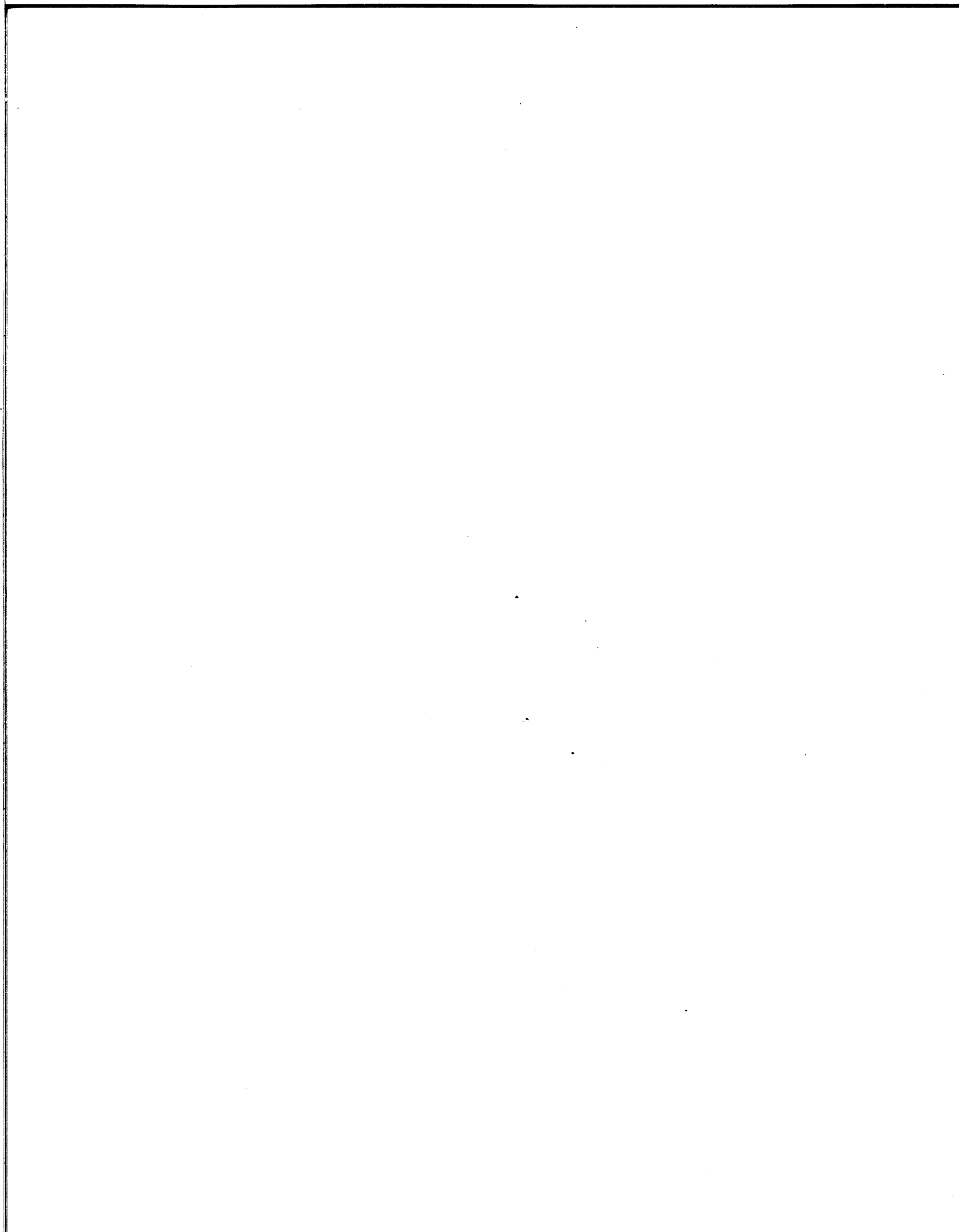




**PROXI™
COBOL Code Generator
User's Guide
(AOS)**



PROXI™
COBOL Code Generator
User's Guide
(AOS)

093-000280-00

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085-series) supplied with the software.

Ordering No. 093-000280
© Data General Corporation, 1981
All Rights Reserved
Printed in the United States of America
Revision 00, May 1981
Licensed Material - Property of Data General Corporation

NOTICE

Data General Corporation (DGC) has prepared this manual for use by DGC personnel, licensees, and customers. The information contained herein is the property of DGC and shall not be reproduced in whole or in part without DGC prior written approval.

The terms and conditions governing the sale of DGC hardware products and the licensing of DGC software consist solely of those set forth in the written contracts between DGC and its customers. No representation or other affirmation of fact contained in this document, including but not limited to statements regarding capacity, response-time performance, suitability for use or performance of products described herein shall be deemed to be a warranty by DGC for any purpose, or give rise to any liability of DGC whatsoever.

