for you. This screen contains a synopsis of what was specified on the characteristics screen when the format was created. When you transmit the summary screen, the next screen (or series of screens) displayed depends on how the format was created.

If the format incorporated any of these options:

- Special editing characters
- Special display control
- An error message field
- Function/command keys

then the next series of screens will include one or more of these displays:

- A special editing display
- A display control and error message display (This is one screen.)
- A function/command key display

Following transmission of these screens, you get a display of the format as it was created. If none of the above options were selected for the format, then this display of the format follows immediately after you transmit the summary screen.

After you transmit the display of the format, the next screen to appear is the list screen. This screen displays detailed characteristics of the fields in the format, such as field ID, I/O directions, and external lengths. At this time, you may also request sublist screens and conditional values screens. Sublist screens provide any field response requirements, default values, range check values, and replenishing values that were specified for any fields via dialogs. Conditional values screens display any conditional features and conditional indicators that were specified for fields, also through dialogs.

Finally, input record screens display the internal location of input fields, and output record screens display the internal location of output fields.

**NOTES:**

1. When you use the *LIST* function, it is displayed in two parts if your format consists of more than 12 lines.
2. If your format is using line 12 or 24, any characters in columns 79 and 80 are not displayed.
3. The preceding notes do not apply to either the *SHOW* or *SPOOL* functions nor do they affect the detailed output of the *LIST* function.

**6.1.2.1. Summary Screen**

Figure 6-1 shows a typical display of the first screen presented for the *LIST* function for an old format.

```
1. NAME: FORMAT01 FORMAT SIZE: 07 X 64 DATE: 05/25/82 CREATED FROM:
2. ALPHA: ENGLISH FILE: YFMT ON RES GEN8
3. LINE ORIGIN: 01,01 OVERLAY: NO ERASE/UNLOCK BY: CONDITIONAL INDICATOR
4. LOWER CASE TRANSLATION: YES NUMBER OF VARIABLES: 0014 ERROR RETRY COUNT:02
5. SPECIAL EDITING CHARACTERS:YES SPECIAL DISPLAY CONTROL: YES
6. ERROR MESSAGE FIELD SPECIFIED: YES DISPLAY RETENTION: YES
7. FUNCTION OR COMMAND KEYS DEFINED: YES
8. FORMAT HAS A NON-DISPLAY CONSTANT: NO
9.
10. INPUT ONLY FIELDS TO BE REPLENISHED: INDICATOR IS Y006
11.
```

Figure 6-1. Summary Screen

Analyzing this screen, we see:

■ Line 1

Displays the format's name (FORMAT01), its size (7 lines, the longest of which is 64 characters), its creation date (05/25/82), and the name of the format from which this format was created (if one exists).

■ Line 2

Displays the language or character set (ENGLISH) that was used, the name of the screen format file (\$Y\$FMT), the volume (RES) containing that file, and your release level (GEN8, indicating release 8.0).

■ Line 3

Displays the row and column number where the format begins on the screen, whether the format is an overlay format (NO), and how the format is erased and the keyboard unlocked (by CONDITIONAL INDICATOR).

- Line 4

Displays whether lowercase translation was specified (YES), the number of variable fields in the format (14), and the error retry count (02).

- Line 5

Displays whether special editing characters were specified (YES) and whether special display control was requested (YES). When special editing is YES, it prompts the appearance of the special editing display. When display control is YES, it prompts the appearance of the display control/error message display. These screens appear after you transmit the summary screen.

- Line 6

Displays whether an error message field was specified (YES) and whether display retention was specified (NO). When YES is specified for the error message field, it prompts the appearance of the display control and error message display.

- Line 7

Displays whether function/command keys were specified (YES). When YES is specified, it prompts the appearance of the function/command key display.

- Line 8

Displays whether the format has a nondisplay constant (NO). When YES, the length of the constant is also displayed on line 8, and the value of the constant is displayed on line 9.

- Line 10

Displays the conditional indicator that controls whether input fields are replenished after input (as it was specified on the conditional erase/replenish screen).

### 6.1.2.2. Special Editing Display

This display is a facsimile of the original edit screen (3.4.3). It lists the user-defined characters, as well as the standard characters, and is shown in Figure 6-2.

| NAME: FORMAT01    FORMAT SIZE: 07 X 64    DATE: 05/25/82 |                    |                   |
|----------------------------------------------------------|--------------------|-------------------|
| SPECIAL EDITING CHARACTERS: YES                          |                    |                   |
| STANDARD CHAR:                                           | MEANING            | = CHAR TO BE USED |
| \$:                                                      | CURRENCY SYMBOL    | = (\$)            |
| .:                                                       | DECIMAL POINT      | = (/)             |
| ,:                                                       | THOUSANDS PUNCT.   | = (,)             |
| *:                                                       | REPLACEMENT CHAR   | = (*)             |
| CR:                                                      | CREDIT ON NEGATIVE | = (CC)            |
| DB:                                                      | DEBIT ON NEGATIVE  | = (DD)            |
| -:                                                       | REPLENISH CHAR.    | = (-)             |

Figure 6-2. Special Editing Display



### 6.1.2.3. Display Control and Error Message Display

This display summarizes what was originally specified on two optional screens: the special display control screen and the error message screen. It lists the display properties for display constants, output variables, input capable variables, and fields entered in error. It also lists the name of the error message field (if one was specified), along with the number of lines in the message and the line number where the message appears. A typical display control and error message display is shown in Figure 6-3.

```

NAME: FORMAT01 FORMAT SIZE: 07 X 64 DATE: 05/25/82
DISPLAY CONSTANTS: (B)(____) OUTPUT VARIABLES
INPUT CAPABLE VARIABLES: (A)(____) FIELDS ENTERED IN ERROR:(A)(1____)
ERROR MESSAGE DISPLAYED ON LINE NUMBER : LAST
ERROR MESSAGE FIELD NAME IS: ERRORMSG MESSAGE DISPLAY ATTRIBUTES:(B)(____)
ERROR MESSAGE CONDITIONALLY DISPLAYED BASED ON INDICATOR (Y 004)

INTENSITY: EMPHASIS

A.NORMAL INTENSITY 1.UNDERLINED 4.REVERSE VIDEO
B.ALTERNATE (LOW) INTENSITY 2.COLUMN SEPARATORS 5.BLINK FIELD
 3.STRIKE THRU BARS

```

Figure 6-3. Display Control and Error Message Display

### 6.1.2.4. Function/Command Key Display

This display is a facsimile of the function/command key screen. It shows key numbers and, if defined, the corresponding response for all possible function/command keys.

Figure 6-4 shows a sample display of the function/command key display.

```

NAME: FORMAT01 FORMAT SIZE: 07 X 64 DATE: 05/25/82
FUNCTION OR COMMAND KEYS DEFINED: YES

KEY# ACCEPT? RESPONSE KEY# ACCEPT? RESPONSE KEY# ACCEPT? RESPONSE
F1 (N) (000) F2 (Y) (001) F3 (Y) (065)
F4 (Y) (003) F5 (Y) (000) F6 (N) (000)
F7 (N) (000) F8 (N) (000) F9 (N) (000)
F10 (N) (000) F11 (N) (000) F12 (N) (000)
F13 (N) (000) F14 (N) (000) F15 (N) (000)
F16 (N) (000) F17 (N) (000) F18 (Y) (025)
F19 (N) (000) F20 (Y) (174) F21 (N) (000)
F22 (N) (000)

```

Figure 6-4. Function/Command Key Display

### 6.1.2.5. The Format Display

This display looks like the after image of pass 2, when the screen was created. It shows the type attributes for every field in the format. Figure 6-5 shows a typical format display.

```

1. PERSONAL CREDIT REPORT 99/99/82
2. NAME:AAAAAAAAAAAAAAAAAAAAAAAAAAAA
3. ADDR:XXXXXXXXXXXXXXXXXXXXXXXXXXXX STATE:AAA ZIP:99999
4. SOCIAL SECURITY:999!99!9999
5. ACCOUNT NUMBER:XXX-999999
6. AMOUNT DUE:$99.999.99
7. BALANCE:$$,$99.99 PAYMENT DUE DATE:99/99/99

```

Figure 6-5. Screen Format

### 6.1.2.6. List Screen

Figure 6-6 shows the next screen to be presented for the LIST function.

```

1. NAME: FORMAT01 FORMAT SIZE: 7 X 64 DATE: 05/25/82
2. FIELD-ID (X,Y) I/O EXTRNAL INTRNAL RESPN REPLN RANGE COND
3. TYP LNGL TYP BYT DIGIT
4. FLD00001 1,55 0 N 2 D 2 2. M -
5. FLD00002 1,58 I N 2 D 2 2. (M) (M) (M)
6. NAME 2,7 I A 30 D 30 30. R -
7. ADDR 3,7 B X 30 D 30 30. R -
8. STATE 3,46 B A 4 D 4 4. R -
9. FLD00006 3,56 B N 5 P 3 5. 0 - ()

```

MORE? (Y)

Figure 6-6. List Screen

Line 1 of the list screen (Figure 6-6) shows the format's name (FORMAT01), its size (7 lines, the longest of which is 64 characters), and its creation date (05/25/82). Line 2 displays the headings for the particular field characteristics listed in each column. The headings and corresponding column contents are:

■ FIELD - ID

This column contains the names of the particular format fields. These are listed in the order defined during creation. In this example, SFG-provided names, as well as user-assigned names, are listed.

- (Y, X)

This column displays the Y, X coordinates (or the physical position) of each field. The Y coordinate is the line on which the field is located, and the X coordinate indicates the position of that field on the line. The field whose ID is FLD00001, for example, is located on line 1 and starts at the 55th character position on that line.

- I/O

This column displays the I/O usage for the fields in a format, where I is for input, O is for output, and B is for both (input and/or output). Refer to 4.1.3 for a detailed discussion of I/O directions.

- EXTRNAL

This column indicates the external display type and the external display length for the format fields. The N 2 in this column means that field FLD00001 is a numeric field with an external length of 2 (underlines). The abbreviations for the external field types are:

| <u>Abbreviation</u> | <u>Meaning</u>            |
|---------------------|---------------------------|
| N                   | Fixed numeric field       |
| A                   | Alphabetic field          |
| X                   | Alphanumeric field        |
| NE                  | Numeric edited field      |
| AE                  | Alphabetic edited field   |
| XE                  | Alphanumeric edited field |

- INTRNAL

This column shows the internal usage of the field and the internal length. The D 2 for the first field means that its internal usage is display and its internal length is two bytes. The P 3 for the last field means that its internal usage is packed and its internal length is three bytes. Refer to 3.5.2 for a discussion of internal length and usage. The abbreviations for the internal usage types are:

| <u>Abbreviation</u> | <u>Meaning</u> |
|---------------------|----------------|
| D                   | Display        |
| B                   | Binary         |
| P                   | Packed decimal |
| Z                   | Zoned decimal  |

## ■ RESPN

This column lists the field response requirements for input or bidirectional fields, where:

M means MUST FILL.

O means OPTIONAL.

R means REQUIRED.

( ) means DEFAULT.

Refer to 3.5.2 for an explanation of field response values.

## ■ REPLN

This column indicates replenishing values for input or bidirectional fields. A single character in the column shows the replenish value for the screen (either an underline or the character selected on the EDIT screen). A set of parentheses indicates that a unique replenish value has been specified for this field. Refer to 3.5.2 for an explanation of replenishing.

## ■ RANGE

This column indicates whether range checking has been requested for a field. The presence of a set of parentheses indicates that range checking has been requested. Refer to 3.5.8 for an explanation of range checking.

## ■ COND

This column indicates whether any conditional indicators were specified to control certain features of the field. A set of parentheses in this column signifies that indicators are used.

At the bottom of the screen, the question MORE? is displayed. If you want a listing of the same type of information for more fields in the format, accept the default value (Y) and transmit. If you don't want any further listings, overwrite the default value with an N and transmit.

### NOTE:

*If any character other than a Y is entered for this question, the SFG assumes N and control is passed to the next screen.*

Through the information presented on this list screen alone, you can tell a great deal about the format without even having created it.

### 6.1.2.7. Sublist Screen

The sublist screen, if initiated, displays the values that were specified (during creation via dialogs) for field response, range checking, and replenishing. Recall that a set of parentheses in the RESPN, REPLN, or RANGE column of the list screen indicates that specific values, other than SFG default values, have been assigned for that field. To find out what these values are, you simply enter a Y in any of these parentheses, depending on which, if any, of these values you want to know. Upon transmission of the list screen (either the last in an entire series of list screens or the one in which you answer N to the question MORE?), the sublist screens are presented.

Let's say that you make the shaded entries on the list screen:

| NAME: FORMAT01      FORMAT SIZE: 7 X 64      DATE: 05/25/82 |          |       |     |         |         |     |     |       |       |       |           |
|-------------------------------------------------------------|----------|-------|-----|---------|---------|-----|-----|-------|-------|-------|-----------|
| 1.                                                          | FIELD-ID | (X,Y) | I/O | EXTRNAL | INTRNAL |     |     | RESPN | REPLN | RANGE | COND      |
| 2.                                                          |          |       |     | TYP     | LNGH    | TYP | BYT | DIGIT |       |       |           |
| 3.                                                          | FLD00001 | 1,55  | O   | N 2     | D 2     | 2.  | M   | -     | ( )   |       |           |
| 4.                                                          | FLD00002 | 1,58  | I   | N 2     | D 2     | 2.  | (M) | (M)   | (M)   |       |           |
| 5.                                                          | NAME     | 2,7   | I   | A 30    | D 30    | 30. | R   | -     |       |       |           |
| 6.                                                          | ADDR     | 3,7   | B   | X 30    | D 30    | 30. | R   | -     |       |       |           |
| 7.                                                          | STATE    | 3,46  | B   | A 4     | D 4     | 4.  | R   | -     |       |       |           |
| 8.                                                          | FLD00006 | 3,56  | B   | N 5     | P 3     | 5.  | O   | -     | ( )   |       |           |
| 9.                                                          |          |       |     |         |         |     |     |       |       |       | MORE? (N) |

Upon transmission of the list screen, the sublist screen (Figure 6-7) is presented for FLD00002:

| 1.  | FIELD-ID       | (Y,X) | I/O | EXTRNAL | INTRNAL |     |     | RESPN | REPLN | RANGE |
|-----|----------------|-------|-----|---------|---------|-----|-----|-------|-------|-------|
| 2.  |                |       |     | TYP     | LNGH    | TYP | BYT | DIGIT |       |       |
| 3.  | FLD00002       | 1,58  | I   | N 2     | D 2     | 2.  | (Y) | (Y)   | (Y)   |       |
| 4.  | DEFAULT VALUE: |       |     |         |         |     |     |       |       |       |
| 5.  | 01             |       |     |         |         |     |     |       |       |       |
| 6.  | REPLENISHING:  |       |     |         |         |     |     |       |       |       |
| 7.  | 99             |       |     |         |         |     |     |       |       |       |
| 8.  | RANGE:         |       |     |         |         |     |     |       |       |       |
| 9.  | BTW            | 01-04 |     |         |         |     |     |       |       |       |
| 10. | BTW            | 06-08 |     |         |         |     |     |       |       |       |
| 11. | GT             | 13    |     |         |         |     |     |       |       |       |

Figure 6-7. Sublist Screen

Analyzing this screen, we see:

- Lines 1 and 2

Display the same headings as lines 2 and 3 of the list screen.

- Line 3

Is a duplicate of line 4 of the list screen.

- Lines 4 and 5

Show that the value 1 was chosen by the format's creator (via dialog screen 2) as the default response (field RSP) for field FLD00002.

- Lines 6 and 7

Show that the replenishing value for the field is 9's instead of underlines.

- Lines 8-11

Show the types of range checks and the corresponding range check values that were selected for field FLD00002 via dialog screen 7. The abbreviations for the range check types are:

| <u>Abbreviation</u> | <u>Meaning</u>        |
|---------------------|-----------------------|
| BTW                 | Between               |
| NEQ                 | Not equal             |
| EQU                 | Equal                 |
| GT                  | Greater than          |
| LT                  | Less than             |
| GE                  | Greater than or equal |
| LE                  | Less than or equal    |

Refer to Section 3 for an explanation of field responses, range checking, and replenishing.

Upon transmission of this sublist screen, another sublist screen with the requested values is displayed for the next field requesting it and so on until all fields are taken care of. If there are no further sublist screens to be displayed, input and output record screens are displayed.

### 6.1.2.8. Conditional Values Screen

This screen, if initiated, displays which features of a field are conditionally controlled and the indicators that control them. You prompt the display of the conditional values screen in much the same way that you prompt the sublist screen. Recall that a set of parentheses in the COND column of the list screen means that conditional indicators were specified for a field. By keying a Y in the parentheses on the list screen, you prompt the display of the conditional values screen.

Figure 6-8 shows a typical conditional values screen display.

| 1.  | FIELD-ID                                                                              | (X,Y) | I/O | EXTRNAL | INTRNAL | RESPN | REPLN | RANGE | COND    |
|-----|---------------------------------------------------------------------------------------|-------|-----|---------|---------|-------|-------|-------|---------|
| 2.  | TYP LNCH TYP BYT DIGIT                                                                |       |     |         |         |       |       |       |         |
| 3.  |                                                                                       |       |     |         |         |       |       |       |         |
| 4.  | FLD00006                                                                              | 3,56  | B   | N       | 5       | P     | 3     | 5     | 0 - (Y) |
| 5.  | FIELD IS TO BE DISPLAYED IF INDICATOR 009 IS ON.                                      |       |     |         |         |       |       |       |         |
| 6.  | FIELD IS TO BE RETAINED.                                                              |       |     |         |         |       |       |       |         |
| 7.  | FIELD IS TO BE PROTECTED IF INDICATOR 002 IS OFF.                                     |       |     |         |         |       |       |       |         |
| 8.  | CURSOR IS EXPLICITLY POSITIONED TO THIS FIELD                                         |       |     |         |         |       |       |       |         |
| 9.  |                                                                                       |       |     |         |         |       |       |       |         |
| 10. |                                                                                       |       |     |         |         |       |       |       |         |
| 11. | DISPLAY PROPERTIES ARE NORMAL INTENSITY WITH REVERSE VIDEO<br>IF INDICATOR 014 IS ON. |       |     |         |         |       |       |       |         |

Figure 6-8. Conditional Values Screen

Analyzing this screen, we see:

- Lines 1 and 2

Display the same headings as line 2 of the list screen.

- Line 4

Is a duplicate of line 4 of the list screen.

- Line 6

Shows that the field (FLD00006) is displayed when indicator 009 is on. (This indicator would have been originally specified on dialog screen 3 when the format was created.)

- Line 7

Shows that the contents of the field are retained whenever indicator 005 is on. (This indicator would have been specified on dialog screen 5.)

- Line 8

Shows that the field is protected when indicator 002 is off. (Originally specified via dialog screen 6)

- Line 9

Shows that the cursor is always positioned at this field. (As specified via dialog screen 4)

- Line 11

Shows the display properties associated with this field when indicator 014 is on. (As specified via dialog screen 4)

To display the conditional values screen for the next field you requested, simply press the XMIT key. Upon transmission of the last conditional value screens, you're presented with the input and output record screens.

#### 6.1.2.9. Input Record Screen

The input record screen displays the internal displacement in bytes for a particular field name in the input record. You may receive an input record screen upon transmission of a sublist screen or list screen if no sublist screen exists. Figure 6-9 contains an input record screen.

|    | INPUT RECORD |              |
|----|--------------|--------------|
|    | FIELD NAME   | DISPLACEMENT |
| 1. |              |              |
| 2. |              |              |
| 3. | FLD00002     | 0            |
| 4. | NAME         | 2            |
| 5. | ADDRESS      | 32           |
| 6. | STATE        | 62           |
| 7. | FLD00006     | 66           |

Figure 6-9. Input Record Screen

Analyzing this screen, we see fields FLD00002, NAME, ADDRESS, STATE, and FLD00006 exist in the input record and that FLD00002 has an internal location of 0, NAME has an internal location of 2, ADDRESS has an internal location of 32, STATE has an internal location of 62, and FLD00006 has an internal location of 66.