DGC reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

PROXI™
COBOL Code Generator
User's Guide
(AOS)
093-000280

Revision History:

Original Release - May 1981 (AOS Rev. 3.00) (AOS COBOL Rev. 3.00)

The following are trademarks of Data General Corporation, Westboro, Massachusetts:

U.S. Registered Trademarks		Trademarks		
DATAPREP	NOVA	AZ-TEXT	ECLIPSE MV/8000	SWAT
ECLIPSE	SUPERNOVA	DASHER	microNOVA	XODIAC
INFOS		DG/L	PROXI	

Preface

The AOS PROXI™ COBOL Code Generator is an interactive system that helps you build compilable source code for business-related applications.

Who Should Read This Manual

We've designed this manual for you, the PROXI user. You'll need a working knowledge of AOS COBOL before attempting to build programs using the PROXI system. Although you may not have to produce a single line of hand-written code, you must be able to work with data-file record fields, build a Working Storage area, and design output formats such as data-entry screens, reports and forms.

The PROXI system makes programming easier: it saves coding time, letting you concentrate on the design of the program.

How the Manual is Organized

We've designed this manual with two types of user in mind: the programmer who is learning the PROXI system, and the experienced PROXI programmer. For the learner, we present detailed explanations of what the PROXI system does and how to use its data-entry screens. For the experienced user, we provide quick-reference tables to summarize information and to make it available as a handy reference.

The manual is arranged as follows:

- | | |
|-----------|--|
| Chapter 1 | Introduces the PROXI system, presenting its features and capabilities. We give you a general overview of the product here. |
| Chapter 2 | Presents the PROXI Program Generator module. This chapter leads you through the series of data-entry screens that allows you to build a program framework. |
| Chapter 3 | Presents the PROXI Screen Generator module. This chapter explains how you design and build screen formats that communicate with the program operator. |

Chapter 4	Presents the PROXI File Definitions module. This chapter shows you how to generate COBOL code that describes the various datafiles your program uses.
Chapter 5	Explains how to compile your PROXI source code. It also supplies a "Do-It-Yourself" documentation form so that you can provide information to program operator.
Appendix A	Contains a glossary of terms.
Appendix B	Lists the PROXI error messages you may encounter. It explains why you got the error, and often how to correct it.
Appendix C	Presents sample PROXI programs and shows how we created them.
Appendix D	Provides a general introduction to the PROXI system for the program operator. It explains the PROXI concepts that the operator needs to know.
Appendix E	Lists the screen format parameters used by each of the screen copy files.
Appendix F	Explains how to change card format copy files to CRT format copy files.

How to Use This Manual

Read the first chapter to get an overview of the entire PROXI system. If you want more specific information about a PROXI module or its functions, you can use the tabs to locate the information you need.

As you begin to use the PROXI system, the manual will lead you step by step through the series of screen formats you'll encounter. We've tried to provide you with all the information you need where you need it; you won't have to flip back and forth.

Each screen description includes:

- The screen image format
- An explanation of what the screen format does
- Detailed information about the prompts (for the less-experienced user)
- Quick reference tables (for the PROXI pro)
- An explanation of what you do next

Related Manuals

You may also want to refer to one or more of the following manuals:

COBOL Reference Manual (AOS) (093-000223)
Introduction to the Advanced Operating System (069-000016)
Learning to Use Your Advanced Operating System (069-000018)
AOS Command Line Interpreter User's Manual (093-000122)
AOS Console User's Handbook (093-000150)
INFOS® System User's Manual (AOS) (093-000152)

What Do You Think?

At the end of this manual you'll find a Remarks Form. This is your direct line to us at Software Documentation - please use it. We want to know what you like and dislike about the manual. We welcome your suggestions, and *we really listen!* Only when the manual does its job can it help you do yours. So, help us help you.

Reader, Please Note:

We use these conventions for command formats in this manual:

COMMAND *required* [*optional*] ...

Where	Means
--------------	--------------

COMMAND	You must enter the command (or its accepted abbreviation) as shown.
----------------	---

required	You must enter some argument (such as a filename). Sometimes, we use:
-----------------	---

{ *required*₁ }
{ *required*₂ }

which means you must enter *one* of the arguments. Don't enter the braces; they only set off the choice.

[<i>optional</i>]	You have the option of entering this argument. Don't enter the brackets; they only set off what's optional.
---------------------	---

...	You may repeat the preceding entry or entries. The explanation will tell you exactly what you may repeat.
-----	---

Additionally, we use certain symbols in special ways:

Symbol	Means
---------------	--------------

)	Press the NEW LINE or carriage return (CR) key on your terminal's keyboard.
---	---

□	Be sure to put a space here. (We use this only when we must; normally, you can see where to put spaces.)
---	--

All numbers are decimal unless we indicate otherwise; e.g., 35₈.