#### 6.1.2.10. Output Record Screen

The output record screen displays the internal displacement in bytes for a particular field name in the output record. You receive an output record screen upon transmission of the last input record screen. Figure 6-10 contains an output record screen.



|    | OUTPUT RECORD |              |
|----|---------------|--------------|
|    | FIELD NAME    | DISPLACEMENT |
| 1. |               |              |
| 2. |               |              |
| 3. | FLD00001      | 0            |
| 4. | ADDR          | 2            |
| 5. | STATE         | 32           |
| 6. | FLD00006      | 36           |

Figure 6-10. Output Record Screen

Analyzing this screen, we see the fields FLD00001, ADDR, STATE, and FLD00006 exist in the output record and that FLD00001 has an internal location of 0, ADDR has an internal location of 2, STATE has an internal location of 32, and FLD00006 has an internal location of 36. When the last output record screen is transmitted, control returns to the home screen.

**NOTE:**

*A field specified as bidirectional (both input and output) appears on both the input and output record screens.*

### 6.1.3. SPOOL Function

This function is initiated by entering the function code 7 on line 1 of the home screen and by supplying the old format's name on line 3 of the home screen. The SPOOL function is similar to the LIST function, except your screen formats are printed rather than displayed and responses are not required. All information regarding your format is always printed. After a job terminates, all screen formats are printed. However, if you want to have any screen formats printed before termination, you can breakpoint the spool file by pressing function key 5.

## 6.2. DELETE AND TERMINATE FUNCTIONS

The DELETE function allows you to delete a format from its library file. It is initiated by entering the function code 4 on line 1 and the old format's name on line 3 of the home screen. Upon transmission of the home screen, the format is displayed at the workstation along with the following question. Note that the underlines will be replaced by the name of the format to be deleted.

IS \_\_\_\_\_ THE SCREEN FORMAT YOU WISH TO DELETE? Y OR N? ( )

If you're sure that you want to delete the displayed format, enter Y and then press the XMIT key. If you decide not to, enter N and then press the XMIT key.

**NOTE:**

*There is no error indication for an incorrect response. If the response to this question is anything other than Y (YES), then N (NO) is automatically assumed and the format is not deleted.*

As mentioned in 3.2.1, the TERMINATE function ends the SFG session.

## 7. Associating Formats with a Program: Using the Screen Format Coordinator (SFC)

### 7.1. PROGRAMMER RESPONSIBILITIES

Now that you know the many functional capabilities that the SFG gives you to create and maintain screen formats, let's see how screen formats are incorporated into a program.

As mentioned in 1.2 and 1.4, the SFC has such responsibilities as retrieving the formats that the program is going to use, checking certain data, and controlling the flow of data between the program and the format. Although it isn't necessary for you to be concerned with the specifics of the SFC's activities, there are still certain responsibilities that you have in making the actual connection between screen formats and the program. Among these are making sure that the SFC knows where to retrieve the format, providing the proper job control statements for the format's file, including the appropriate references to the formats within the program, and incorporating any indicators that your format relies on.

### 7.2. JOB CONTROL

To execute a program that uses screen formats, you are responsible for providing the proper job control statements. These statements are basically the same as they would be for any program that's executed at a workstation.

You must provide a device assignment set or DVC/LFD sequence for every file that the program uses, including the workstation file, and, if the format is stored on a location other than the `$Y$FMT` library (see A.1.), the format's library file. In addition to the standard job control language (JCL), you must indicate within the DVC/LFD sequence for the workstation that screen format services are to be provided. You do this via the `// USE SFS` statement.

The format for the USE statement is:

```
//[symbol] USE SFS [, { [format-file-lfd-1]/[format-file-lfd-2] }]
 [, { SY$FMT
 format-file-lfd
 }]
[, initial-screen] [, { nnn }]
 [, { 1 }]
[, screen-format-1=alias-1, ..., screen-format-12=alias-12]
```

As you can see, most of the parameters in this statement are optional, so that if your format is stored on \$Y\$FMT the following alone is sufficient to indicate the use of screen format services:

```
// USE SFS
```

With this statement included in the DVC/LFD sequence for the workstation file, your program can retrieve any format from \$Y\$FMT simply by issuing the proper calling statement (peculiar to the program's language) and the name of the format being called. The optional parameters may be used as necessary and are explained in detail in A.1.

The job control stream to execute a program using screen formats must always include a DVC/LFD sequence, which contains the USE statement, for the workstation. Also, note that when two or more workstations access formats in the same screen format file, the DVC, USE, and LFD job control statements (with a unique LFD) must be provided for each workstation. And that, when the workstation accesses only one screen format file residing on RES, no DVC/LFD sequence is required for the file.

**NOTE:**

*If more than one screen format file resides on RES but not in \$Y\$FMT, a DVC/LFD sequence (with a unique LFD) must be provided for the file in order for the workstation to access it.*

## Example 1:

```

// JOB MYJOB
.
.
.
// DVC 50
// VOL D00029 } DVC/LFD sequence for the user library
// LBL FORMAT } with a unique LFD
// LFD FMT
// DVC 200
// USE SFS,FMT } DVC/LFD sequence for
// LFD WRKSTN } the workstation
.
.
.
// EXEC JOB01

```

This example shows the job control statements required for a single workstation to access a single format on a user library. There are two DVC/LFD sequences in this example: one indicating the user library where the format resides (on disk volume D00029), and one indicating a workstation, where the device code number is 200 and the name that the program uses to reference it is WRKSTN. Note that the DVC/LFD sequence for the user library provides a unique LFD for the format. That LFD is specified in the USE statement to indicate which format is to be used.

## Example 2:

```

// JOB PAYROL
.
.
.
// DVC 200
// USE SFS } DVC/LFD sequence for
// LFD WRKSTN } the workstation
.
.
.
// EXEC PRSNL

```

This example shows the job control statements required for a single workstation to access a single format that resides in the screen format library \$Y\$FMT on RES. It contains a DVC/LFD sequence for the workstation, where the device code number is 200 and the name that the program uses to reference the workstation is WRKSTN. The format to be used is on \$Y\$FMT; therefore, no DVC/LFD sequence is given for the file.

## Example 3:

```
// DVC 50
// VOL 123456
// LBL FORMAT
// LFD FMT1
// DVC RES
// LBL YFMT
// LFD FMT2
// DVC 200 (2)
// USE SFS,FMT1/FMT2
// LFD WRKSTN
```

This example indicates formats required by your program are located in two format libraries. When your program requests a screen format, the file labeled FORMAT is searched first. If the requested format is not located, the secondary format file is searched.

For more information concerning the USE SFS statement and for more specific job control examples, see A.1. For general job control information, refer to the job control user guide.

## 7.3. PROGRAM CONSIDERATIONS

In addition to issuing the proper job control statements, you must make certain specifications within your program to indicate that a screen format is being used. These specifications are relatively simple and specifically related to the language in which the program is written.

### 7.3.1. RPG II

Considerations for an RPG II program using screen formats relate to the following specifications forms: file, input, and output.

- File Specifications Form

On this form, you must specify the workstation terminal as a primary file or a demand file. The block length is always omitted, and the device name (columns 40-46) must be entered as WORKSTN. You must also specify other files used by the program (excluding the format library file); however, you may not designate any of these as secondary files if the workstation file is the primary file.

- Input Format Specifications Form

RPG II initializes the workstation program by generating a blank record. This record must be anticipated on the input format specifications form.

Every variable data field with an I/O direction of input or both input and output (bidirectional) must be identified on the input format specifications form. The field names need not correspond to those that were assigned to the actual format fields during generation.

The field location entries on the RPG II form refer to the position of these fields with respect to the input data rather than to the position of the input fields on the screen. This is simply because data is either read from the format (from input or bidirectional fields) or written to the format (to output or bidirectional fields) automatically - according to the sequence in which the variable data fields were described when the format was created.

Field lengths are counted sequentially. The FROM entry for the first input or bidirectional field is always 1, and the TO entry is always the number of the last underline in the field as defined in the format. The FROM entry for the next field picks up where the TO entry for the previous field left off. The underlines are counted in relation to this number to get the TO entry.

Let's say that the first input or bidirectional field in the format has a length of 10, and the second has a length of 12. The FROM and TO entries for the first field are 1 and 10. Entries for the second field are therefore 11 and 22. Output fields are not counted at all. If an output field is located between two input or bidirectional fields, simply bypass it.

Control levels, matching fields, and look-ahead fields may not be used with workstation input.

#### ■ Output Format Specifications Form

Every variable data field with an I/O direction of output or bidirectional must be identified on the output format specifications form. This includes any bidirectional field that was defined on the input format specifications form.

The END POSITION entries refer to the end positions within the output data and not to their positions on the screen. For example, if there are two output or bidirectional fields with lengths of 10 and 12 respectively, the end position for the first field is 10, and that of the second field is 22.

The name of the format, as it was specified on the home screen, must be entered in the CONSTANT OR EDIT WORD field (columns 45-52) and enclosed in apostrophes. The length of the format name must be entered in column 43 with the letter K preceding it (column 42).

Use of a first-page indicator is not permitted.

Figure 7-1 gives examples of screen formats and the accompanying input/output specifications that must be made for each format.





**PERSONAL CREDIT REPORT**

NAME: [REDACTED]

ADDR: [REDACTED] STATE: [REDACTED] ZIP: [REDACTED]

30 32 37

SOCIAL SECURITY: [REDACTED]

PAST DUE AMOUNT: [REDACTED]

47

PAYMENT: [REDACTED]

NEW BALANCE: [REDACTED] PAYMENT DUE DATE: [REDACTED] / [REDACTED] / [REDACTED]

57 59 61 63

- [REDACTED] Input
- [REDACTED] Both
- [REDACTED] Output

c. Display 2



RPG II  
**OUTPUT FORMAT SPECIFICATIONS**

PROGRAM \_\_\_\_\_ PROGRAMMER \_\_\_\_\_ DATE \_\_\_\_\_ PAGE \_\_\_\_\_ OF \_\_\_\_\_ PAGES

| PAGE NO | FORM TYPE | LINE NO | FILE NAME | STACKER SELECT/<br>F-FETCH OVERFLOW |   |   | SPACE | SKIP | OUTPUT INDICATORS |       |        |       |   |     | FIELD NAME | DATA FORMAT<br>P/B/L/R | END POSITION<br>IN<br>OUTPUT<br>RECORD | CONSTANT OR EDIT WORD | NOT USED | PROGRAM IDENTIFICATION |
|---------|-----------|---------|-----------|-------------------------------------|---|---|-------|------|-------------------|-------|--------|-------|---|-----|------------|------------------------|----------------------------------------|-----------------------|----------|------------------------|
|         |           |         |           | A                                   | N | D |       |      | BEFORE            | AFTER | BEFORE | AFTER | N | NOT |            |                        |                                        |                       |          |                        |
| 1       | 0         | 01      | WORKIST.N |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 02      |           |                                     |   |   |       |      |                   |       |        |       |   |     | K8         |                        |                                        |                       |          |                        |
| 1       | 0         | 03      |           |                                     |   |   |       |      |                   |       |        |       |   |     | 30         |                        |                                        |                       |          |                        |
| 1       | 0         | 04      |           |                                     |   |   |       |      |                   |       |        |       |   |     | 32         |                        |                                        |                       |          |                        |
| 1       | 0         | 05      |           |                                     |   |   |       |      |                   |       |        |       |   |     | 37         |                        |                                        |                       |          |                        |
| 1       | 0         | 06      |           |                                     |   |   |       |      |                   |       |        |       |   |     | 47         |                        |                                        |                       |          |                        |
| 1       | 0         | 07      |           |                                     |   |   |       |      |                   |       |        |       |   |     | 57         |                        |                                        |                       |          |                        |
| 1       | 0         | 08      |           |                                     |   |   |       |      |                   |       |        |       |   |     | 59         |                        |                                        |                       |          |                        |
| 1       | 0         | 09      |           |                                     |   |   |       |      |                   |       |        |       |   |     | 61         |                        |                                        |                       |          |                        |
| 1       | 0         | 10      |           |                                     |   |   |       |      |                   |       |        |       |   |     | 63         |                        |                                        |                       |          |                        |
| 1       | 0         | 11      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 12      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 13      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 14      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 15      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 16      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 17      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 18      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 19      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |
| 1       | 0         | 20      |           |                                     |   |   |       |      |                   |       |        |       |   |     |            |                        |                                        |                       |          |                        |

d. Output form

Figure 7-1. Screen Formats and Accompanying RPG II Specifications (Part 2 of 2)

Figure 7-1 is analyzed as follows.

■ Display 1

The screen format showing the FROM and TO positions as they relate to the length of every input and bidirectional field.

■ The Input Format Specifications Form

The input specifications showing field locations (lengths) for all input and bidirectional fields. The corresponding field names do not match the SFG-assigned field names.

■ Display 2

The screen format showing the end positions for every output and bidirectional field in the format.

■ The Output Format Specifications Form

The output specifications showing the end position (length) of every output and bidirectional format field and the format name. Field names are not SFG-assigned.

**NOTES:**

1. *The entries that appear on the RPG II forms in Figure 7-1 pertain only to the format data fields. Likewise, the terms input, output, and bidirectional refer only to the direction in which the data will flow between the program and the format. Files for the actual data (a master file, for example, containing names, addresses, etc) must be indicated by entries on the appropriate RPG II forms. Also, I/O directions pertaining to data on these files may deviate from the I/O directions for the format. A new address, for example, may be input to the format, but at the same time it may serve as output to (update) a master.*
2. *If during the format's creation, editing for a numeric field involved use of the decimal point or a sign (+, -, etc), you must indicate the decimal places and signs for that field on the coding form. (See 2.3 for editing rules.)*

Refer to the RPG II user guide for general information on RPG II programs.

### 7.3.2. COBOL

The following subsections explain:

- The major COBOL coding conventions for using screen format services
- How to code for conditional indicators, based on these conventions
- How to code for function keys and multiple workstation users

### 7.3.2.1. Major Coding Considerations

The major considerations for a COBOL program using screen formats relate to the following divisions:

Environment division – SPECIAL NAMES paragraph

Data division – working storage section

Procedure division – ACCEPT and DISPLAY statements

The basic considerations for each of these divisions are:

#### ■ Environment Division

The SPECIAL NAMES paragraph is used to associate screen format services with a workstation. The basic format for the actual statement is:

```
SYSFORMAT IS mnemonic-name ASSIGN TO workstation lfd-name
```

(Additional clauses associated with this format are discussed in 7.3.2.3.)

The mnemonic-name is arbitrary, but the lfd-name must match the name specified for the workstation in your job control DVC-LFD sequence as indicated by the following:

#### DVC-LFD SEQUENCE FOR THE WORKSTATION

```
.
. .
. .
// DVC 200
// USE SFS
// LFD WORKSTN
. .
. .
. .
```

#### THE SPECIAL NAMES STATEMENT

```
SYSFORMAT IS mnemonic-name ASSIGN TO WORKSTN
```

#### ■ Data Division

You may define an area in the working storage section that will be used to hold input data from the format supplied by the workstation operator and output data from the COBOL program to be displayed on the format.

The data structures you define need not have the same names as those that were assigned to the fields during the format's generation. Picture clauses associated with the data structures should, however, correspond to the field lengths and type attributes that were assigned via passes 1 and 2 of the SFG session. Assumed decimal places (v) and signs (s) should be specified wherever applicable. In addition, if the data's internal usage is going to be packed, for example, this should also be indicated via the appropriate USAGE clause (USAGE COMP-3).

#### ■ Procedure Division

Input and output activity between the program and a screen format results from the issuance of DISPLAY or ACCEPT statements in the COBOL program's procedure division.

##### - DISPLAY Statement

Used to display data from the program as output on the screen format

##### - ACCEPT Statement

Used to accept any input data supplied to the format by a workstation operator

The basic formats for both statements are as follows:

```

DISPLAY {identifier-1} [{literal-1} [{identifier-2} ; ...]] UPON mnemonic-name
 [USING {identifier} [{literal}]]

ACCEPT identifier-1 [, identifier-2 , ...] FROM mnemonic-name
 [USING {identifier} [{literal}]]

```

(Additional clauses associated with these statements are discussed in 7.3.2.3.)

The identifiers following the verb DISPLAY or ACCEPT are either group item names or elementary item names for the format's input/output data structure as defined in working storage. As many identifiers as necessary may be listed after the first identifier. Literals, instead of identifiers, may be used with DISPLAY.

The mnemonic name associated with UPON must be the one specified in the SPECIAL NAMES statement.

If specified, the identifier or nonnumeric literal associated with the USING clause is a 1- to 8-character format name (as defined on the home screen). USING is required in only two cases. It is needed to specify the first screen format to be displayed at the workstation if you do not specify the initial-screen parameter in the // USE SFS job control statement. It is also needed to specify a format change in a program using more than one format. In either case, once you specify USING for a particular format, you can, but need not, specify it for subsequent transactions (ACCEPTs or DISPLAYs) with the same format.

The issuance of the first DISPLAY statement for a particular format results in the following:

- The screen format (named on the USE SFS job control statement after the USING clause) is automatically displayed on the workstation screen.
- The data indicated by the DISPLAY verb is displayed in the appropriate (output or bidirectional) fields of the format.

As a general rule, a given screen format must be displayed *upon* before it can be accepted *from*. This is true both for the initial screen format display and every time there is a visible change to that format on the workstation. An exception to this rule occurs when a screen format has no variable output data (if, for example, the format is being used strictly for input). In this case, either an ACCEPT or a DISPLAY statement may be the first transaction with a screen format.

With both ACCEPT and DISPLAY, it is mandatory that the number of characters transmitted *equal* or *exceed* the total number of characters expected by the screen format (the format size). In other words, there is no provision for selectively reading from or writing to only some of the format's fields. You may not, for example, display identifier 1, identifier 2 ... and not identifiers 3, 4, and 5 if these, as elementary (O2) items in working storage, describe three of the format's output fields.

- Displaying

If the total number of characters to be displayed exceeds the format size, excess characters will be truncated. If the number of characters to be displayed is smaller than the format size, a run-time error will result.

- Accepting

If the number of characters to be accepted exceeds the format size, trailing blanks are supplied (to the program) to satisfy the request of the ACCEPT statement.

If the number of characters accepted is less than the format size, a run-time error results.

Consider the screen format in Figure 7-2.

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

```

SALES STATUS
SALESMAN: _____
ITEM: _____
QTY SOLD: _____
COMMISSION:$ _____

```

Figure 7-2. Sample SCREEN01

The field characteristics as defined during creation of the format are:

| Field Name | I/O Usage | Internal Usage | Internal Length | Type       |
|------------|-----------|----------------|-----------------|------------|
| SALESMAN   | Input     | Display        | 20              | Alphabetic |
| ITEM       | Both      | Display        | 7               | Alphabetic |
| QTYSOLD    | Both      | Packed         | 3               | Numeric    |
| FLD00004   | Output    | Display        | 7               | Numeric    |

The format name, as defined on the home screen, is SCREEN01 and it resides on the \$Y\$FMT library. The DVC-LFD sequence for the workstation must be included in the job control stream to execute a program using this format.

```
// JOB SALES
.
.
.
// DVC 200
// USE SFS
// LFD WORKSTN
.
.
.
// EXEC SUMMARY
```

Figure 7-3 shows COBOL program entries associated with SCREEN01 as they relate to the environment, data, and procedure divisions.

```

A B
8 12

ENVIRONMENT DIVISION.
.
.
.
SPECIAL NAMES
 ① SYSFORMAT IS SCREENS ASSIGN TO WORKSTN.
.
.
.
DATA DIVISION.
.
.
.
WORKING-STORAGE SECTION.
 01 OUTAREA
 { 02 ITEM PICTURE 9(7).
 ② { 02 QUANTITY PICTURE 9(5) USAGE COMP-3. ③
 02 COMMISSION PICTURE 9(5)V99. ④
 01 INAREA
 02 SALESMAN PICTURE 9(20).
 ⑤ { 02 ITEM PICTURE A(7).
 02 QUANTITY PICTURE 9(5) USAGE COMP-3.
PROCEDURE DIVISION.
.
.
.
 ⑥ ⑦ ⑧
 DISPLAY OUTAREA UPON SCREENS USING SCREEN01.
.
.
.
 ⑨ DISPLAY ITEM OF OUTAREA, QUANTITY OF OUTAREA COMMISSION
 UPON SCREENS.
.
.
.
 ⑩ { ACCEPT INAREA FROM SCREENS.
 .
 .
 .
 ACCEPT SALESMAN, ITEM OF INAREA, QUANTITY OF INAREA
 FROM SCREENS.