Finally, in examples we use

THIS TYPEFACE TO SHOW YOUR ENTRY!
THIS TYPEFACE FOR SYSTEM QUERIES AND RESPONSES.

) is the CLI prompt.

Contacting Data General

If you:

- Have comments on this manual -- Please use the prepaid Remarks Form that appears after the Index.
- Require additional manuals -- Please contact your local Data General sales representative.
- Experience software problems -- Please notify your local Data General systems engineer.

End of Preface

Contents

Chapter 1 - Introduction to the PROXI™ System

PROXI Features	1-1
PROXI Modules	1-1
The Interactive Session	1-2
Data Files	1-2
Supplementary Code	1-2
Modular Construction	1-3
The Data Dictionary	1-3
The PROXI Programs	1-3
The File Maintenance Program	1-3
The File Inquiry Program	1-4
The Report Writer Program	1-4
The Form Printing Program	1-4
The PROXI Keyboard	1-4
How the PROXI Session Works	1-6
The Data-Entry Screen	1-7
Messages	1-7
Screen Numbers	1-8
Correcting Mistakes	1-8
The Any Change Cycle	1-8
How To Build a Complete Program	1-10
Editing PROXI Files	1-11
When You Change an Existing Program	1-11
Files and File-Naming Conventions	1-11
Installing the PROXI System	1-12
How to Begin a PROXI Session	1-12
The Main Menu	1-13

Chapter 2 - The Program Generator Module

When to Generate Code	2-1
PG-1	2-2

Creating a New Program	2-3
PG-2	2-4
The File Maintenance Program: An Overview	2-7
The Maintenance Menu	2-9
Additional Operator Screens	2-9
Building a File Maintenance Program	2-9
The File Maintenance Program Parameters	2-10
Reference Files	2-10
Program Screen Formats	2-12
Using Your Own Code	2-12
Own Code Status Flag	2-12
Generating the Source Code	2-12
PG-3	2-14
PG-4	2-16
PG-5	2-18
The File Inquiry Program: An Overview	2-21
Building a File Inquiry Program	2-22
The File Inquiry Program Parameters	2-23
Reference Files	2-23
Program Screen Formats	2-24
Using Your Own Code	2-24
Own Code Status Flag	2-24
Generating the Source Code	2-24
PG-6	2-26
PG-7	2-28
PG-8	2-30
The Report Writer Program: An Overview	2-33
The Report	2-33
The Title	2-35
Legend Lines	2-35
Heading Lines	2-35
Detail Lines	2-35
Conditional Fields	2-35
Control Breaks and Total Lines	2-36
Accumulators and Computations	2-36
Automatic Totaling	2-36
Logical Testing	2-37
Report Input	2-37
Sorting	2-37
Range Selection	2-37
Building a Report Writer Program	2-38
The Interactive Session	2-40
The PROXI Function Keys	2-40
PG-9	2-42
PG-10	2-44
PG-11	2-46
PG-12	2-48
PG-13	2-50
PG-14	2-52
PG-15	2-54
PG-16	2-56
PG-17	2-58
PG-18	2-60
PG-19	2-62
PG-20	2-64
PG-21	2-66
PG-22	2-68
PG-23	2-70
PG-24	2-72

The Form Printing Program: An Overview	2-75
The Form	2-75
Conditional Printing	2-76
Print Fields	2-76
Conditional Fields	2-76
Top-of-Form Lines	2-77
Detail Lines	2-77
Page Break Lines	2-77
Control Break and Total Lines	2-77
Accumulators and Computations	2-78
Automatic Totaling	2-78
Page Numbering	2-78
Logical Testing	2-79
Form Input	2-79
Range Selection	2-79
Building a Form Printing Program	2-79
The Interactive Session	2-82
The PROXI Function Keys	2-82
PG-25	2-84
PG-26	2-86
PG-27	2-88
PG-28	2-90
PG-29	2-92
PG-30	2-94
PG-31	2-96
PG-32	2-98
PG-33	2-100
PG-34	2-102
PG-35	2-104
PG-36	2-106
PG-37	2-108
PG-38	2-110
PG-39	2-112
PG-40	2-114
PG-41	2-116
PG-42	2-118
PG-43	2-120
PG-44	2-122
Changing an Existing Program	2-125
The Sequence of Screens	2-125
Adding and Deleting	2-126
The SKIP IT Key	2-126
Reminders	2-126
PG-46	2-128

Chapter 3 - The Screen Generator Module

Using the Screen Generator	3-1
The Screen Format	3-3
Variable Fields	3-3
Numbered Prompts	3-4
Record Key Fields	3-4
Manipulating the Screen - Blanking and Overlays	3-5
Defining Fields	3-6
Validity Tests	3-8
Using Your Own Code	3-8
Generating Copy Files	3-9