```

Figure 7-3. COBOL Program Entries for SCREEN01 (Part 1 of 2)



## NOTES:

- ① SPECIAL NAMES paragraph specifying SCREENS as the mnemonic name and WORKSTN as the lfd-name assigned via JCL for the workstation.
- ② Data names can, but do not have to, match the names assigned to the format's fields during creation.
- ③ Field was specified as packed during creation (dialog screen I); therefore, USAGE is indicated.
- ④ Editing was specified, via the type attribute screen, for this numeric field during generation of the format. Only the assumed decimal place should be indicated here.
- ⑤ The format fields corresponding to these two items (ITEM and QTYSOLD) were specified with an I/O direction of B (for both input and output, or bidirectional). Both items must be specified in OUTAREA as well as INAREA. The item names may be the same, as shown here, or different.
- ⑥ DISPLAY verb may use the group item name.
- ⑦ SCREENS is the mnemonic name used in the SYSFORMAT IS statement.
- ⑧ The USING clause specifies the first format to be used by the program. (The initial-screen parameter was not specified in the // USE SFS job control statement.) USING is not necessary, but can be specified, in subsequent DISPLAY or ACCEPT statements for the same format.
- ⑨ This DISPLAY statement lists elementary item names as identifiers.
- ⑩ ACCEPT statements showing both the group item and elementary item names as identifiers.

*Figure 7-3. COBOL Program Entries for SCREEN01 (Part 2 of 2)*

In this example, only one format is being used with the program. As many formats as desired may be used, however, as long as the appropriate entries are made in the data division and all DISPLAY and ACCEPT statements reference the proper format name in the USING clause.

You may not, in either the ACCEPT or DISPLAY statement, selectively list identifiers. The following would cause a run-time error since the format is expecting to receive output data for the field COMMISSION as well as ITEM and QUANTITY:

```
DISPLAY ITEM,QUANTITY UPON SCREENS USING SCREEN01
```

Since the 02 data division declarations for bidirectional fields may have the same INAREA as well as OUTAREA names (ITEM and QUANTITY), you are provided with an opportunity to specify the following or similar operations in the procedure division:

| A | B  |
|---|----|
| 8 | 12 |

---

```
DISPLAY OUTAREA UPON SCREENS USING SCREEN01.
ACCEPT INAREA FROM SCREENS USING SCREEN01.
MOVE CORRESPONDING INAREA TO OUTAREA.
DISPLAY OUTAREA UPON SCREENS USING SCREEN01.
```

Consider now the format in Figure 7-4.

PERSONAL CREDIT REPORT

NAME: \_\_\_\_\_

ADDR: \_\_\_\_\_

ACCNT NUMBER: \_\_\_\_\_

PAST DUE AMOUNT:\$\_\_\_\_.\_\_\_\_

PAYMENT DUE:\$\_\_\_\_.\_\_\_\_

NEW BALANCE:\$\_\_\_\_.\_\_\_\_

ACCOUNT STATUS: \_\_\_\_\_

Figure 7-4. Sample SCREEN02

The field characteristics as defined during creation are:

| Field Name | I/O Usage | Internal Usage | Internal Length | Type         |
|------------|-----------|----------------|-----------------|--------------|
| NAME       | Input     | Display        | 25              | Alphabetic   |
| ADDR       | Both      | Display        | 30              | Alphanumeric |
| FLD00003   | Input     | Display        | 9               | Numeric      |
| PAST-AMT   | Output    | Display        | 7               | Numeric      |
| PAYMT-DUE  | Input     | Display        | 7               | Numeric      |
| NEW-BAL    | Output    | Display        | 7               | Numeric      |
| FLD00007   | Output    | Display        | 25              | Alphanumeric |

The format name as defined on the home screen is SCREEN02, and it resides on the \$Y\$FMT library. The DVC-LFD sequence for the workstation is:

```

// JOB CREDIT
.
.
// DVC 200
// USE SFS
// LFD WORKSTN
.
.
// EXEC REPORTS

```

Figure 7-5 shows COBOL program entries that relate to the environment and data divisions.

| A                                       | B                                            |
|-----------------------------------------|----------------------------------------------|
| 8                                       | 12                                           |
| ENVIRONMENT DIVISION.                   |                                              |
| .                                       |                                              |
| .                                       |                                              |
| .                                       |                                              |
| SPECIAL NAMES.                          |                                              |
| SYSFORMAT IS FORMATFL ASSIGN TO WORKSTN |                                              |
| .                                       |                                              |
| .                                       |                                              |
| .                                       |                                              |
| DATA DIVISION.                          |                                              |
| .                                       |                                              |
| .                                       |                                              |
| .                                       |                                              |
| WORKING-STORAGE SECTION.                |                                              |
| 77                                      | INACTIVE PIC X(25) VALUE 'ACCOUNT INACTIVE'. |
| 77                                      | OVERDUE PIC X(25) VALUE 'ACCOUNT OVERDUE'.   |
| 01                                      | OUTAREA.                                     |
| 02                                      | ADDR PIC X(30).                              |
| 02                                      | PAST-AMT PIC 9(5)V99.                        |
| 02                                      | NEW-BAL PIC 9(5)V99.                         |
| 02                                      | ACCNT-STATUS PIC X(25) VALUE SPACES.         |
| 01                                      | INAREA.                                      |
| 02                                      | NAME PIC (25).                               |
| 02                                      | ADDR PIC X(30).                              |
| 02                                      | ACCNT-NUM PIC 9(9).                          |
| 02                                      | PAYMT-DUE PIC 9(5)V99.                       |

Figure 7-5. COBOL Program Entries for SCREEN02

In this example, the utility of being able to list several identifiers in a single ACCEPT or DISPLAY statement can easily be demonstrated. If, for example, you want the literal 'ACCOUNT INACTIVE' to be displayed as output in the ACCOUNT STATUS field (FLD00007 as defined by the SFG and ACCNT-STATUS as described in working-storage) on the format, you simply issue the following statement in the procedure division:

```
DISPLAY ADDR,PAST-AMT,NEW-BAL,INACTIVE UPON FORMATFL USING SCREEN02.
```

If you don't want to display anything for ACCOUNT STATUS on the format, you simply use:

```
DISPLAY OUTAREA UPON FORMATFL USING SCREEN02.
```

You may not, however, use:

```
DISPLAY ADDR,PAST-AMT,NEW-BAL UPON FORMATFL USING SCREEN02.
```

Refer to the 1974 ANSI COBOL programmer reference for more information concerning COBOL programming in general.

### 7.3.2.2. Using Indicators

You can specify two types of indicators when creating a screen format:

1. Option indicators

These affect the physical characteristics of your screen format at run time. They are output from your program to the screen format coordinator.

2. Response indicators

These are the indicators associated with function keys. They are input from the screen format to your program. (Function key indicators require additional coding that option indicators do not require. See 7.3.2.3 for details.)

For indicators to work, you must define them in your program and send their values to the SFC. You do this the same way that you define and send data to output fields or accept data from input fields on a screen format (as discussed in the previous subsection). First you define indicator fields in the WORKING-STORAGE SECTION. For option indicators, you use the DISPLAY statement to pass indicator field data to the SFC. For response indicators, you use the ACCEPT statement to pass a value from the screen format to your program.

When you organize the WORKING-STORAGE SECTION of your program, follow these guidelines:

- The area where you define option indicator fields must precede the area where you define output data fields. The indicator field area can be part of a group that includes the output data fields.
- The area where you define response indicator fields must precede the area where you define input data fields. The response indicator area can also be a part of a group that includes the input data fields.

Option indicator fields must be made a part of the DISPLAY statement along with the output data fields. Similarly, response indicator fields must be made a part of the ACCEPT statement, along with the input data fields. Where indicators are used, the SFC automatically expects the leading bytes from a DISPLAY or an ACCEPT operation to be indicators.

The following example illustrates how to handle indicators in your program.

Once again, consider the format shown in Figure 7-2:

```
 SALES STATUS
SALESMAN: _____
ITEM: _____
QTY SOLD: _____
COMMISSION: $ _ , _ . _ _
```

This time, suppose the format was created with option indicators 10 and 20 and response indicator 30. (To keep this explanation brief and easy to follow, the functions performed by the indicators are not discussed.) One way of defining the indicators, turning them on, and identifying them to the SFC is shown in Figure 7-6.

A B  
8 12

-----  
ENVIRONMENT DIVISION.

.  
.  
.

## SPECIAL NAMES.

SYSFORMAT IS SCREENS ASSIGN TO WORKSTN.

.  
.  
.

## DATA DIVISION.

.  
.  
.

## WORKING STORAGE SECTION.

01 OUTAREA. (1)

02 INDICATOR-REGION

03 IND-10

PICTURE 9.

03 IND-20

PICTURE 9.

02 DATA-REGION

03 ITEM

PICTURE 9(7).

03 QUANTITY

PICTURE 9(5) USAGE IS COMP-3.

03 COMMISSION

PICTURE 9(5)V99.

01 INAREA.

02 INDICATOR-AREA-IN.

03 IND-30

PICTURE 9.

02 DATA-REGION-IN.

02 SALESMAN

PICTURE 9(20).

02 ITEM

PICTURE A(7).

02 QUANTITY

PICTURE 9(5) USAGE IS COMP-3.

## PROCEDURE DIVISION.

.  
.  
.

MOVE 1 TO IND-10. (2)

DISPLAY OUTAREA UPON SCREENS USING SCREEN01. (3)

.  
.  
.

MOVE 1 TO IND-20. (4)

DISPLAY IND-10,IND-20,ITEM,QUANTITY,COMMISSION UPON SCREENS USING SCREEN01.

.  
.  
.

ACCEPT INAREA FROM SCREENS USING SCREEN01. (5)

.  
.  
.

- ① Here, the indicators are grouped with the output data fields. Since the format was created with indicators, the SFC knows that the first several bytes of OUTAREA are going to be indicator fields, not data fields.
- ② A simple MOVE statement is one way of turning an indicator on or off. An indicator is on when it contains the value of 1 and it is off when the value is 0. Although not shown in the example, you must remember to set an indicator value back to 0 to turn an indicator off.
- ③ Here the DISPLAY verb uses the group name OUTAREA, thus passing both the indicator region and the data region to the SFC. Only the data fields appear on the workstation operator's screen.
- ④ Here, instead of using the group name OUTAREA, the elementary items are listed in the DISPLAY statement. The result, however, is the same. The rules discussed in 7.3.2.2 regarding the DISPLAY verb also apply when indicators are used.
- ⑤ The field for the response indicator IND-30 is included in the ACCEPT statement. If an operator presses the user-assigned function key associated with that indicator, a value of 1 is passed back to the indicator field, setting the indicator on.

Figure 7-6. COBOL Program Entries for Indicators (Part 2 of 2)

### 7.3.2.3. Additional Coding Considerations for Function Keys and Multivolume Workstations

As noted previously, the SPECIAL NAMES paragraph and the ACCEPT and DISPLAY verbs have additional clauses not yet discussed. These additional clauses report function key input to a COBOL program and support multivolume workstations (where several terminals operate under the same lfd name).

The complete statement for the SPECIAL NAMES paragraph is:

```
SYSFORMAT IS mnemonic-name ASSIGN TO workstation lfd-name
[CONTROL AREA IS record-name] [WITH FUNCTION-KEYS] [WITH CONNECT-FREE]
```

The CONTROL AREA clause specifies a 40-character area that receives data describing workstation activity. Among the benefits of having a control area are:

- Providing a better means of recovery when an error occurs. This is especially useful in a multiple-workstation environment. If your program encounters a problem with one workstation, it can be directed to handle the problem without affecting the other workstations connected to it.
- Reporting when a workstation is connected to or freed from your program and directing your program to follow a course of action as a result.
- Reporting function/command key input. You can direct your program to perform some operation when a workstation operator presses a user-assigned function/command key.

The control area can be defined in the WORKING-STORAGE or LINKAGE sections of your program. You code the control area as follows:

```

Ø1 WRKSTN-CONTROL.
 Ø5 WS-ID PIC 999.
 Ø5 FILLER PIC X.
 Ø5 WS-STATUS PIC XX.
 Ø5 FUNCTION-KEY PIC 99.
 Ø5 FORMAT-NAME PIC X(8).
 Ø5 NUMBER-CONNECTED PIC 99.
 Ø5 SIZE-OF-DATA TRANSFER PIC 9(5).
 Ø5 FILLER PIC X(17).

```

- WRKSTN-CONTROL is a sample record name for the control area.
- WS-ID identifies the workstation where the most recent ACCEPT or DISPLAY operation was performed.
- WS-STATUS reports a 2-character status code for the workstation that performed the most recent ACCEPT or DISPLAY. Status code details are listed in Table 7-1.

Table 7-1. Status Keys

| Status Key 1            | Status Key 2                                                                    |
|-------------------------|---------------------------------------------------------------------------------|
| 0 Successful completion | 0 No further information                                                        |
| 1 At end                | 0 Function key 15 received                                                      |
|                         | 6 Only active terminal has disconnected<br>(Status key 98 may take precedence.) |
| 2 Invalid format        | 3 Format not found                                                              |
|                         | 4 Format constructed incorrectly                                                |
| 3 Permanent error       | 0 No further information                                                        |
| 9 Workstation exception | 1 Terminal not compatible with format                                           |
|                         | 2 Statement not compatible with format                                          |
|                         | 3 Data not compatible with format                                               |
|                         | 4 Data area not large enough for format                                         |
|                         | 5 Function key, no data                                                         |
|                         | 7 New device connected, no data                                                 |
|                         | 8 Device disconnected (freed), no data                                          |
| 9 Device not connected  |                                                                                 |



- **FORMAT-NAME** is the name of the screen format that is active on the workstation terminal that was the object of the most recent **ACCEPT** or **DISPLAY**.
- **FUNCTION-KEY** holds the number (from 1 to 22) of the function key pressed by an operator prior to the most recent **ACCEPT**. It is set to zero instead of a function key whenever data is accepted. The **FUNCTION-KEY** field is maintained only if the **WITH FUNCTION-KEYS** phrase is specified.
- **SIZE-OF-DATA-TRANSFER** holds the number of characters actually sent to or received from the workstation screen. On a **DISPLAY** or **ACCEPT** operation, this number reflects the number of characters required by the format.

The **WITH FUNCTION-KEYS** clause specifies that whenever function key input is received on an **ACCEPT** verb, that function key value is reported to the control area.

The **WITH CONNECT-FREE** clause specifies that your program is directed to a particular procedure (an **EXCEPTION** path that is described later) when a terminal connects to or disconnects from your program.

The complete format for the **ACCEPT** statement is:

```
ACCEPT identifier-1 [,identifier-2...]
FROM SPECIFIC mnemonic-name
 [USING { identifier-3 }
 { literal }]
[ON EXCEPTION imperative-statement]
```

**SPECIFIC** has meaning only for multivolume workstations. It indicates that input is to be accepted only from the particular workstation whose ID is given in **WS-ID**. If you use **SPECIFIC**, your program waits until the specified workstation supplies data, effectively locking out the other workstations.

You should avoid including a **USING** clause in an **ACCEPT** statement when you're working with multivolume workstation files. Under such conditions, the object of the **USING** clause in an **ACCEPT** statement is always the workstation that performed the most recent **ACCEPT** or **DISPLAY** (the workstation identified by **WS-ID** in the control area). This means if you specify a particular format name in the **USING** clause, that format will appear on only one workstation – the one that performed the last **DISPLAY** or **ACCEPT**. As a result, the workstation (and corresponding format) supplying the data for the **ACCEPT** statement may be different from the format (and corresponding workstation) specified by the **USING** clause.

The **ON EXCEPTION** clause directs your program to follow a different path whenever certain conditions arise, such as:

- When certain errors occur
- If a workstation is connected to or freed from your program
- When a user-assigned function key is pressed (This does not include function keys associated with a response indicator.)

If you specify a control area, you must include the ON EXCEPTION clause in both the ACCEPT and DISPLAY statements. If you do not specify a control area, you cannot use the ON EXCEPTION clause.

The ON EXCEPTION clause lets your COBOL program diagnose and recover from errors that occur as a result of communication between the program and the workstation. When an error occurs, the nature of the problem is reported in the WS-STATUS field of the control area as a 2-digit status code. Your program is directed to follow an exception path that you define. For example, consider the following:

```
ACCEPT INAREA FROM SCREENS USING SCREEN01
ON EXCEPTION PERFORM DIAGNOSIS-ROUTINE.
```

If an operator enters data that is not compatible with the active screen format, the status code 93 (Table 7-1) is reported to the WS-STATUS field of the control area. The ON EXCEPTION path is activated and the DIAGNOSIS-ROUTINE is performed. The DIAGNOSIS-ROUTINE could prompt some corrective action to be taken, as follows:

```
DIAGNOSIS-ROUTINE.
 IF WS-STATUS = 93
 PERFORM BAIL-OUT.
```

You can code procedures for the various status codes listed in Table 7-1, including procedures to follow when a workstation is connected to or freed from your program. As long as you include the CONNECT-FREE clause in the SYSFORMAT statement, each time a workstation connects to or is freed from your program, a status code is reported in the WS-STATUS field and an exception path is activated. Table 7-2 summarizes the effects of connect/free reporting.

Table 7-2. Effects of Connect-Free Reporting

| Workstation Options               | Response to Connect or Free                                                                                                                                                                        |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CONTROL AREA WITH CONNECT/FREE    | Set WS-STATUS to 97 or 98.<br>Set WS-ID to report the device that connected or freed.<br>Update NUMBER-CONNECTED.<br>Execute the EXCEPTION clause.                                                 |
| CONTROL AREA without CONNECT/FREE | Update NUMBER-CONNECTED.<br>If NUMBER-CONNECTED = 0, set status for end-of-file and execute the EXCEPTION clause.<br>Otherwise, do not return control to the COBOL program until data is received. |
| CONTROL AREA not specified        | If no terminal remains connected, terminate the program abnormally.<br>Otherwise, do not return control to the COBOL program until data is received.                                               |

A user-assigned function key can also activate an exception path as long as no response indicator is associated with it. When an operator presses a user-assigned function key, the function key value is reported in the FUNCTION-KEY field, status code 95 is returned to the WS-STATUS field, and the ON EXCEPTION clause is activated.

For example, suppose you want your own help screen displayed whenever function key 5 is pressed. As long as you've specified a CONTROL AREA and the WITH FUNCTION-KEYS clause, you can code your program as follows:

```
ACCEPT INAREA FROM SCREENS USING SCREEN01
ON EXCEPTION PERFORM DIAGNOSIS-ROUTINE
.
.
.
DIAGNOSIS-ROUTINE.
 IF WS-STATUS = 95 PERFORM PROCESS-FUNCTION-KEY.
.
.
.
PROCESS-FUNCTION-KEY.
 IF FUNCTION-KEY = 5
 PERFORM DISPLAY-HELP
```

If a function key is associated with a response indicator, the exception path is not activated. Only the function key number is returned to the FUNCTION-KEYS field of the control area. You must code your program to test for the value returned in the response indicator field and then direct your program to perform a specific function.

Table 7-3 summarizes the effects of function key input.

Table 7-3. Effects of Function Key Reporting

| Workstation Options                     | Response to Function Key Input                                                                              |                                                                                                      |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
|                                         | Response Indicator Set by Function Key                                                                      | Response Indicators Absent or Unaffected                                                             |
| CONTROL AREA with WITH FUNCTION-KEYS    | Return indicators without screen data.<br>Set FUNCTION-KEY.<br>Do not activate the EXCEPTION clause.        | Set FUNCTION-KEY.<br>Activate the EXCEPTION clause.                                                  |
| CONTROL AREA without WITH FUNCTION-KEYS | Return indicators without screen data.<br>Do not set FUNCTION-KEY.<br>Do not activate the EXCEPTION clause. | Ignore the function key input.<br>Do not return control to the COBOL program until data is received. |
| CONTROL AREA not specified              | Return indicators without screen data.                                                                      | Ignore the function key input.<br>Do not return control to the COBOL program until data is received. |

The complete format for the DISPLAY verb is:

```

DISPLAY identifier-1 [identifier-2]...
 UPON mnemonic-name
 [USING { identifier-3 }
 { literal }]
 [ON EXCEPTION imperative-statement]

```

The EXCEPTION clause for the DISPLAY statement performs most of the same functions as it does for the ACCEPT statement. It directs your program to follow an alternate path if certain errors occur when data is sent out to a screen format at a particular workstation.

Refer to the 1974 ANSI COBOL programmer reference for more information concerning COBOL programming in general.

#### 7.3.2.4. Considerations for Workstations with Interprogram Communication

When you combine separately compiled COBOL programs to form a single run unit and to interact with the same workstation, these guidelines apply:

- In all programs, the SYSFORMAT IS statements should be identical except for the mnemonic-name, which is local to each program. While most workstation functions will still behave correctly if the SYSFORMAT statements aren't identical, function key input and connect-free events may occur without being reported to the COBOL program.
- You should establish a CONTROL AREA in the main program, reference it in the LINKAGE SECTION of the subprograms, and include it as an operand in the USING clause in CALL statements.

#### 7.3.3. FORTRAN

Screen format services is activated by the FORTRAN system whenever the SCREEN=name option occurs in the sequential READ or WRITE statement as shown in the following examples:

```

1. READ (20,100,SCREEN='TESTSCR1')I
2. REAL*8 NAME/'TESTSCR2'/
 .
 .
 .
 WRITE (20,200,SCREEN=NAME)

```

The unit argument must be connected to a workstation device, and the screen name specified by the literal or variable after 'SCREEN=' must be the 1- to 8-character name that was assigned to the format (via the home screen) when it was created.

In cases where a format is used for both input and output, the screen name must be specified in the first WRITE statement to create the output fields. The same screen name must then be specified in a READ statement.

Both formatted and unformatted I/O is permitted for a screen format.

■ Formatted I/O

Each screen field must be defined with DISPLAY. The field lengths must exactly match the lengths of the actual screen format fields as specified during the SFG session. The following example requires that 'TESTSCR1' have a 3-character input display field:

```
READ (20,100,SCREEN='TESTSCR1')I
100 FORMAT (I3)
```



## ■ Unformatted I/O

Unformatted I/O is only permitted for characters and integers. Logical and real values are restricted. The internal lengths defined for the screen format must match the FORTRAN data sizes.

| <u>Data</u> | <u>Size</u> |
|-------------|-------------|
| INTEGER     | 4           |
| REAL        | 4           |
| REAL*8      | 8           |
| LOGICAL     | 4           |
| INTEGER*2   | 2           |
| LOGICAL*1   | 1           |
| COMPLEX     | 8           |
| COMPLEX*16  | 16          |

Refer to the FORTRAN IV programmer reference for more information concerning FORTRAN IV programming in general.

### 7.3.4. BAL

The imperative macroinstructions associated with using screen format services are OPEN, CLOSE, DMSSEL (SELECT), DMINP (INPUT), and DMOUT (OUTPUT).

#### 7.3.4.1. Open a File (OPEN)

You use this macroinstruction to initialize a file. This instruction establishes a logical access path (LAP) to the physical file. Only one file can be opened when you issue this instruction. If you want to access more than one file, you must issue an OPEN instruction for each of the files. You must issue this instruction for a file before you attempt any processing.

Format:

| LABEL  | △OPERATION△ | OPERAND                                                                                                                                                                                 |
|--------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [name] | OPEN        | $\left\{ \begin{array}{l} \text{cdibname} \\ (1) \\ 1 \end{array} \right\} \left[ , \left\{ \begin{array}{l} (\text{ribname}) \\ (\emptyset) \\ \emptyset \end{array} \right\} \right]$ |

Positional Parameter 1:

cdibname

Is the symbolic address of the CDIB that contains the screen name that was assigned to the file by the LFD job control statement.

(1) or 1

Indicates that you have preloaded register 1 with the address of the CDIB for the file to be opened.

**Positional Parameter 2:****ribname**

Is the symbolic address of the RIB associated with the file to be opened.

**(0) or 0**

Indicates that you have preloaded register 0 with the address of the RIB associated with the file. If you load register 0 with a value of zero, this indicates that there is no RIB associated with the file. (This also means that a RIB cannot reside at relative address X'000000'.)

If a symbolic address is specified for both positional parameter 1 and positional parameter 2, the comma shown in the instruction format is optional.

**Example:**

```
OPEN FILE1 RIB1
```

**7.3.4.2. Close a File (CLOSE)**

You use this macroinstruction to terminate file processing. Once a file is closed, it cannot be accessed until it is reopened by issuing an OPEN macroinstruction. Only one file can be closed when you issue this instruction. If you have more than one file open in your program, you must issue a CLOSE instruction for each of the files.

**Format:**

| LABEL  | △OPERATION△ | OPERAND                      |
|--------|-------------|------------------------------|
| [name] | CLOSE       | { cdibname }<br>{ (1) }<br>1 |

**Positional Parameter 1:****cdibname**

Is the symbolic address of the CDIB that contains the screen name that was assigned to the file by the LFD job control statement.

**(1) or 1**

Indicates that you have preloaded register 1 with the address of the CDIB associated with the file to be closed.

**Example:**

```
1. CLOSE SCREEN1
2. CLOSE 1
```

1. The file whose CDIB has the symbolic address SCREEN1 is to be closed.
2. The file whose CDIB address has been preloaded into register 1 is to be closed.



### 7.3.4.3. Retrieve a Record (DMINP)

The DMINP imperative makes data from a device available to a user program within a work area. The work area must be large enough to handle all input that has been entered by the operator. If function keys entered set response indicators, the appropriate number of response indicators must precede the data in the work area.

Format:

| LABEL  | △OPERATION△ | OPERAND                                 |
|--------|-------------|-----------------------------------------|
| [name] | DMINP       | {cdibname}, {work area}<br>{(∅)}<br>{∅} |

Positional Parameter 1:

cdibname

Specifies the logical file name that symbolically identifies the CBIB to be affected.

(1) or 1

Indicates that register 1 contains the address of the CDIB.

Positional Parameter 2:

work area

Specifies the symbolic address of a work area that is to contain all response indicators and input data.

(∅) or ∅

Indicates that register 0 is the address of a work area.

NOTES:

1. You need not issue a DMSEL imperative prior to the first imperative if JCL specification was provided with an initial screen selection.
2. If you issue the DMINP imperative without issuing the DMOUT imperative, screen format services issues a DMOUT imperative for the format in effect (whether specific or from JCL). All input fields are displayed with the replenish value. If any output or bidirectional fields exist, however, an error is returned and the DMOUT is not issued.
3. If the input record size specified for variable-length records is not large enough, an error is returned and no read operation is issued.
4. For variable-length records, the size of the variable data plus the number of response indicators plus 4 followed by two blanks is moved to the first two bytes of the work area.

## ■ Error Codes Returned

If the DMINP is unsuccessful, CD\$DDINF+11 contains one of the following error codes:

- X'04' Invalid input record size
- X'05' Output only screen format
- X'06' Output only device
- X'07' Invalid SFS imperative sequence
- X'10' Record level format error
- X'11' Imperative abnormally terminated
- X'12' Field level format error
- X'13' DSD field level format error
- X'14' Undecipherable characters in the input buffer
- X'15' Physical I/O error
- X'16' INT-1 conversion error
- X'17' Maximum retries exceeded
- X'18' DSD syntax error
- X'19' Function key 16 entered

#### 7.3.4.4. Select a Screen Format (DMSEL,SCREEN)

You use the DMSEL,SCREEN imperative macroinstruction to indicate which screen format is to be displayed.

Format:

| LABEL  | △OPERATION△ | OPERAND                                                                             |
|--------|-------------|-------------------------------------------------------------------------------------|
| [name] | DMSEL       | { cdibname }, SCREEN,<br>{ (1) }<br>1<br><br>{ screen format name }<br>{ (0) }<br>0 |

Positional Parameter 1:

cdibname

Is the symbolic address of the CDIB that contains the name that was assigned to the file by the LFD job control statement.

(1) or 1

Indicates that you have preloaded register 1 with the address of the CDIB.

Positional Parameter 2:

SCREEN

Indicates that you are selecting a screen format from a screen format file.

Positional Parameter 3:

screen format name

Is the symbolic address of an 8-byte area that contains the name of the screen format to be selected.

(0) or 0

Indicates that you have preloaded the address of the screen format name in register 0.

**Example:**

```
DMSEL WORKSTN,SCREEN,S1
```

Selects the screen format named in the 8-byte area S1.

If the format name provided contains all X'00', the effect is to terminate screen format services. If you subsequently issue a DMSEL with a format name, screen format services is reactivated.

**NOTE:**

*Resetting screen format services is not permitted for a multiworkstation file. An invalid imperative error code is returned should an attempt be made to do so.*

- Error Conditions Returned (CD\$DDINF+11)
  - X'21' Format name specified could not be found
  - X'22' I/O error while reading the format
  - X'20' Format width greater than screen width
  - X'11' Imperative abnormally terminated

### 7.3.4.5. Output a Record (DMOUT)

The DMOUT imperative permits the user data to be merged from the user's work area with a previously selected format, which is then presented to the device. The keyboard is also unlocked.

Format:

| LABEL  | △OPERATION△ | OPERAND                                                                                                                                                                 |
|--------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [name] | DMOUT       | $\left\{ \begin{array}{l} \text{cdibname} \\ (1) \\ 1 \end{array} \right\}, \left\{ \begin{array}{l} \text{work area} \\ (\emptyset) \\ \emptyset \end{array} \right\}$ |

Positional Parameter 1:

cdibname

Specifies the logical file name that symbolically identifies the CDIB to be affected.

(1) or 1

Indicates that register 1 contains the address of the CDIB.

Positional Parameter 2:

work area

Specifies a positional parameter defining a user work area that contains any option indicators and all fields to be merged with the format.

( $\emptyset$ ) or  $\emptyset$

Indicates register 0 points to the user's work area.

#### ■ Error conditions returned

If the output record size specified for variable-length records is not large enough, an error will be returned and the partial buffer will not be sent.

- Error codes reported in CD\$DDINF+ 11
  - X'03 Invalid output record size
  - X'08' I/O or INT-2 conversion error
  - X'09' Invalid screen format
  - X'11' Imperative abnormally terminated
  - X'26' Output verification error
  - X'29' Device freed

### 7.3.5. ESCORT

When you create a screen format, you give ESCORT a special format in which to output file data. You still designate a structure, but instead of your file being output in the default structure format, it's linked with your screen format. When you designate a screen format, the fields you link to it must be identical to the fields contained in a structure. They must have the same name, and they must be in the same order as the corresponding fields in your structure. Both the format and your structure must also have the same number of fields.

For example, if you designate 10 fields in a format and link it to a structure with only 9 fields, you get an error because ESCORT keeps looking for the 10th field.

It is possible, however, to have a structure with 10 fields and an output format using only 5 fields, but only the first 5 fields of your structure are output. Although you can do it this way, it's always better to have a direct structure/format linkup to eliminate confusion.

Depending on the type of ESCORT program you use with a format (ENTER DATA, CHANGE DATA, SELECT DATA), the following requirements apply:

- ENTER DATA

Specify all fields in the format as input (I).

## ■ CHANGE DATA

Specify all fields in the format as both input and output (B).

Protect the decimal point in decimal numbers with a slash.

You cannot use packed decimal fields in a format. The initial display does not display underlines for the packed field.

When the format is first displayed, only the key field is read. ESCORT ignores all other data until the master record is displayed and changed.

## ■ SELECT (display) DATA

Specify all fields as both input and output (B) or as output only (O). When you display data using only output fields, you must define at least one input field to hold the screen until the XMIT key is pressed. ESCORT ignores any data sent from this input field (B or I).

### 7.3.6. IMS Programming Considerations

#### ■ General Considerations:

- UNISCOPE 100 and 200 terminals must have the PROTECT feature.
- UTS 400 terminals (if operating in native mode) must have the PROTECT feature; FCC switch set to FCC; and control page set to XMIT variable.
- SFS parameter specifying the number of terminals that use screen formats at the same time.
- RESFMT parameter indicating the maximum number of screen formats that remain resident between SFS calls.
- CALL 'BUILD' and CALL/ZG#CALL BUILD commands call screen format services in COBOL and BAL action programs, respectively. A "K" in column 42 of the output specifications forms calls screen format services in RPG II action programs. You must not specify the USE SFS job control statement.

#### ■ RPG II Action Program Considerations:

- Input Format Specifications Form

Every variable data field with an I/O direction of input or both input and output (bidirectional) must be identified on the input format specifications form. The field names need not correspond to those that were assigned to the actual format fields during generation.

The field location entries on the RPG II form refer to the position of these fields with respect to the input data rather than to the position of the input fields on the screen. This is simply because data is either read from the format (from input or bidirectional fields) or written to the format (to output or bidirectional fields) automatically – according to the sequence in which the variable data fields were described when the format was created.

Field lengths are counted sequentially. The FROM entry for the first input or bidirectional field is always 1, and the TO entry is always the number of the last underline in the field as defined in the format. The FROM entry for the next field picks up where the TO entry for the previous field left off. The underlines are counted in relation to this number to get the TO entry.

Let's say that the first input or bidirectional field in the format has a length of 10, and the second has a length of 12. The FROM and TO entries for the first field are 1 and 10. Entries for the second field are therefore 11 and 22. Output fields are not counted at all. If an output field is located between two input or bidirectional fields, simply bypass it.

Control levels, matching fields, and look-ahead fields may not be used with workstation input.

- Output Format Specifications Form

Every variable data field with an I/O direction of output or bidirectional must be identified on the output format specifications form. This includes any bidirectional field that was defined on the input format specifications form.

The END POSITION entries refer to the end positions within the output data and not to their positions on the screen. For example, if there are two output or bidirectional fields with lengths of 10 and 12 respectively, the end position for the first field is 10, and that of the second field is 12.

The name of the format, as it was specified on the home screen, must be entered in the CONSTANT OR EDIT WORD field (columns 45-52) and enclosed in apostrophes. The length of the format name must be entered in column 43 with the letter K preceding it (column 42).

Use of a first-page indicator is not permitted.

■ COBOL Action Programs

The COBOL compiler does not interface with screen format services. All interfaces with screen format services are through IMS (CALL BUILD).

■ BAL Action Programs

Same as COBOL.



## 7.4. CONSIDERATIONS FOR THE WORKSTATION OPERATOR

The following guidelines apply when you enter data on a screen format:

- After you enter data on a screen, make sure the cursor is beyond the last field of the format before transmitting. If you return to a particular field *to correct an error before transmission*, correct the error, position the cursor beyond the last field, and then transmit.
- If you return to a particular field *to correct an error after transmission*, simply correct the error and transmit. You don't have to position the cursor beyond the last field.
- The following are function keys you should not use and operations you should not perform while a screen format is active. Disruption of device control information in the workstation and possibly termination of input transmission may occur. (An SF16 error results.) The restricted keys and operations are:
  - INSERT LINE
  - DELETE LINE
  - LINE DUP
  - DISPLAY ERASE
  - FCC GEN
  - FCC CLEAR
  - FCC ENABLE
  - RESET
  - TAB SET (causes an SF14 error)
  - Changing the control page XMIT option to other than VAR
  - Switching to simulated system mode on a UNISCOPE terminal
  - Powering off the workstation



## Appendix A. Job Control Considerations

### A.1. THE USE SFS STATEMENT

You must indicate within the DVC/LFD sequence for the workstation that screen formats are to be used. This is done via the USE SFS job control statement. The optional positional parameters in this statement may be used as necessary. The format for the statement is:

```
//[symbol] USE SFS [, { [[format-file-lfd-1]/[format-file-lfd-2]] }]
 [{ $YSFMT }]
 [format-file-lfd]
[, initial-screen] [, { nnn }]
 [{ 1 }]
[, screen-format-1=alias-1, ..., screen-format-12=alias-12]
```

where:

SFS

Indicates that the program is to use a screen format.

```
[, { [[format-file-lfd-1]/[format-file-lfd-2]] }]
 [{ $YSFMT }]
 [format-file-lfd]
```

In this parameter, you can provide an lfd name for up to two screen format files. Any name you use must match an lfd name specified in a previously defined device assignment set for a screen format file. (Screen format files are always MIRAM files.) The format-file-lfd is one to eight alphanumeric characters long.

When coding this parameter, remember the following:

- If you omit a format-file-lfd name, it is assumed that all screen formats used reside in the system file \$YSFMT.
- If you code /format-file-lfd-2 alone, \$YSFMT is examined first then the file indicated by format-file-lfd-2.

- If you code `format-file-lfd-1/` alone, the file indicated by `format-file-lfd-1` is examined first then `$$FMT`.
- If you code `format-file-lfd` alone, only the file indicated by `format-file-lfd` is examined.
- If you specify `format-file-lfd-1/format-file-lfd-2`, a search will be made in `format-file-lfd-1` first. If the format isn't found, `format-file-lfd-2` will be searched.

#### `initial-screen`

The initial screen name refers to the 1- to 8-character name (as defined on the home screen) for a format. Use of this parameter depends on the program's language.

#### `BAL`

The initial screen name indicates the first format to be used by the program. A `DMSEL` imperative need not be issued to use this format as the initial format. Refer to 7.3.4 and the consolidated data management macro language user guide/programmer reference for further information on `BAL` macros.

#### `COBOL`

The initial screen name indicates the first format to be used by the program.

Refer to 7.3.2 and the 1974 ANSI COBOL programmer reference for further information on the `SYSFORMAT IS` statement and the use of screen formats with a COBOL program.

#### `RPG II`

This parameter should not be used with an RPG II program. Refer to 7.3.1 and the RPG II user guide for further discussion of RPG II programming and screen formats.

#### `FORTRAN IV`

This parameter may be used with a FORTRAN program. Refer to 7.3.3 and the FORTRAN IV programmer reference for more information on FORTRAN programming.

`{ nnn }`  
`{ 1 }`

This parameter indicates the number of formats to be present in main storage at any one time. It should only be used if the program is going to alternate regularly between two or more formats. The default for this parameter is 1, but you may specify any number from 2 to 255. The number of formats that the program uses is not limited to the number specified here.

If you accept the default, regardless of how many formats a program is going to use, only one format will ever be present in main storage. When the first format is called by the program, it is placed in main storage. When the second is called, it replaces the first, and so on.

The number 3 entered for this parameter indicates that three formats can reside in main storage throughout the program use. You might use this parameter when a program is constantly going to alternate between the same three formats. As a result, the I/O activity that would be necessary to bring the formats back and forth (if the default were accepted) is eliminated. By the same token, however, more main storage is used up to accommodate three formats.

Remember, the number 3 does not restrict the program to the use of only three formats. It means that three formats can reside in main storage at once. If the program uses only three, the same three will always be present. If it uses more, which three formats are present depends on which ones the program calls and when.

`screen-format-1=alias-1,...`

This parameter permits you to equate the name that was given to the format when it was created with a name that the program uses for that format.

Suppose SCREEN01 was the name specified on the home screen during generation. Supplying the alias parameter as SCREEN01=MYSCREEN means that the program is using the name MYSCREEN instead of SCREEN01. No more than 12 alias names may be specified.

The following is an example of the USE statement with some accompanying parameters.

```
// USE SFS,FORMFIL,,2,SCREEN02=MYSCREEN
```

This statement indicates:

- The formats to be used are on a library file other than \$Y\$FMT. The lfd-name for that file is FORMFIL.
- Two formats will reside in main storage.
- The program will use the name MYSCREEN when referencing SCREEN02.

The following are some examples of job control DVC/LFD sequences containing the USE SFS statement.

## Example 1:

```
// JOB PAYROL
.
.
.
// DVC 200 } DVC/LFD sequence
// USE SFS } for the workstation
// LFD WRKSTN } file
.
.
.
// EXEC PRSNL
```

## Example 2:

```
// JOB PAYROL
.
.
.
// DVC 200 } DVC/LFD
// USE SFS,,,,,SCREEN02=MYSCREEN } sequence
// LFD WRKSTN } for the
. } workstation
. } file
.
// EXEC PRSNL
```

Both of these examples contain a DVC/LFD sequence for the workstation where the device code number is 200 and the name that the program will use to reference the workstation is WRKSTN. In both cases, the formats to be used are on \$Y\$FMT; therefore, no LFD/DVC sequence is given for this file and the USE statement does not specify a format's file lfd name. The USE statement in example 2 specifies the alias parameter.

Now consider these examples:

Example 1:

```
// JOB PAYROL
.
.
.
// DVC 50 }
// VOL D00028 } DVC/LFD sequence
// LBL FORMFIL } for the user
// LFD SCRNS } library file
.
.
.
// DVC 200 }
// USE SFS,SCRNS } DVC/LFD sequence
// LFD WRKSTN } for the
 } workstation
.
.
.
// EXEC PRSNL
```

Example 2:

```
// JOB PAYROL
.
.
.
// DVC 50 }
// VOL D00028 } DVC/LFD sequence
// LBL FORMIL } for the user
// LFD SCRNS } library file
.
.
.
// DVC 200 }
// USE SFS,SCRNS,,3,SCRN03=MYSCRN } DVC/LFD
// LFD WRKSTN } sequence
 } for the
 } workstation
.
.
.
// EXEC PRSNL
```

In both examples, the formats are stored on a user-assigned MIRAM library file; consequently, a DVC/LFD sequence for this file must be given. The USE SFS statement within the workstation DVC/LFD sequence must specify the lfd-name for the user library file as shown. The number of formats to be present in main storage at any one time is specified as 3. The alias parameter is also specified. For more information concerning job control in general, refer to the job control user guide.

## A.2. JOB CONTROL FOR MULTIPLE WORKSTATIONS

When building a job control stream for a job that uses screen formats, you may specify the use of more than one workstation. This allows several operators to work with a screen format at once. The following is an example of a job control stream that specifies multiple workstation use.

|                                                                       |                                                                                                 |
|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| Request for EDT<br>(the general editor)                               | LOGON MG2<br>EDT                                                                                |
| Device assignment<br>set (DVC/LFD<br>sequence) for<br>the workstation | { // JOB SCREENIT<br>// DVC 200(3),REQ(3)<br>// USE SFS<br>// UID W01,W02,W03<br>// LFD WORKSTN |
| Device assignment<br>set for the<br>payroll program                   | { // DVC 51<br>// VOL D000028<br>// LBL LODLIBRARY<br>// LFD LODLIB                             |
| Device assignment<br>set for the<br>input data<br>file                | { // DVC 52<br>// VOL D000028<br>// LBL INPAYROLL<br>// LFD INPAY                               |
| Device assignment<br>set for the<br>output data<br>file               | { // DVC 52<br>// VOL D000028<br>// LBL OUTPAYROLL<br>// LFD OUTPAY                             |
|                                                                       | // EXEC PAYROLL,LODLIB<br>/&                                                                    |
| EDT commands                                                          | { @WRITE MODULE=SCREENIT,FILENAME=SAVEJOBS,VSN=D000028<br>@HALT<br>LOGOFF                       |

The DVC for the workstation indicates that three workstations are to be used and that the job will not be scheduled until three workstations are available (as indicated by the REQ parameter). The UID statement provides three workstation user IDs.

Although not directly related to the DVC and UID statements for multiple workstation use, there are some other items related to this job control stream worth mentioning.

- When working with multiple workstations, you may include the FREE statement either within the job control stream or enter it directly at the workstation as a command to free one or more workstations when their use is no longer required.



- As shown in the example, you may use the EDT, @WRITE, and @HALT commands so that the next time you want to run this job all you have to do is enter

```
LOGON MG2
RV SCREENIT,(SAVEJOBS,D000028)
```

at the workstation.

- You must include the USE SFS statement (A.1) in the DVC/LFD sequence for the workstation whenever screen formats are used.
- Since there is no DVC/LFD sequence for a screen format file, it is assumed that all formats are in the library \$Y\$FMT on SYSRES.

For more information about job control, see the job control user guide. For more information about EDT, see the general editor user guide/programmer reference.



## Appendix B. Help Screens

To be supplied



## Appendix C. Function Keys

The following function keys provide general assistance in using the SFG.

- Function key 1

Aborts processing of the current function and returns you to the HOME SCREEN.

- Function key 5

Causes the screen format generator to breakpoint the spool file to a printer. All records currently in the spooled printer output are directed to a printer (pending an available device). Records added to the spool file after the breakpoint are held until a subsequent breakpoint is requested or until the end of the job.

- Function key 13

Requests help for the user. The SFG will display the appropriate procedures and rules that apply to the particular stage of format generation with which you are involved. Press function key 14 to terminate the help screen or press the XMIT key to receive additional HELP screens.

- Function key 14

Returns you to normal processing after HELP is displayed.

- Function key 15

Indicates end of input data.

- Function key 16

Indicates input data cannot be entered properly.

- Function key 20

Restores the screen to its original contents for the current pass if it has inadvertently been destroyed. If pressed during pass 1 or when you are using modify option 7 (CHANGE TEMPLATE ONLY – NO INTERNAL CHANGES), an SFG90 message (FUNCTION NOT SUPPORTED) appears.



## Appendix D. Alphabet Table

The following table indicates characters considered alphabetic in addition to A-Z, a-z, and the space character.

| Keyboard           | Character | ASCII | EBCDIC |
|--------------------|-----------|-------|--------|
| Denmark and Norway | Å         | 5D    | 4F     |
|                    | å         | 7D    | D0     |
|                    | Ø         | 5C    | E0     |
|                    | ø         | 7C    | 6A     |
|                    | Æ         | 5B    | 4A     |
|                    | æ         | 7B    | C0     |
| France             | ç         | 5C    | E0     |
|                    | é         | 7B    | C0     |
|                    | è         | 7D    | D0     |
|                    | ù         | 7C    | 6A     |
|                    | à         | 40    | 7C     |
| Germany            | Ä         | 5B    | 4A     |
|                    | Ö         | 5C    | E0     |
|                    | Ü         | 5D    | 4F     |
|                    | ä         | 7B    | C0     |
|                    | ö         | 7C    | 6A     |
|                    | ü         | 7D    | D0     |
| Italy              | à         | 7B    | C0     |
|                    | è         | 7D    | D0     |
|                    | ì         | 7E    | A1     |
|                    | ò         | 7C    | 6A     |
|                    | ù         | 60    | 79     |
| Sweden and Finland | é         | 5D    | 4F     |
|                    | Ä         | 5B    | 4A     |
|                    | Ö         | 5C    | E0     |
|                    | Å         | 5D    | 4F     |
|                    | Æ         | 40    | 7C     |
|                    | Ü         | 5E    | 5F     |
|                    | ä         | 7B    | C0     |
|                    | ö         | 7C    | 6A     |
|                    | å         | 7D    | D0     |
| Spain              | Ñ         | 5C    | E0     |
|                    | ñ         | 7C    | 6A     |





## Appendix E. SFS Support of Display Attributes

| Display Attribute    | Device           |                  |                                   |                   |                                               |
|----------------------|------------------|------------------|-----------------------------------|-------------------|-----------------------------------------------|
|                      | U 200            | U 400            | UTS/20D<br>UTS/20                 | UTS/40D<br>UTS/40 | Comments                                      |
| Display Intensity:   |                  |                  |                                   |                   |                                               |
| Normal               | Normal intensity | Normal intensity | Normal intensity                  | Normal intensity  |                                               |
| Alternate Brightness | Normal intensity | Low intensity    | Low/reverse/normal (See comment.) | Low intensity     | Dependent upon control page setting           |
| Low                  | Normal intensity | Low intensity    | Low/reverse/normal (See comment.) | Low intensity     | Dependent upon control page setting           |
| Nondisplay           | Normal intensity | Nondisplay       | Normal intensity                  | Nondisplay        |                                               |
| Highlighting:        |                  |                  |                                   |                   |                                               |
| Blink                | Blink character  | Blink field      | Blink field (See comment.)        | Blink field       | Blink character for Katakana UTS/20D & UTS/20 |
| Reverse video        | Blink character  | Blink field      | Blink field                       | Reverse video     | Blink character for Katakana UTS/20D & UTS/20 |
| Emphasis:            |                  |                  |                                   |                   |                                               |
| Underline            | N/A              | N/A              | N/A                               | Underline         |                                               |
| Strike thru          | N/A              | N/A              | N/A                               | Strike thru       |                                               |
| Column separator     | N/A              | N/A              | N/A                               | Column separator  |                                               |



# Index

| Term                              | Reference  | Page |
|-----------------------------------|------------|------|
| <b>A</b>                          |            |      |
| A character                       |            |      |
| indicating alphabetic fields      | 2.2        | 2-1  |
| indicating alphanumeric fields    | 2.2        | 2-2  |
| Action programs                   |            |      |
| BAL                               | 7.3.6      | 7-36 |
| COBOL                             | 7.3.6      | 7-36 |
| RPG II                            | 7.3.6      | 7-35 |
| Alphabet                          |            |      |
| characteristics screen            | 3.3        | 3-5  |
|                                   | 3.3.2      | 3-6  |
| English as default                | 3.3.2      | 3-6  |
| foreign characters                | Appendix D |      |
| Alphabetic characters             |            |      |
| error message field               | 3.4.5      | 3-19 |
| foreign                           | Appendix D |      |
| lowercase                         | 3.3.1      | 3-6  |
| uppercase                         | 3.3.1      | 3-6  |
| Alphabetic fields                 |            |      |
| default values for dialog screens | 3.5.8      | 3-40 |
| editing                           | 2.3.1      | 2-2  |
|                                   | 4.1.2      | 4-5  |
| range checking                    | 3.5.7      | 3-38 |
| rules                             | 2.2        | 2-1  |
| type attributes                   | 4.1.2      | 4-5  |
| variable data fields              | 1.3.2.2    | 1-9  |
| Alphanumeric fields               |            |      |
| default values for dialog screens | 3.5.8      | 3-40 |
| editing                           | 2.3.1      | 2-2  |
| range checking                    | 3.5.7      | 3-38 |
| rules                             | 2.2        | 2-2  |
| type attributes                   | 4.1.2      | 4-5  |
| variable data fields              | 1.3.2.2    | 1-9  |

| Term                                                 | Reference               | Page |
|------------------------------------------------------|-------------------------|------|
| Alternate display properties                         | Appendix E<br>3.4.4     | 3-18 |
| Asterisk (*) character                               | 2.3.2                   | 2-3  |
| Attributes                                           | See type<br>attributes. |      |
| <b>B</b>                                             |                         |      |
| B (both), input and output<br>(bidirectional) fields | 4.1.3                   | 4-8  |
| BAL                                                  |                         |      |
| close a file (CLOSE)                                 | 7.3.4.2                 | 7-28 |
| imperative macroinstructions                         | 7.3.4                   | 7-27 |
| open a file (OPEN)                                   | 7.3.4.1                 | 7-27 |
| output a record (DMOUT)                              | 7.3.4.5                 | 7-33 |
| retrieve a record (DMINP)                            | 7.3.4.3                 | 7-29 |
| select a screen format<br>(DMSSEL,SCREEN)            | 7.3.4.4                 | 7-31 |

| Term                                          | Reference | Page | Term                                                  | Reference | Page |
|-----------------------------------------------|-----------|------|-------------------------------------------------------|-----------|------|
| Bidirectional fields                          |           |      | <b>C</b>                                              |           |      |
| characteristics screen                        | 3.3.9     | 3-10 | CD\$DDINF +11 error codes                             |           |      |
| conditional display                           | 3.5.3     | 3-33 | DMINP imperative macroinstruction                     | 7.3.4.3   | 7-30 |
| conditional protection                        | 3.5.6     | 3-37 | DMOUT imperative macroinstruction                     | 7.3.4.5   | 7-34 |
| conditional retention                         | 3.5.5     | 3-37 | DMSEL,SCREEN imperative macroinstruction              | 7.3.4.4   | 7-32 |
| dialog screen 3                               | 3.5.3     | 3-33 |                                                       |           |      |
| dialog screen 4                               | 3.5.4     | 3-35 |                                                       |           |      |
| dialog screen 5                               | 3.5.5     | 3-36 |                                                       |           |      |
| dialog screen 6                               | 3.5.6     | 3-37 | CHANGE I/O option                                     |           |      |
| display retention                             | 3.3.9     | 3-10 | modification summary                                  | 5.2       | 5-6  |
| display retention screen                      | 3.4.6     | 3-21 | modify screen                                         | 5.1.1     | 5-2  |
| ESCORT                                        | 7.3.5     | 7-35 |                                                       | 5.1.1.3   | 5-4  |
| IMS                                           | 7.3.6     | 7-35 |                                                       |           |      |
| list screen                                   | 6.1.2.6   | 6-7  | CHANGE TEMPLATE ONLY - NO INTERNAL CHANGES option     |           |      |
| nondisplay screen                             | 3.4.8     | 3-23 | modification summary                                  | 5.2       | 5-6  |
| RPG II                                        | 7.3.1     | 7-5  | modify screen                                         | 5.1.1     | 5-2  |
|                                               | 7.3.6     | 7-35 |                                                       | 5.1.1.4   | 5-4  |
| SHOW function                                 | 6.1.1     | 6-1  | CHANGE TEMPLATE option                                |           |      |
| specifying                                    | 1.3.2.2   | 1-10 | modification summary                                  | 5.2       | 5-6  |
|                                               | 4.1.3     | 4-8  | modify screen                                         | 5.1.1     | 5-2  |
| See also I/O fields and variable data fields. |           |      |                                                       | 5.1.1.1   | 5-3  |
| Blank character                               |           |      | CHANGE TYPE option                                    |           |      |
| edit screen                                   | 3.4.3     | 3-17 | modification summary                                  | 5.2       | 5-6  |
| editing alphabetic and alphanumeric fields    | 2.3.1     | 2-2  | modify screen                                         | 5.1.1     | 5-2  |
| simple insertion in numeric fields            | 2.3.2     | 2-3  |                                                       | 5.1.1.2   | 5-3  |
|                                               |           |      | Characteristics screen                                |           |      |
|                                               |           |      | alphabet (line 3)                                     | 3.3.2     | 3-6  |
|                                               |           |      | creating new format                                   | 3.3       | 3-5  |
|                                               |           |      |                                                       | 4.1       | 4.2  |
|                                               |           |      | cursor positioning and overwriting                    | 3.6       | 3-42 |
|                                               |           |      | display retention (line 9)                            | 3.3.9     | 3-10 |
|                                               |           |      | erasing and unlocking options (lines 4 and 5)         | 3.3.4     | 3-7  |
|                                               |           |      | error message field (line 8)                          | 3.3.8     | 3-10 |
|                                               |           |      | error retry counts (line 6)                           | 3.3.5     | 3-8  |
|                                               |           |      | function keys (line 10)                               | 3.3.10    | 3-10 |
|                                               |           |      | lowercase translation (line 2)                        | 3.3.1     | 3-6  |
|                                               |           |      | nondisplayed field (line 11)                          | 3.3.11    | 3-11 |
|                                               |           |      | original and overlay formats (line 3)                 | 3.3.3     | 3-6  |
|                                               |           |      | prompting additional screens                          | 3.3       | 3-5  |
|                                               |           |      |                                                       | 3.4       | 3-11 |
|                                               |           |      | related optional and dialog screens                   | Table 3-1 | 3-11 |
|                                               |           |      |                                                       | 3.4       | 3-11 |
|                                               |           |      |                                                       | 3.5       | 3-24 |
|                                               |           |      | sample format                                         | 4.2.1.1   | 4-13 |
|                                               |           |      | special display control (line 7)                      | 3.3.7     | 3-9  |
|                                               |           |      | special editing characters (line 6)                   | 3.3.6     | 3-9  |
|                                               |           |      | CLOSE (close a file), BAL imperative macroinstruction | 7.3.4.2   | 7-28 |

| Term                                                       | Reference                               | Page | Term                                              | Reference | Page  |
|------------------------------------------------------------|-----------------------------------------|------|---------------------------------------------------|-----------|-------|
| COBOL                                                      |                                         |      | Conditional retention dialog<br>(dialog screen 5) | 3.5.5     | 3-36  |
| connect-free reporting                                     | Table 7-2                               | 7-24 | Conditional values screen                         | 6.1.2     | 6-2   |
| data division                                              | 7.3.2.1                                 | 7-10 |                                                   | 6.1.2.8   | 6-11  |
| environment division                                       | 7.3.2.1                                 | 7-10 | CR character                                      |           |       |
| function key reporting                                     | Table 7-3                               | 7-25 | dialog screen 1                                   | 3.5.1     | 3-27  |
| function keys and multivolume<br>workstations              | 7.3.2.3                                 | 7-21 | editing rules for signed output<br>fields         | Table 2-1 | 2-6   |
| indicators                                                 | 7.3.2.2                                 | 7-18 | fixed insertion in numeric fields                 | 2.3.2     | 2-3   |
| interprogram communication<br>considerations               | 7.3.2.4                                 | 7-26 | CREATE-FROM function                              |           |       |
| major coding considerations                                | 7.3.2.1                                 | 7-10 | characteristics screen                            | 3.3.2     | 3-6   |
| procedure division                                         | 7.3.2.1                                 | 7-11 | dialog screen 1                                   | 3.5.1     | 3-27  |
| status keys                                                | Table 7-1                               | 7-22 | home screen                                       | 3.2       | 3-1   |
| Comma                                                      |                                         |      | modifying formats                                 | 3.2.1     | 3-3   |
| dialog screen 1                                            | 3.5.1                                   | 3-29 | new format name                                   | 5.1       | 5-1   |
| editing input or bidirectional<br>fields                   | 1.3.2.3                                 | 1-11 | old format name and library                       | 5.2       | 5-6   |
| protecting                                                 | 3.5.1                                   | 3-30 | CREATE function                                   |           |       |
| simple and fixed insertion in<br>numeric fields            | 2.3.2                                   | 2-3  | basic screen format                               | 4.1       | 4-1   |
| Command keys                                               | See function/<br>command key<br>screen. |      | characteristics screen                            | 3.3       | 3-5   |
| Conditional display dialog<br>(dialog screen 3)            | 3.5.3                                   | 3-33 | complex screen format                             | 4.1       | 4-2   |
| Conditional display properties dialog<br>(dialog screen 4) | 3.5.4                                   | 3-34 | dialog screens                                    | 4.2       | 4-10  |
| Conditional erase/replenish screen                         |                                         |      | duplicate lines                                   | 4.2.1     | 4-10  |
| characteristics screen                                     | Table 3-1                               | 3-11 | home screen                                       | 3.5       | 3-24  |
| sample format                                              | 3.4.2                                   | 3-13 | optional screens                                  | 4.2       | 4-10  |
| use                                                        | 4.2.1                                   | 4-10 | original formats                                  | 4.2.1.7   | 4-17  |
| Conditional indicators                                     |                                         |      | overlay formats                                   | 4.4       | 4-24  |
| conditional values screen                                  | 6.1.2.8                                 | 6-11 | pass 1, basic format                              | 3.2       | 3-1   |
| dialog screens                                             | 3.4.1                                   | 3-12 | pass 1, complex format                            | 4.1       | 4-2   |
| display properties                                         | 3.5                                     | 3-24 | pass 2, basic format                              | 3.2.2     | 3-4   |
| erase                                                      | 3.5.4                                   | 3-34 | pass 2, complex format                            | 3.4       | 3-11  |
| error message screen                                       | 3.4.2                                   | 3-13 | pass 3, basic format                              | 4.2       | 4-10  |
| list screen                                                | 3.4.5                                   | 3-20 | pass 3, complex format                            | 3.3.3     | 3-6   |
| optional screens                                           | 6.1.2.6                                 | 6-8  | screen format                                     | 3.3.3     | 3-6   |
| replenish                                                  | 3.4.1                                   | 3-12 | sequence of steps                                 | 4.3       | 4-20a |
| retain                                                     | 3.4.2                                   | 3-13 | Currency sign                                     |           |       |
| See also indicators.                                       | 3.5.5                                   | 3-36 | automatically protected                           | 2.4       | 2-5   |
| Conditional protection dialog<br>(dialog screen 6)         | 3.5.6                                   | 3-37 | edit screen                                       | 3.5.1     | 3-30  |
|                                                            |                                         |      | editing input or<br>bidirectional fields          | 3.4.3     | 3-16  |
|                                                            |                                         |      | fixed and floating insertion<br>in numeric fields | 1.3.2.3   | 1-11  |
|                                                            |                                         |      |                                                   | 2.3.2     | 2-3   |

| Term                             | Reference | Page | Term                                       | Reference             | Page |
|----------------------------------|-----------|------|--------------------------------------------|-----------------------|------|
| Cursor                           |           |      | <b>D</b>                                   |                       |      |
| characteristics screen           | 3.6       | 3-41 | DB character                               |                       |      |
| dialog screens                   | 3.5       | 3-25 | edit screen                                | 3.4.3                 | 3-16 |
| home screen                      | 3.6       | 3-42 | editing rules for signed output fields     | Table 2-1             | 2-6  |
| initial I/O screen               | 4.1.3     | 4-9  | fixed insertion in numeric fields          | 2.3.2                 | 2-3  |
| optional screens                 | 3.6       | 3-42 | Decimal point                              |                       |      |
| positioning, creating a format   | 3.6       | 3-41 | automatic alignment                        | 2.4                   | 2-5  |
| positioning, filling in a format | 7.4       | 7-37 | dialog screen 1                            | 3.5.1                 | 3-29 |
| template screen                  | 4.1.1     | 4-3  | editing input or bidirectional fields      | 1.3.2.3               | 1-11 |
| type attribute screen            | 4.1.2     | 4-7  | fixed insertion in numeric fields          | 2.3.2                 | 2-3  |
|                                  |           |      | protection                                 | 2.4                   | 2-5  |
|                                  |           |      | Default values                             |                       |      |
|                                  |           |      | characteristics screen                     | 4.1                   | 4-3  |
|                                  |           |      |                                            | 4.2.1.1               | 4-13 |
|                                  |           |      | dialog screen 2                            | 3.5.2                 | 3-31 |
|                                  |           |      | dialog screen 4                            | 3.5.4                 | 3-34 |
|                                  |           |      | dialog screens                             | 3.5.8                 | 3-40 |
|                                  |           |      | home screen                                | 3.2.2                 | 3-4  |
|                                  |           |      |                                            | 4.1                   | 4-2  |
|                                  |           |      |                                            | 4.2.1.1               | 4-13 |
|                                  |           |      |                                            | 5.1                   | 5-2  |
|                                  |           |      | I/O screen                                 | 4.1.3                 | 4-8  |
|                                  |           |      | overwriting                                | 3.6                   | 3-41 |
|                                  |           |      | special display                            | 3.5.4                 | 3-34 |
|                                  |           |      | DELETE function                            |                       |      |
|                                  |           |      | home screen                                | 3.2                   | 3-1  |
|                                  |           |      |                                            | 3.2.1                 | 3-3  |
|                                  |           |      |                                            | 6.2                   | 6-13 |
|                                  |           |      | old format name and library use            | 3.2.3                 | 3-4  |
|                                  |           |      |                                            | 6.1                   | 6-1  |
|                                  |           |      | Device assignment set                      | See DVC-LFD sequence. |      |
|                                  |           |      | Dialog screen 1                            |                       |      |
|                                  |           |      | changing I/O direction of field            | 3.5.1                 | 3-27 |
|                                  |           |      | default values                             | 3.5.8                 | 3-40 |
|                                  |           |      |                                            | 3.5.1                 | 3-30 |
|                                  |           |      | field requesting dialogs                   | 3.5.1                 | 3-26 |
|                                  |           |      | function                                   | 3.5.1                 | 3-26 |
|                                  |           |      |                                            | Table 3-2             | 3-25 |
|                                  |           |      | internal length calculated by SFG          | 3.5.1                 | 3-27 |
|                                  |           |      | rules                                      | 3.5.1                 | 3-30 |
|                                  |           |      | sample format                              | 4.2.1                 | 4-11 |
|                                  |           |      |                                            | 4.2.1.7               | 4-17 |
|                                  |           |      | selecting other dialog screens             | 3.5.1                 | 3-28 |
|                                  |           |      | type attributes and I/O direction of field | 3.5.1                 | 3-28 |
|                                  |           |      |                                            | Table 3-3             | 3-30 |
|                                  |           |      | type of internal usage                     | 3.5.1                 | 3-27 |
|                                  |           |      | warning messages on internal length        | 3.5.1                 | 3-28 |

| Term                              | Reference | Page | Term                                                              | Reference | Page                       |
|-----------------------------------|-----------|------|-------------------------------------------------------------------|-----------|----------------------------|
| Dialog screen 2                   |           |      | default values                                                    | 3.5.8     | 3-40                       |
| alphabetic characters             | 3.5.2     | 3-32 | dialog screen 1                                                   | 3.5.1     | 3-26                       |
| default values                    | 3.5.8     | 3-40 | dialog screen 2                                                   | 3.5.2     | 3-31                       |
|                                   | 3.5.2     | 3-31 | dialog screen 3                                                   | 3.5.3     | 3-33                       |
| display of field                  | 3.5.2     | 3-31 | dialog screen 4                                                   | 3.5.4     | 3-34                       |
| field response                    | 3.5.2     | 3-31 | dialog screen 5                                                   | 3.5.5     | 3-36                       |
| function                          | 3.5.2     | 3-31 | dialog screen 6                                                   | 3.5.6     | 3-37                       |
|                                   | Table 3-2 | 3-25 | dialog screen 7                                                   | 3.5.7     | 3-38                       |
| numeric values                    | 3.5.2     | 3-32 | duplicate lines                                                   | 4.4       | 4-25                       |
| replenished input field           | 3.5.2     | 3-33 | initiating                                                        | 4.1.3     | 4-7                        |
| sample format                     | 4.2.1     | 4-11 | LIST function                                                     | 6.1.2     | 6-2                        |
|                                   | 4.2.1.7   | 4-17 | range checking                                                    | 3.5.7     | 3-38                       |
| warning messages for line 3       | 3.5.2     | 3-32 | related dialog screens                                            | Table 3-1 | 3-11                       |
|                                   |           |      |                                                                   | 4.2       | 4-10                       |
| Dialog screen 3                   |           |      | related optional screens                                          | Table 3-1 | 3-11                       |
| default values                    | 3.5.8     | 3-40 |                                                                   | 4.2       | 4-10                       |
| function                          | 3.5.3     | 3-33 | special display properties                                        | 3.5.4     | 3-34                       |
|                                   | Table 3-2 | 3-25 | use                                                               | 3.5       | 3-24                       |
|                                   |           |      |                                                                   | 4.2       | 4-10                       |
| Dialog screen 4                   |           |      |                                                                   | 4.2.1.6   | 4-16                       |
| conditional display properties    | 3.5.4     | 3-34 |                                                                   | 4.2.1.7   | 4-17                       |
| default display properties        | 3.5.4     | 3-34 | Direction of fields                                               |           |                            |
| default values                    | 3.5.8     | 3-40 | dialog screen 1                                                   | 3.5.1     | 3-26                       |
| function                          | 3.5.4     | 3-34 | list screen                                                       | 6.1.2.6   | 6-8                        |
|                                   | Table 3-2 | 3-25 | RPG II                                                            | 7.3.1     | 7-5                        |
| intensity/emphasis                | 3.5.4     | 3-34 | variable data fields                                              | 1.3.2.2   | 1-9                        |
|                                   | Table 3-4 | 3-36 | See also I/O directions, I/O fields,<br>and bidirectional fields. |           |                            |
| sample format                     | 4.2.1     | 4-11 |                                                                   |           |                            |
|                                   | 4.2.1.7   | 4-18 | Display attributes                                                |           | See display<br>properties. |
| Dialog screen 5                   |           |      |                                                                   |           |                            |
| conditional retention             | 3.5.8     | 3-36 | Display constants                                                 |           |                            |
|                                   | 3.4.6     | 3-21 | display properties                                                | 3.4.4     | 3-17                       |
| default values                    | 3.5.8     | 3-40 | emphasis                                                          | 3.4.4     | 3-17                       |
| function                          | 3.5.5     | 3-36 | format layout                                                     | 4.1.1     | 4-3                        |
|                                   | Table 3-2 | 3-25 | intensity                                                         | 3.4.4     | 3-17                       |
| Dialog screen 6                   |           |      | part of screen format                                             | 1.3.2.1   | 1-9                        |
| conditional protection            | 3.5.6     | 3-37 | protected and unprotected parts                                   | 1.3.2.3   | 1-11                       |
| default values                    | 3.5.8     | 3-40 |                                                                   |           |                            |
| function                          | 3.5.6     | 3-37 | Display control and error message                                 |           |                            |
|                                   | Table 3-2 | 3-25 | display                                                           | 6.1.2.3   | 6-5                        |
| Dialog screen 7                   |           |      |                                                                   | 6.1.2     | 6-2                        |
| default values                    | 3.5.8     | 3-40 | Display functions                                                 |           |                            |
| function                          | 3.5.7     | 3-38 | home screen                                                       | 3.2.1     | 3-2                        |
|                                   | Table 3-2 | 3-25 | LIST                                                              | 3.2.1     | 3-3                        |
| range checking                    | 3.5.7     | 3-38 |                                                                   | 3.2.3     | 3-4                        |
|                                   |           |      |                                                                   | 6.1.2     | 6-1                        |
| Dialogs                           |           |      | SHOW                                                              | 3.2.1     | 3-3                        |
| changing field characteristics    | 5.1       | 5-1  |                                                                   | 3.2.3     | 3-4                        |
| characteristics screen selections | Table 3-1 | 3-11 | SPOOL                                                             | 6.1.1     | 6-1                        |
| conditional display               | 3.5.3     | 3-33 |                                                                   | 3.2.1     | 3-3                        |
| conditional indicators            | 3.4.1     | 3-12 |                                                                   | 3.2.3     | 3-4                        |
|                                   | 3.5       | 3-24 |                                                                   | 6.1.3     | 6-13                       |
| conditional protection            | 3.5.6     | 3-37 | use                                                               | 3.2.1     | 3-2                        |
| conditional retention             | 3.5.5     | 3-36 |                                                                   | 6.1       | 6-1                        |

| Term                              | Reference  | Page | Term                                  | Reference | Page |
|-----------------------------------|------------|------|---------------------------------------|-----------|------|
| Display properties                |            |      | DMSEL,SCREEN (select a screen format) |           |      |
| alternate                         | Appendix E |      | BAL imperative macroinstruction       | 7.3.4     | 7-27 |
| conditional                       | 3.5.4      | 3-34 | CD\$DDINF+11 error conditions         | 7.3.4.4   | 7-31 |
| default                           | 3.5.4      | 3-34 | returned                              | 7.3.4.4   | 7-32 |
| dialog screen 4                   | 3.5.4      | 3-34 | Duplicate lines                       | 4.4       | 4-24 |
| display constants                 | 3.4.4      | 3-18 | DVC-LFD sequence                      |           |      |
| display control and error message |            |      | COBOL                                 | 7.3.2     | 7-10 |
| display                           | 6.1.2.3    | 6-5  | examples                              | A.1       | A-3  |
| error message screen              | 3.4.5      | 3-20 | job control                           | 7.2       | 7-1  |
| fields                            | 3.4.4      | 3-17 |                                       |           |      |
| fields in error                   | 3.4.4      | 3-18 |                                       |           |      |
| input fields                      | 3.4.4      | 3-17 |                                       |           |      |
| output fields                     | 3.4.4      | 3-17 |                                       |           |      |
| selections                        | Table 3-4  | 3-36 |                                       |           |      |
| special display properties dialog | 3.5.4      | 3-34 |                                       |           |      |
| special display screen            | 3.4.4      | 3-17 |                                       |           |      |
| Display retention                 |            |      |                                       |           |      |
| characteristics screen            | 3.3        | 3-5  |                                       |           |      |
|                                   | 3.3.9      | 3-10 |                                       |           |      |
| display retention screen          | 3.4.6      | 3-21 |                                       |           |      |
| Display retention screen          |            |      |                                       |           |      |
| characteristics screen            | 3.3        | 3-5  |                                       |           |      |
|                                   | 3.3.9      | 3-10 |                                       |           |      |
| field display attributes          | 3.4.6      | 3-21 |                                       |           |      |
|                                   | 3.5.4      | 3-34 |                                       |           |      |
| field level display retention     | 3.4.6      | 3-21 |                                       |           |      |
|                                   | 3.5.5      | 3-36 |                                       |           |      |
| use                               | 3.4        | 3-11 |                                       |           |      |
|                                   | 3.4.6      | 3-21 |                                       |           |      |
| DMINP (retrieve a record)         |            |      |                                       |           |      |
| BAL imperative macroinstruction   | 7.3.4      | 7-27 |                                       |           |      |
|                                   | 7.3.4.3    | 7-29 |                                       |           |      |
| CD\$DDINF+11 error codes returned | 7.3.4.3    | 7-30 |                                       |           |      |
| DMOUT                             | 7.3.4.3    | 7-29 |                                       |           |      |
| DMSEL                             | 7.3.4.3    | 7-29 |                                       |           |      |
| function keys                     | 7.3.4.3    | 7-29 |                                       |           |      |
| input record size                 | 7.3.4.3    | 7-29 |                                       |           |      |
| response indicators               | 7.3.4.3    | 7-29 |                                       |           |      |
| variable-length records           | 7.3.4.3    | 7-30 |                                       |           |      |
| work area                         | 7.3.4.3    | 7-29 |                                       |           |      |
| DMOUT (output a record)           |            |      |                                       |           |      |
| BAL imperative macroinstruction   | 7.3.4      | 7-27 |                                       |           |      |
|                                   | 7.3.4.5    | 7-33 |                                       |           |      |
| CD\$DDINF+11 error codes          | 7.3.4.5    | 7-34 |                                       |           |      |
| error conditions returned         | 7.3.4.5    | 7-33 |                                       |           |      |



| Term                                                | Reference  | Page | Term                                        | Reference | Page |
|-----------------------------------------------------|------------|------|---------------------------------------------|-----------|------|
| <b>E</b>                                            |            |      |                                             |           |      |
| Edit characters                                     |            |      | Erase indicator                             | 3.4.2     | 3-13 |
| edit screen                                         | 3.4.3      | 3-16 | Erasing option                              |           |      |
| variable data fields                                | 4.1.2      | 4-5  | characteristics screen                      | 3.3       | 3-5  |
|                                                     |            |      |                                             | 3.3.4     | 3-7  |
|                                                     |            |      |                                             | A.2       | A-7  |
| Edit mask                                           |            |      | conditional erase/replenish screen          | 3.4.2     | 3-15 |
| dialog screen 1                                     | 3.5.1      | 3-26 |                                             | 4.2.1     | 4-10 |
| editing input or bidirectional fields               | 1.3.2.3    | 1-11 | Error message display                       | 6.1.2.3   | 6-5  |
| editing numeric fields                              | 2.4        | 2-4  | Error message field, characteristics screen | 3.3       | 3-5  |
| Edit screen                                         |            |      |                                             | 3.3.8     | 3-10 |
| rules for completion                                | 3.4.3      | 3-16 | Error message screen                        |           |      |
| special editing characters (characteristics screen) | 3.3.6      | 3-9  | characteristics screen                      | 3.4.5     | 3-18 |
|                                                     | 3.4.3      | 3-16 |                                             | 3.3.8     | 3-10 |
| special editing display use                         | 6.1.2.2    | 6-4  | display control and error message display   | 6.1.2.3   | 6-5  |
|                                                     | 3.4        | 3-11 | emphasis                                    | 3.4.5     | 3-20 |
| yen sign for Katakana                               | 3.4.3      | 3-16 | intensity                                   | 3.4.5     | 3-20 |
|                                                     |            |      | sample format                               | 4.2.1     | 4-10 |
|                                                     |            |      |                                             | 4.2.1.3   | 4-14 |
| Editing                                             |            |      | use                                         | 3.4       | 3-11 |
| alphabetic and alphanumeric fields                  | 2.3.1      | 2-2  |                                             | 3.4.5     | 3-18 |
| edit screen                                         | 3.4.3      | 3-16 | Error retry counts, characteristics screen  | 3.3       | 3-5  |
| examples                                            | Table 2-2  | 2-7  |                                             | 3.3.5     | 3-8  |
| insertion                                           | 2.3.1      | 2-2  | Errors                                      |           |      |
| numeric fields                                      | 2.3.2      | 2-3  | dialog screen 1                             | 3.5.1     | 3-28 |
| results                                             | 2.4        | 2-4  | dialog screen 2                             | 3.5.2     | 3-33 |
| rules for signed output fields                      | Table 2-1  | 2-6  | dialog screen 7                             | 3.5.7     | 3-39 |
| special characters                                  | 3.3.6      | 3-9  | duplicate lines                             | 4.4       | 4-25 |
| suppression/replacement                             | 2.3.2      | 2-4  | error message display                       | 6.1.2.3   | 6-5  |
| variable data fields                                | 4.1.2      | 4-5  | fields in                                   | 3.4.4     | 3-7  |
|                                                     |            |      | function key                                | 3.4.7     | 3-23 |
| Editing characters                                  |            |      | indicated on workstation                    | 1.3.1     | 1-7  |
| initiating dialogs                                  | 4.1.3      | 4-8  | initial I/O screen                          | 4.1.3     | 4-8  |
| input or bidirectional fields                       | 1.3.2.3    | 1-11 | messages (home screen)                      | 3.7       | 3-43 |
| special                                             | 3.3.6      | 3-9  | rules for dialog screens                    | 3.5       | 3-25 |
|                                                     | 3.4.3      | 3-16 | type attribute screen                       | 4.1.2     | 4-7  |
|                                                     |            |      | type of data                                | 1.4       | 1-13 |
| Emphasis                                            |            |      | ESCORT                                      | 7.3.5     | 7-34 |
| alternate display properties                        | Appendix E |      | Exclamation point                           |           |      |
| device support                                      | Appendix E |      | dialog screen 1                             | 3.5.1     | 3-28 |
| dialog screen 4                                     | 3.5.4      | 3-34 | editing alphabetic and alphanumeric fields  | 2.3.1     | 2-2  |
| display constants                                   | 3.4.4      | 3-18 | editing numeric fields                      | 2.3.2     | 2-3  |
| display property                                    | 3.4.4      | 3-17 | initiating dialogs                          | 4.1.3     | 4-8  |
|                                                     | Table 3-4  | 3-36 | results of editing                          | 2.4       | 2-4  |
| error message screen                                | 3.4.5      | 3-20 | type attribute screen                       | 4.1.2     | 4-6  |
| fields in error                                     | 3.4.4      | 3-18 |                                             |           |      |
| input fields                                        | 3.4.4      | 3-17 |                                             |           |      |
| output fields                                       | 3.4.4      | 3-17 |                                             |           |      |
| selections                                          | Table 3-4  | 3-36 |                                             |           |      |
| special display properties                          | 3.5.4      | 3-34 |                                             |           |      |
| special display screen                              | 3.4.4      | 3-17 |                                             |           |      |
|                                                     | 3.5.4      | 3-55 |                                             |           |      |
| English, default alphabet                           | 3.3.2      | 3-6  |                                             |           |      |

| Term                                    | Reference                                   | Page |
|-----------------------------------------|---------------------------------------------|------|
| External field length                   |                                             |      |
| defined                                 | 2.3.1                                       | 2-3  |
| editing examples                        | Table 2-2                                   | 2-7  |
| O and B counted                         | 2.3.1                                       | 2-2  |
| screen layout                           | 4.1.1                                       | 4-3  |
| <b>F</b>                                |                                             |      |
| Field I/O directions                    | See I/O directions and direction of fields. |      |
| Field response                          |                                             |      |
| dialog screen 2                         | 3.5.2                                       | 3-31 |
| sublist screen                          | 6.1.2.7                                     | 6-9  |
| Field type attributes                   |                                             |      |
| CREATE-FROM and MODIFY functions        | 5.1                                         | 5-1  |
| defined                                 | 2.1                                         | 2-1  |
| modify screen                           | 5.1.1.2                                     | 5-3  |
| rules                                   | 2.2                                         | 2-1  |
| specifying                              | 4.1.2                                       | 4-5  |
| File specifications form (RPG II)       | 7.3.1                                       | 7-5  |
| Fixed insertion, editing numeric fields | 2.3.2                                       | 2-3  |
| Flag, conditional indicator             | 3.4.1                                       | 3-12 |

| Term                                       | Reference           | Page         |
|--------------------------------------------|---------------------|--------------|
| Floating insertion, editing numeric fields | 2.3.2               | 2-4          |
| Format                                     | See screen formats. |              |
| Format display                             | 6.1.2.5             | 6-6          |
| Format names                               |                     |              |
| FSTATUS command                            | 3.2.3               | 3-5          |
| home screen                                | 3.2                 | 3-2          |
| list screen                                | 6.1.2.6             | 6-6          |
| new                                        | 3.2.2               | 3-4          |
| old                                        | 3.2.3               | 3-4          |
| summary screen                             | 6.1.2.1             | 6-3          |
| <b>FORTRAN</b>                             |                     |              |
| program using screen formats               | 7.3.3               | 7-26         |
| USE SFS statement                          | A.1                 | A-2          |
| FSTATUS command, existing format names     | 3.2.3               | 3-5          |
| Function/command key display               | 6.1.2.4<br>6.1.2    | 6-5<br>6-2   |
| Function/command key screen                |                     |              |
| characteristics screen                     | 3.3.10              | 3-10         |
| function/command key display               | 6.1.2.4             | 6-5          |
| function keys                              | Appendix C          |              |
| use                                        | 3.4<br>3.4.7        | 3-11<br>3-22 |
| Function keys                              |                     |              |
| BAL                                        | 7.3.4.3             | 7-29         |
| characteristics screen                     | 3.3<br>3.3.10       | 3-5<br>3-10  |
| COBOL                                      | 7.3.2.3             | 7-21         |
| dialog screens                             | 3.5                 | 3-25         |
| function/command key display               | 6.1.2.4             | 6-5          |
| function/command key screen                | 3.4.7               | 3-22         |
| home screen                                | 3.2                 | 3-2          |
| use                                        | Appendix C          |              |

| Term                               | Reference  | Page | Term                                      | Reference                   | Page |
|------------------------------------|------------|------|-------------------------------------------|-----------------------------|------|
| <b>H</b>                           |            |      | <b>I</b>                                  |                             |      |
| HELP function                      |            |      | I - for input, initial I/O screen         | 4.1.3                       | 4-7  |
| dialog screens                     | 3.5        | 3-25 | Imperative macroinstructions, BAL         | See BAL.                    |      |
| prompting by SFG                   | 1.3.1      | 1-7  | IMS programming considerations            | 7.3.6                       | 7-35 |
| requesting screens                 | 3.8        | 3-44 | Indicators                                |                             |      |
|                                    | Appendix B |      | BAL                                       | 7.3.4.3                     | 7-29 |
| Help screens                       | Appendix B |      | COBOL                                     | 7.3.2.2                     | 7-18 |
| Home screen                        |            |      | conditional                               | See conditional indicators. |      |
| CREATE-FROM function               | 3.2.1      | 3-3  | display retention                         | 3.4.6                       | 3-21 |
|                                    | 3.2.2      | 3-4  | erase                                     | 3.4.2                       | 3-15 |
|                                    | 5.1        | 5-1  | option                                    | 3.4.7                       | 3-22 |
|                                    | 5.2        | 5-6  |                                           | 7.3.2.2                     | 7-18 |
| CREATE function                    | 3.2.1      | 3-2  | protection                                | 3.5.6                       | 3-37 |
|                                    | 3.2.2      | 3-4  | replenish                                 | 3.4.2                       | 3-13 |
|                                    | 3.3.3      | 3-6  | response                                  | 3.4.7                       | 3-22 |
|                                    | 4.1        | 4-1  |                                           | 7.3.2.2                     | 7-18 |
| cursor positioning and overwriting | 3.6        | 3-41 |                                           | 7.3.4.3                     | 7-29 |
| DELETE function                    | 3.2.1      | 3-3  | retain                                    | 3.5.3                       | 3-34 |
|                                    | 3.2.3      | 3-4  |                                           | 3.5.5                       | 3-37 |
|                                    | 6.2        | 6-13 |                                           | 3.5.6                       | 3-38 |
| display functions                  | 6.1        | 6-1  | RPG II                                    | 3.4.7                       | 3-23 |
| functions provided by SFG          | 3.2.1      | 3-2  | special display properties                | 3.5.4                       | 3-35 |
| LIST function                      | 3.2.1      | 3-3  | syntax                                    | 3.4.1                       | 3-12 |
|                                    | 3.2.3      | 3-4  |                                           | 3.5.3                       | 3-33 |
|                                    | 6.1.2      | 6-1  |                                           | 3.5.4                       | 3-35 |
| MODIFY function                    | 3.2.1      | 3-3  |                                           | 3.5.5                       | 3-37 |
|                                    | 3.2.3      | 3-4  |                                           | 3.5.6                       | 3-37 |
|                                    | 5.1        | 5-1  | Initial I/O screen                        | 4.1.3                       | 4-7  |
| new format name and library        | 3.2.2      | 3-4  | Input data                                |                             |      |
| old format name and library        | 3.2.3      | 3-4  | transfer by screen format                 |                             |      |
| sample format                      | 4.2.1.1    | 4-12 | coordinator                               | 1.2                         | 1-4  |
| SHOW function                      | 3.2.1      | 3-3  | variable data fields                      | 1.3.2.2                     | 1-9  |
|                                    | 3.2.3      | 3-4  | Input fields                              | See I/O fields.             |      |
|                                    | 6.1.1      | 6-1  | Input format specifications form (RPG II) | 7.3.1                       | 7-5  |
| SPOOL function                     | 3.2.1      | 3-3  |                                           | Fig. 7-1                    | 7-7  |
|                                    | 3.2.3      | 3-4  |                                           | 7.3.6                       | 7-35 |
|                                    | 6.1.3      | 6-13 | Input record screen                       | 6.1.2                       | 6-2  |
| TERMINATE function                 | 3.2.1      | 3-3  |                                           | 6.1.2.9                     | 6-12 |
|                                    | 6.2        | 6-13 | Insertion editing                         |                             |      |
|                                    |            |      | alphabetic and alphanumeric fields        | 2.3.1                       | 2-2  |
|                                    |            |      | dialog screen 1                           | Table 3-2                   | 3-25 |
|                                    |            |      |                                           | 3.5.1                       | 3-26 |
|                                    |            |      | numeric fields                            | 2.3.2                       | 2-3  |

| Term                                                                 | Reference                 | Page |
|----------------------------------------------------------------------|---------------------------|------|
| <b>Intensity</b>                                                     |                           |      |
| alternate display properties                                         | Appendix E                |      |
| device support                                                       | Appendix E                |      |
| dialog screen 4                                                      | 3.5.4                     | 3-34 |
| display constants                                                    | 3.4.4                     | 3-18 |
| display property                                                     | 3.4.4                     | 3-17 |
|                                                                      | Table 3-4                 | 3-36 |
| error message screen                                                 | 3.4.5                     | 3-20 |
| fields in error                                                      | 3.4.4                     | 3-18 |
| input fields                                                         | 3.4.4                     | 3-17 |
| output fields                                                        | 3.4.4                     | 3-17 |
| selections                                                           | Table 3-4                 | 3-36 |
| special display properties                                           | 3.5.4                     | 3-34 |
| special display screen                                               | 3.4.4                     | 3-17 |
| <b>Internal field length</b>                                         |                           |      |
| defined                                                              | 2.3.1                     | 2-3  |
| dialog screen 1                                                      | Table 3-2                 | 3-25 |
|                                                                      | 3.5.1                     | 3-26 |
| editing examples                                                     | Table 2-2                 | 2-7  |
| O and B not counted                                                  | 2.3.1                     | 2-2  |
| results of editing                                                   | 2.4                       | 2-4  |
| <b>I/O directions</b>                                                |                           |      |
| bidirectional                                                        | 4.1.3                     | 4-7  |
| dialog screen 1                                                      | 3.5                       | 3-26 |
|                                                                      | 3.5.1                     | 3-26 |
| duplicate lines                                                      | 4.4                       | 4-24 |
| initial I/O screen                                                   | 4.1.3                     | 4-7  |
| input                                                                | 4.1.3                     | 4-7  |
| modify screen                                                        | 5.1.1                     | 5-2  |
| output                                                               | 4.1.3                     | 4-7  |
| See also bidirectional fields, I/O fields, and variable data fields. |                           |      |
| <b>I/O fields</b>                                                    |                           |      |
| bidirectional                                                        | See bidirectional fields. |      |
| classifications                                                      | 1.3.2.2                   | 1-9  |
| COBOL                                                                | 7.3.2.1                   | 7-10 |
| conditional display                                                  | 3.5.3                     | 3-33 |
| conditional erase/replenish screen                                   | 3.4.2                     | 3-14 |
| conditional protection                                               | 3.5.6                     | 3-37 |
| conditional retention                                                | 3.5.5                     | 3-36 |
| default value                                                        | 3.5.2                     | 3-31 |
| dialog screen 1                                                      | 3.5                       | 3-24 |
|                                                                      | 3.5.1                     | 3-26 |
| dialog screen 2                                                      | 3.5.2                     | 3-31 |
| dialog screen 3                                                      | 3.5.3                     | 3-33 |
| dialog screen 6                                                      | 3.5.6                     | 3-37 |
| dialog screen 7                                                      | 3.5.7                     | 3-38 |
| display properties                                                   | 3.4.4                     | 3-17 |
| display retention screen                                             | 3.4.6                     | 3-21 |
| editing characters                                                   | 1.3.2.3                   | 1-11 |
| editing rules for signed output fields                               | Table 2-1                 | 2-6  |

| Term                                   | Reference | Page  |
|----------------------------------------|-----------|-------|
| emphasis                               | 3.4.4     | 3-17  |
|                                        | 3.4.6     | 3-21  |
| ESCORT                                 | 7.3.5     | 7-34  |
| field response                         | 3.5.2     | 3-31  |
| FORTTRAN                               | 7.3.3     | 7-26a |
| function/command key screen            | 3.4.7     | 3-22  |
| IMS                                    | 7.3.6     | 7-35  |
| initial I/O screen                     | 4.1.3     | 4-7   |
| intensity                              | 3.4.4     | 3-17  |
|                                        | 3.4.6     | 3-21  |
| LIST function                          | 6.1.2     | 6-1   |
| list screen                            | 6.1.2.6   | 6-7   |
| nondisplay screen                      | 3.4.8     | 3-23  |
| protected and unprotected parts        | 1.3.2.3   | 1-11  |
| range checking                         | 3.5.7     | 3-38  |
| replenishing value                     | 3.5.2     | 3-31  |
| RPG II                                 | 7.3.1     | 7-5   |
| SHOW function                          | 6.1.1     | 6-1   |
| special display properties             | 3.5.4     | 3-35  |
| specifying directions                  | 4.1.3     | 4-7   |
| specifying type attributes and editing | 4.1.2     | 4-5   |
| type attributes                        | 2.1       | 2-1   |
|                                        | 2.2       | 2-2   |
| See also variable data fields.         |           |       |
| I/O screen display                     | 4.1.3     | 4-7   |

**J**

|                       |     |     |
|-----------------------|-----|-----|
| <b>Job control</b>    |     |     |
| multiple workstations | A.2 | A-6 |
| statements            | 7.2 | 7-1 |
| USE SFS statement     | A.1 | A-1 |

**K**

|                          |                    |      |
|--------------------------|--------------------|------|
| <b>Katakana</b>          |                    |      |
| default currency sign    | 3.4.3              | 3-16 |
| no lowercase translation | 3.3.1              | 3-6  |
|                          | 4.1.1              | 4-5  |
| Keys, function           | See function keys. |      |

| Term                               | Reference | Page | Term                                | Reference | Page |
|------------------------------------|-----------|------|-------------------------------------|-----------|------|
| <b>L</b>                           |           |      | <b>M</b>                            |           |      |
| Language, characteristics screen   | 3.3.2     | 3-6  | Minus sign                          |           |      |
| Library file (user-assigned)       |           |      | editing rules for signed output     |           |      |
| DVC/LFD sequence                   | 7.2       | 7-1  | fields                              | Table 2-1 | 2-6  |
| modifying formats                  | A.1       | A-3  | fixed and floating insertion in     |           |      |
|                                    | 5.1       | 5-1  | numeric fields                      | 2.3.2     | 2-3  |
| LIST function                      |           |      | MIRAM user library, storing formats | 1.2       | 1-4  |
| conditional values screen          | 6.1.2.8   | 6-11 | MODIFY function                     |           |      |
| display control and error message  |           |      | characteristics screen              | 3.3.2     | 3-6  |
| display                            | 6.1.2.3   | 6-5  | dialog screen 1                     | 3.5.1     | 3-26 |
| format display                     | 6.1.2.5   | 6-6  | formats                             | 3.2.1     | 3-3  |
| function/command key display       | 6.1.2.4   | 6-5  |                                     | 3.3.2     | 3-6  |
| home screen                        | 3.2       | 3-1  |                                     | 3.3.3     | 3-6  |
|                                    | 3.2.1     | 3-3  |                                     | 5.1       | 5-1  |
|                                    | 3.2.3     | 3-4  | home screen                         | 3.2       | 3-2  |
| input record screen                | 6.1.2.9   | 6-12 |                                     | 3.2.1     | 3-3  |
| list screen                        | 6.1.2.6   | 6-6  |                                     | 3.2.3     | 3-4  |
| old format name and library        | 3.2.3     | 3-4  | language                            | 3.3.2     | 3-6  |
| output record screen               | 6.1.2.10  | 6-12 | modify screen                       | 5.1.1     | 5-2  |
| special editing display            | 6.1.2.2   | 6-4  | old format name and library         | 3.2.3     | 3-4  |
| sublist screen                     | 6.1.2.7   | 6-9  | original and overlay formats        | 3.3.3     | 3-6  |
| summary screen                     | 6.1.2.1   | 6-3  | summary                             | 5.2       | 5-6  |
| use                                | 3.2.1     | 3-3  |                                     |           |      |
|                                    | 6.1       | 6-1  | Modify screen                       |           |      |
|                                    | 6.1.2     | 6-1  | completing                          | 5.1.1     | 5-2  |
| List screen                        | 6.1.2     | 6-2  | dialog screen 1                     | 3.5.1     | 3-27 |
|                                    | 6.1.2.6   | 6-6  | modification summary                | 5.2       | 5-6  |
|                                    | 6.1.2.7   | 6-9  | option 1 (CHANGE TEMPLATE)          | 5.1.1.1   | 5-3  |
| Lowercase translation              |           |      | option 2 (CHANGE TYPE)              | 5.1.1.2   | 5-3  |
| characteristics screen             | 3.3       | 3-5  | option 3 (CHANGE I/O)               | 5.1.1.3   | 5-4  |
|                                    | 3.3.1     | 3-6  | option 7 (CHANGE TEMPLATE ONLY -    |           |      |
| range checking                     | 3.5.7     | 3-39 | NO INTERNAL CHANGES)                | 5.1.1.4   | 5-4  |
| Low-intensity display              |           |      |                                     |           |      |
| protected and unprotected parts of |           |      |                                     |           |      |
| screen format                      | 1.3.2.3   | 1-11 |                                     |           |      |
| special display screen             | 3.4.4     | 3-17 |                                     |           |      |

| Term                                        | Reference       | Page         | Term                                                   | Reference                  | Page               |
|---------------------------------------------|-----------------|--------------|--------------------------------------------------------|----------------------------|--------------------|
| <b>N</b>                                    |                 |              | <b>O</b>                                               |                            |                    |
| New format name and library,<br>home screen | 3.2<br>3.2.2    | 3-2<br>3-4   | O - for output, I/O screen                             | 4.1.3                      | 4-8                |
| Nondisplay screen                           |                 |              | Old format name and library,<br>home screen            | 3.2<br>3.2.3               | 3-2<br>3-4         |
| characteristics screen                      | 3.3.11          | 3-11         | OPEN (open a file), BAL imperative<br>macroinstruction | 7.3.4.1                    | 7-27               |
| use                                         | 3.4<br>3.4.8    | 3-11<br>3-23 | Option indicators                                      |                            |                    |
| Nondisplayed field                          |                 |              | COBOL                                                  | 7.3.2.2                    | 7-18               |
| characteristics screen                      | 3.3             | 3-5          | function/command key screen                            | 3.4.7                      | 3-22               |
| nondisplay screen                           | 3.3.11<br>3.4.8 | 3-11<br>3-23 | Optional screens                                       |                            |                    |
| Numeric fields                              |                 |              | characteristics screen                                 | Table 3-1                  | 3-11               |
| default values for dialog screens           | 3.5.8           | 3-40         | conditional erase/replenish screen                     | 3.4.2                      | 3-13               |
| editing                                     | 2.3.2           | 2-3          | conditional indicators                                 | 3.4.1                      | 3-12               |
| range checking                              | 3.5.7           | 3-38         | dialog screens                                         | Table 3-1                  | 3-11               |
| rules                                       | 2.2             | 2-1          | display retention screen                               | 3.4.6                      | 3-21               |
| type attributes                             | 4.1.2           | 4-5          | edit screen                                            | 3.4.3                      | 3-16               |
| variable data fields                        | 1.3.2.2         | 1-9          | error message screen                                   | 3.4.5                      | 3-18               |
| 9 character                                 |                 |              | function/command key screen                            | 3.4.7                      | 3-22               |
| indicating alphanumeric fields              | 2.2             | 2-2          | nondisplay screen                                      | 3.4.8                      | 3-23               |
| indicating numeric fields                   | 2.2             | 2-2          | special display screen                                 | 3.4.4                      | 3-17               |
|                                             |                 |              | use                                                    | 3.4                        | 3-11               |
|                                             |                 |              | Original format, characteristics screen                | 3.3<br>3.3.3<br>4.1        | 3-5<br>3-6<br>4-2  |
|                                             |                 |              | Output data, transfer by SFC                           | 1.2                        | 1-4                |
|                                             |                 |              | Output fields                                          | See I/O<br>fields.         |                    |
|                                             |                 |              | Output format specifications form<br>(RPG II)          | 7.3.1<br>Fig. 7-1<br>7.3.6 | 7-6<br>7-8<br>7-36 |
|                                             |                 |              | Output record screen                                   | 6.1.2<br>6.1.2.10          | 6-2<br>6-12        |
|                                             |                 |              | Overlay formats                                        |                            |                    |
|                                             |                 |              | advantage                                              | 4.3.1                      | 4-20a              |
|                                             |                 |              | characteristics screen                                 | 3.3.3                      | 3-6                |
|                                             |                 |              | creation                                               | 4.3.2                      | 4-22               |
|                                             |                 |              | SOE character                                          | 4.3.2                      | 4-23               |
|                                             |                 |              | use                                                    | 4.3                        | 4-20a              |
|                                             |                 |              |                                                        | 4.3.1                      | 4-20a              |

| Term                                                         | Reference | Page | Term                                                        | Reference | Page |
|--------------------------------------------------------------|-----------|------|-------------------------------------------------------------|-----------|------|
| <b>Overwriting</b>                                           |           |      | <b>R</b>                                                    |           |      |
| completing SFG screens                                       | 3.6       | 3-41 | Range checking                                              |           |      |
| dialog screen 1                                              | 3.5.1     | 3-27 | default values                                              | 3.5.8     | 3-40 |
| dialog screen 2                                              | 3.5.2     | 3-31 | dialog screen 1                                             | 3.5.1     | 3-28 |
| dialog screen 4                                              | 3.5.4     | 3-35 | dialog screen 7                                             | 3.5.7     | 3-38 |
| dialog screen 7                                              | 3.5.7     | 3-38 | list screen                                                 | 6.1.2.6   | 6-8  |
| edit screen                                                  | 3.4.3     | 3-16 | sublist screen                                              | 6.1.2.7   | 6-9  |
| initiating dialogs                                           | 4.1.3     | 4-8  | Replenish indicator, conditional/<br>erase replenish screen | 3.4.2     | 3-13 |
| I/O screen                                                   | 4.1.3     | 4-8  | Replenishing value                                          |           |      |
| modifying formats                                            | 5.1.1     | 5-2  | conditional erase/replenish screen                          | 3.4.2     | 3-13 |
| <b>P</b>                                                     |           |      | default values                                              | 3.5.8     | 3-40 |
| <b>Passes</b>                                                |           |      | dialog screen 2                                             | 3.5.2     | 3-31 |
| dialog screens                                               | 3.5       | 3-24 | edit screen                                                 | 3.4.3     | 3-16 |
|                                                              | 4.1.3     | 4-7  | list screen                                                 | 6.1.2.6   | 6-8  |
|                                                              | 4.2.1     | 4-11 | sublist screen                                              | 6.1.2.7   | 6-9  |
| display constants                                            | 1.3.2.1   | 1-9  | Response indicators                                         |           |      |
| optional screens                                             | 4.2       | 4-10 | BAL                                                         | 7.3.4.3   | 7-29 |
|                                                              | 4.2.1.2   | 4-13 | COBOL                                                       | 7.3.2.2   | 7-18 |
|                                                              | 4.2.1.3   | 4-14 | function/command key screen                                 | 3.4.7     | 3-22 |
| pass 1 - providing the layout                                | 4.1.1     | 4-3  | RPG II                                                      | 3.4.7     | 3-23 |
|                                                              | 4.2.1.4   | 4-14 | RPG II                                                      |           |      |
| pass 2 - specifying type attributes<br>and editing           | 4.1.2     | 4-5  | file specifications form                                    | 7.3.1     | 7-5  |
|                                                              | 4.2.1.5   | 4-15 | input format specifications form                            | 7.3.1     | 7-5  |
| pass 3 - specifying I/O directions<br>and initiating dialogs | 4.1.3     | 4-7  | output specifications form                                  | 7.3.1     | 7-6  |
|                                                              | 4.2.1.6   | 4-16 | response indicators                                         | 3.4.7     | 3-23 |
| review                                                       | 4.1.3     | 4-9  | screen formats                                              | 7.3.1     | 7-5  |
| specifying field type                                        | 1.4       | 1-13 | Fig. 7-1                                                    |           | 7-7  |
| variable data fields                                         | 1.3.2.2   | 1-9  | USE SFS statement                                           | A.1       | A-2  |
| <b>Plus sign</b>                                             |           |      | RV SFGEN command, activating the SFG                        | 3.1       | 3-1  |
| editing rules for signed output<br>fields                    | Table 2-1 | 2-6  |                                                             |           |      |
| fixed and floating insertion in<br>numeric fields            | 2.3.2     | 2-3  |                                                             |           |      |
| <b>Protected fields</b>                                      |           |      |                                                             |           |      |
| description                                                  | 1.3.2.3   | 1-11 |                                                             |           |      |
| SHOW function                                                | 6.1.1     | 6-1  |                                                             |           |      |

| Term                                      | Reference                 | Page | Term                                       | Reference            | Page  |
|-------------------------------------------|---------------------------|------|--------------------------------------------|----------------------|-------|
| <b>S</b>                                  |                           |      |                                            |                      |       |
| Screen format coordinator                 |                           |      |                                            |                      |       |
| BAL                                       | 7.3.4.5                   | 7-33 | duplicate lines                            | 4.4                  | 4-24  |
| COBOL                                     | 7.3.2.2                   | 7-18 | edit screen                                | 3.4.3                | 3-16  |
| conditional erase/replenish screen        | 3.4.2                     | 3-14 | erasing option                             | See erasing option.  |       |
| conditional indicators                    | 3.4.1                     | 3-12 | error and warning messages                 | 3.7                  | 3-43  |
| dialog screen 2                           | 3.5.2                     | 3-31 | error message field                        | 3.3.8                | 3-10  |
| dialog screen 5                           | 3.5.5                     | 3-37 | error message screen                       | 3.4.5                | 3-18  |
| dialog screen 7                           | 3.5.7                     | 3-39 | error retry counts                         | 3.3.5                | 3-8   |
| display retention                         | 3.3.9                     | 3-10 | format display                             | 6.1.2.5              | 6-6   |
| display retention screen                  | 3.4.6                     | 3-21 | function                                   | 1.2                  | 1-4   |
| edit screen                               | 3.4.3                     | 3-17 |                                            | Fig. 1-4             | 1-5   |
| error message screen                      | 3.4.5                     | 3-19 |                                            | 1.3                  | 1-6   |
| FORTRAN                                   | 7.3.3                     | 7-26 |                                            | 3.2.1                | 3-2   |
| function                                  | 1.2                       | 1-4  |                                            | 5.1                  | 5-1   |
|                                           | Fig. 1-4                  | 1-5  |                                            | 6.1                  | 6-1   |
|                                           | 1.4                       | 1-13 |                                            | 6.2                  | 6-13  |
| function/command key screen               | 3.4.7                     | 3-22 | function/command key display               | 6.1.2.4              | 6-5   |
| job control                               | 7.2                       | 7-1  | function/command key screen                | 3.4.7                | 3-22  |
| option indicators                         | 3.4.7                     | 3-22 | function keys                              | 3.3.10               | 3-10  |
| program considerations                    | 7.3                       | 7-5  | HELP function                              | 3.8                  | 3-44  |
| programmer responsibilities               | 7.1                       | 7-1  | help screens                               | Appendix B           |       |
| range checking                            | 3.5.7                     | 3-39 | home screen                                | 3.2                  | 3-1   |
| REPLENISH SCREEN option                   | 3.3.4                     | 3-7  | initiating dialogs                         | 4.1.3                | 4-7   |
| response indicators                       | 3.4.7                     | 3-22 | input record screen                        | 6.1.2.9              | 6-12  |
| RPG II                                    | 7.3.1                     | 7-5  | LIST function                              | See LIST function.   |       |
| software element of screen                |                           |      | list screen                                | 6.1.2.6              | 6-6   |
| format services                           | 1.2                       | 1-4  | lowercase translation                      | 3.3.1                | 3-6   |
| UNLOCK KEYBOARD option                    | 3.3.4                     | 3-8  | MODIFY function                            | See MODIFY function. |       |
| workstation considerations                | 7.3.2.4                   | 7-26 |                                            | 5.1                  | 5-1   |
|                                           | 7.4                       | 7-37 | modifying formats                          | 5.2                  | 5-6   |
| Screen format generator                   |                           |      | new format name and library                | 3.2.2                | 3-4   |
| activating                                | 3.1                       | 3-1  | nondisplay screen                          | 3.4.8                | 3-23  |
| alphabet                                  | 3.3.2                     | 3-6  | nondisplayed field                         | 3.3.11               | 3-11  |
| basic screen formats                      | 4.1                       | 4-1  | old format name and library                | 3.2.3                | 3-4   |
| characteristics screen                    | 3.3                       | 3-5  | optional screens                           | 3.4                  | 3-11  |
| complex screen formats                    | 4.2                       | 4-10 |                                            | 4.2                  | 4-10  |
| conditional erase/replenish screen        | 3.4.2                     | 3-13 | original formats                           | 3.3.3                | 3-6   |
| conditional indicators                    | 3.4.1                     | 3-12 |                                            | 4.1                  | 4-2   |
| conditional values screen                 | 6.1.2.8                   | 6-11 | output record screen                       | 6.1.2.10             | 6-12  |
| CREATE-FROM function                      | See CREATE-FROM function. |      | overlay formats                            | 3.3.3                | 3-6   |
| CREATE function                           | See CREATE function.      |      |                                            | 4.3.2                | 4-22  |
| cursor positioning and overwriting        | 3.6                       | 3-41 |                                            | 4.3.1                | 4-20a |
| default values for dialog screens         | 3.5.8                     | 3-40 | passes                                     | See passes.          |       |
| DELETE function                           | See DELETE function.      |      | prompting                                  | 1.3.1                | 1-7   |
|                                           | function.                 |      | sample format                              | 4.2.1                | 4-10  |
| dialog screens                            | 3.5                       | 3-24 | SHOW function                              | See SHOW function.   |       |
|                                           | 4.2                       | 4-10 |                                            |                      |       |
| display control and error message display | 6.1.2.3                   | 6-5  | software element of screen format services | 1.2                  | 1-4   |
| display functions                         | See display functions.    |      | special display control                    | 3.3.7                | 3-9   |
| display retention                         | 3.3.9                     | 3-10 | special display screen                     | 3.4.4                | 3-17  |
| display retention screen                  | 3.4.6                     | 3-21 | special editing characters                 | 3.3.6                | 3-9   |
|                                           |                           |      | special editing display                    | 6.1.2.2              | 6-4   |
|                                           |                           |      | SPOOL function                             | See SPOOL function.  |       |



| Term                               | Reference                      | Page | Term                                     | Reference                      | Page  |
|------------------------------------|--------------------------------|------|------------------------------------------|--------------------------------|-------|
| Screen format generator (cont)     |                                |      | overlay                                  | 3.3.3                          | 3-6   |
| sublist screen                     | 6.1.2.7                        | 6-9  |                                          | 4.3                            | 4-20a |
| summary screen                     | 6.1.2.1                        | 6-3  | program considerations                   | 7.3                            | 7-5   |
| TERMINATE function                 | See TERMINATE function.        |      | program use                              | 7.1                            | 7-1   |
| unlocking option                   | 3.3.4                          | 3-7  | programmer responsibilities              | 7.1                            | 7-1   |
|                                    | 3.4.2                          | 3-15 | prompting                                | 1.3.1                          | 1-7   |
|                                    | 4.1                            | 4-3  | protected and unprotected parts          | 1.3.2.3                        | 1-11  |
|                                    | 4.2.1.1                        | 4-13 | RPG II                                   | 7.3.1                          | 7-5   |
|                                    |                                |      | sample                                   | 4.2.1                          | 4-10  |
| Screen format services             |                                |      | SYSRES volume                            | 1.2                            | 1-4   |
| format creation                    | 1.2                            | 1-1  | system library (\$Y\$FMT)                | 1.2                            | 1-4   |
| format use                         | 1.2                            | 1-1  | template screen                          | Fig. 4-1                       | 4-3   |
| interactive component of OS/3      | 1.2                            | 1-1  | type attribute screen                    | Fig. 4-2                       | 4-6   |
| screen format coordinator (SFG)    | See screen format coordinator. |      | type attributes and editing              | 4.1.2                          | 4-5   |
| screen format generator (SFG)      | See screen format generator.   |      | variable data fields                     | 1.3.2.2                        | 1-9   |
| software elements                  | 1.2                            | 1-4  | SFC                                      | See screen format coordinator. |       |
| Screen formats                     |                                |      | SFG                                      | See screen format generator.   |       |
| BAL                                | 7.3.4                          | 7-27 | SFS                                      | See screen format services.    |       |
| basic                              | 4.1                            | 4-1  |                                          |                                |       |
| characteristics screen             | See characteristics screen.    |      | SHOW function                            |                                |       |
| COBOL                              | 7.3.2                          | 7-9  | home screen                              | 3.2                            | 3-1   |
| complex                            | 4.2                            | 4-10 |                                          | 3.2.1                          | 3-3   |
| components                         | 1.3.2                          | 7-9  | old format name and library use          | 3.2.3                          | 3-4   |
| creating                           | 1.3.1                          | 1-7  |                                          | 3.2.1                          | 3-3   |
| cursor positioning and overwriting | 3.6                            | 3-41 |                                          | 6.1                            | 6-1   |
| deleting                           | 1.3.1                          | 1-7  |                                          | 6.1.1                          | 6-1   |
|                                    | 6.2                            | 6-13 | Signed output fields, editing rules      | Table 2-1                      | 2-6   |
|                                    | 3.2.1                          | 3-3  |                                          |                                |       |
|                                    | 3.2.3                          | 3-4  | Simple insertion, editing numeric fields | 2.3.2                          | 2-3   |
| dialogs                            | 3.5                            | 3-24 |                                          |                                |       |
| display constants                  | 1.3.2.1                        | 1-9  | Slash, dialog screen 1                   | 3.5.1                          | 3-30  |
| displaying                         | 1.2                            | 1-1  |                                          |                                |       |
|                                    | 1.3.1                          | 1-7  | SOE character, overlay format            | 4.3.2                          | 4-23  |
|                                    | 6.1                            | 6-7  |                                          |                                |       |
| duplicate lines                    | 4.4                            | 4-24 | Special display control                  |                                |       |
| error and warning messages         | 3.7                            | 3-43 | characteristics screen                   | 3.3                            | 3-5   |
| error indication                   | 1.3.1                          | 1-7  |                                          | 3.3.7                          | 3-9   |
| ESCORT                             | 7.3.5                          | 7-34 | special display screen                   | 3.4.4                          | 3-17  |
| FORTRAN                            | 7.3.3                          | 7-26 |                                          |                                |       |
| home screen                        | See home screen.               |      | Special display screen                   |                                |       |
|                                    | 1.2                            | 1-1  | display control and error                |                                |       |
| input                              | 7.2                            | 7-1  | message display                          | 6.1.2.3                        | 6-5   |
| job control                        | 1.2                            | 1-4  | display properties                       | 3.4.4                          | 3-17  |
| layout                             | 4.1.1                          | 4-3  |                                          | Appendix E                     |       |
|                                    | 1.2                            | 1-4  | use                                      | 3.4                            | 3-11  |
| MIRAM user library                 | 1.3.1                          | 1-7  |                                          | 3.4.4                          | 3-17  |
| modifying                          | 3.2.1                          | 3-3  |                                          |                                |       |
|                                    | 5.1                            | 5-1  |                                          |                                |       |
| original                           | 3.3.3                          | 3-6  |                                          |                                |       |
|                                    | 4.1                            | 4-1  |                                          |                                |       |
| output                             | 1.2                            | 1-1  |                                          |                                |       |

| Term                                     | Reference | Page | Term | Reference | Page |
|------------------------------------------|-----------|------|------|-----------|------|
| Special editing characters               |           |      |      |           |      |
| characteristics screen                   | 3.3       | 3-1  |      |           |      |
| edit screen                              | 3.3.6     | 3-9  |      |           |      |
| type attribute screen                    | 3.4.3     | 3-16 |      |           |      |
|                                          | 4.1.2     | 4-7  |      |           |      |
| Special editing display                  | 6.1.2.2   | 6-4  |      |           |      |
|                                          | 6.1.2     | 6-2  |      |           |      |
| SPOOL function                           |           |      |      |           |      |
| home screen                              | 3.2       | 3-1  |      |           |      |
|                                          | 3.2.1     | 3-3  |      |           |      |
| old format name and library use          | 3.2.3     | 3-4  |      |           |      |
|                                          | 3.2.1     | 3-3  |      |           |      |
|                                          | 6.1       | 6-1  |      |           |      |
|                                          | 6.1.3     | 6-13 |      |           |      |
| Start of entry character, overlay format | 4.3.2     | 4-23 |      |           |      |
| Sublist screen                           | 6.1.2     | 6-2  |      |           |      |
|                                          | 6.1.2.7   | 6-9  |      |           |      |
| Summary screen                           | 6.1.2.1   | 6-3  |      |           |      |
| Suppression/replacement editing          | 2.3.2     | 2-3  |      |           |      |
| Syntax, indicator                        |           |      |      |           |      |
| conditional                              | 3.4.1     | 3-12 |      |           |      |
| dialog screen 3                          | 3.5.3     | 3-33 |      |           |      |
| dialog screen 4                          | 3.5.4     | 3-35 |      |           |      |
| dialog screen 5                          | 3.5.5     | 3-37 |      |           |      |
| dialog screen 6                          | 3.5.6     | 3-37 |      |           |      |
| SYSRES volume, storing formats           | 1.2       | 1-4  |      |           |      |
| System library (\$Y\$FMT)                |           |      |      |           |      |
| home screen                              | 3.2.2     | 3-4  |      |           |      |
| storing formats                          | 1.2       | 1-4  |      |           |      |

## T

| Term                  | Reference | Page |
|-----------------------|-----------|------|
| Template screen       |           |      |
| duplicate lines       | 4.4       | 4-24 |
| overlay format        | 4.3.2     | 4-22 |
| pass 1                | 4.1.1     | 4-3  |
|                       | 4.2.1.4   | 4-14 |
| pass 2                | 4.1.2     | 4-5  |
|                       | 4.2.1.5   | 4-15 |
| pass 3                | 4.1.3     | 4-7  |
|                       | 4.2.1.6   | 4-16 |
| typical               | Fig. 4-1  | 4-3  |
| TERMINATE function    |           |      |
| home screen           | 3.2       | 3-1  |
|                       | 3.2.1     | 3-3  |
| use                   | 3.2.1     | 3-3  |
|                       | 6.1       | 6-1  |
|                       | 6.2       | 6-13 |
| Type attribute screen | 4.1.2     | 4-5  |
| Type attributes       |           |      |
| defined               | 2.1       | 2-1  |
| duplicate lines       | 4.4       | 4-24 |
| format display        | 6.1.2.5   | 6-6  |
| LIST function         | 6.1.2     | 6-2  |
| rules                 | 2.1       | 2-1  |
| SHOW function         | 6.1.1     | 6-1  |
| specifying            | 4.1.2     | 4-5  |
| typical screen        | Fig. 4-2  | 4-6  |

## U

| Term                  | Reference | Page |
|-----------------------|-----------|------|
| Underlines            |           |      |
| dialog screen 2       | 3.5.2     | 3-31 |
| edit screen           | 3.4.3     | 3-16 |
| format layout         | 4.1.1     | 4-4  |
| initiating dialogs    | 4.1.3     | 4-8  |
| input fields          | 1.3.2.2   | 1-9  |
|                       | 2.4       | 2-5  |
| result of editing     | 2.4       | 2-4  |
| type attribute screen | 4.1.2     | 4-6  |
| variable data fields  | 1.3.2.2   | 1-9  |
|                       | 4.2.1.4   | 4-14 |

| Term                                              | Reference | Page |
|---------------------------------------------------|-----------|------|
| Unlocking option                                  |           |      |
| characteristics screen                            | 3.3       | 3-5  |
| conditional erase/replenish screen                | 3.3.4     | 3-7  |
| creating basic format                             | 3.4.2     | 3-15 |
| creating complex format                           | 4.1       | 4-3  |
| creating complex format                           | 4.2.1.1   | 4-13 |
| Unprotected fields                                |           |      |
| description                                       | 1.3.2.3   | 1-11 |
| SHOW function                                     | 6.1.1     | 6-1  |
| Uppercase alphabetical characters,<br>translation | 3.3.1     | 3-6  |
| USE SFS job control statement                     | 7.2       | 7-1  |
|                                                   | A.1       | A-1  |

**V**

|                                                  |         |      |
|--------------------------------------------------|---------|------|
| Variable data fields                             |         |      |
| considerations                                   | 2.1     | 2-1  |
| dialog screens                                   | 3.5     | 3-24 |
| display retention screen                         | 3.5.1   | 3-27 |
| duplicate lines                                  | 3.4.6   | 3-21 |
| edit characters                                  | 4.4     | 4-24 |
| IMS                                              | 4.1.2   | 4-3  |
| I/O direction                                    | 7.3.6   | 7-36 |
| layout                                           | 4.2.1.6 | 4-16 |
| pass 1                                           | 1.3.2.2 | 1-9  |
| pass 2                                           | 4.2.1.4 | 4-14 |
| pass 3                                           | 4.1.1   | 4-3  |
| protected and unprotected parts                  | 4.2.1.4 | 4-14 |
| RPG II                                           | 4.1.2   | 4-5  |
| type attributes                                  | 4.2.1.5 | 4-15 |
| type attributes                                  | 4.1.3   | 4-7  |
| type attributes                                  | 4.2.1.6 | 4-16 |
| type attributes                                  | 1.3.2.3 | 1-11 |
| type attributes                                  | 7.3.1   | 7-5  |
| type attributes                                  | 2.2     | 2-1  |
| type attributes                                  | 4.1.2   | 4-5  |
| type attributes                                  | 4.2.1.5 | 4-15 |
| See also bidirectional fields and<br>I/O fields. |         |      |

| Term                                   | Reference | Page |
|----------------------------------------|-----------|------|
| <b>W</b>                               |           |      |
| Warning messages                       | 3.7       | 3-43 |
| Workstation considerations             |           |      |
| interprogram communication<br>operator | 7.3.2.4   | 7-26 |
|                                        | 7.4       | 7-37 |

**X**

|                                                |     |     |
|------------------------------------------------|-----|-----|
| X character, indicating<br>alphanumeric fields | 2.2 | 2-2 |
|------------------------------------------------|-----|-----|

**Y**

|                                                 |       |      |
|-------------------------------------------------|-------|------|
| Yen sign, default currency sign<br>for Katakana | 3.4.3 | 3-16 |
|-------------------------------------------------|-------|------|

**Z**

|                                                                   |       |     |
|-------------------------------------------------------------------|-------|-----|
| Z character, suppression/replacement<br>editing in numeric fields | 2.3.2 | 2-3 |
| Zero character                                                    |       |     |
| editing alphabetic and<br>alphanumeric fields                     | 2.3.2 | 2-3 |
| insertion in numeric fields                                       | 2.3.2 | 2-3 |



## USER COMMENT SHEET

Your comments concerning this document will be welcomed by Sperry Univac for use in improving subsequent editions.

*Please note: This form is not intended to be used as an order blank.*

---

*(Document Title)*

---

*(Document No.)*

---

*(Revision No.)*

---

*(Update No.)*

### Comments:

Cut along line.

**From:**

---

*(Name of User)*

---

*(Business Address)*

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)  
Thank you for your cooperation

FOLD



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

**SPERRY UNIVAC**

ATTN.: SYSTEMS PUBLICATIONS

P.O. BOX 500  
BLUE BELL, PENNSYLVANIA 19424



CUT

FOLD