The Printing Functions	3-9
Changing a Screen Format	3-9
The PROXI Function Keys	3-9
SG-1	3-10
SG-2	3-14
SG-3	3-16
SG-4	3-20
SG-5	3-22
SG-6	3-24
SG-7	3-28
SG-8	3-30
SG-9	3-32
SG-10	3-36
Modifying the Screen Format Image	3-38
The BLANK SCREEN Question	3-38
Changing the Screen Field Descriptions	3-38
Changing the Validation Tests	3-40
The Own Code Parameters	3-40
Screen Copy Files	3-40
SG-11	3-42
SG-12	3-44
SG-13	3-50
SG-14	3-54
SG-15	3-58
SG-16	3-60

Chapter 4 - The File Definitions Module

The File Definitions Copy Files	4-2
FD-1	4-4
FD-2	4-8
FD-3	4-10
FD-4	4-14
FD-5	4-16
FD-6	4-18
FD-7	4-22
FD-8	4-24
FD-9	4-28
FD-10	4-32
FD-11	4-34

Chapter 5 - Running and Documenting a PROXI Program

Compiling the Code	5-1
Other Hints	5-1
Binding the Program	5-1
Documenting the Program	5-2

Appendix A - PROXI Terminology

Appendix B - PROXI Error Messages

Error Messages	B-1
Error Codes	B-2

Appendix C - Sample Programs Built Using the PROXI System

The File Maintenance Program	C-1
Main Menu	C-2
Screen FD-1	C-2
Screen FD-2	C-2
Screen FD-3	C-2
Screen FD-1	C-2
Screen FD-4	C-2
Screen FD-5	C-2
Screen FD-6	C-2
Screen FD-1	C-2
Screen FD-7	C-2
Screen FD-8	C-3
Screen FD-1	C-3
Screen FD-9	C-3
Screen FD-1	C-3
Main Menu	C-3
Screen SG-1	C-3
Screen SG-2	C-3
Screen SG-3	C-3
Screen SG-4	C-4
Screen SG-6	C-4
Screen SG-1	C-5
Screen SG-13	C-5
Screen SG-1	C-5
Screen SG-14	C-5
Screen SG-1	C-6
Main Menu	C-6
Screen PG-1	C-6
Screen PG-2	C-6
Screen PG-3	C-6
Screen PG-5	C-6
Screen PG-1	C-6
Main Menu	C-6
The Report Writer Program	C-6
Main Menu	C-7
Screen PG-1	C-7
Screen PG-2	C-7
Screen PG-9	C-7
Screen PG-11	C-8
Screen PG-12	C-8
Screen PG-13	C-8
Screen PG-14	C-8
Screen PG-15	C-8
Screen PG-17	C-8
Screen PG-18	C-9
Screen PG-19	C-9
Screen PG-17	C-9
Screen PG-19	C-9
Screen PG-17	C-10
Screen PG-21	C-10
Screen PG-22	C-10
Screen PG-23	C-10
Screen PG-22	C-10
Screen PG-21	C-10
Screen PG-24	C-10
Screen PG-1	C-10
Main Menu	C-10

Appendix D - To the Program Operator

The Keyboard	D-1
The Screen	D-2
Making Entries	D-2
Default Values	D-2
Field Characteristics	D-2
Controlling the Cursor	D-3
The Types of PROXI Program	D-3
The Printing Program	D-3
Key Range Selection	D-3
Output Method	D-4
The Interactive Program	D-4
The Any Change Cycle	D-5
The NEXT RECORD Key	D-6
Operator Error Messages	D-6

Appendix E - Screen Format Parameters

Appendix F - Using CRT Format PROXI Code

Tables

Table	Caption	
1-1	PROXI Function Keys	1-5
1-2	Control and Editing Characters	1-5
1-3	PROXI Files	1-12
2-1	Computation Symbols	2-62
2-2	Computation Symbols	2-70
2-3	Computation Symbols	2-94
2-4	Computation Symbols	2-104
2-5	Computation Symbols	2-112
2-6	Computation Symbols	2-120
3-1	Screen Editing Keys	3-18
3-2	Cursor Control Key Combinations	3-19
4-1	Clause Codes	4-20
4-2	Clause Codes	4-26
D-1	Control and Editing Characters	D-3
E-1	Screen Format Parameters	E-2

Illustrations

Figure Caption

1-1	The PROXI Terminal Templates	1-4
1-2	The PROXI Menus	1-6
1-3	Responding to a Prompt	1-7
1-4	How We Represent a Screen Format.	1-8
1-5	The Any Change Cycle	1-9
2-1	The Four PROXI Program Types.	2-6
2-2	Flow Chart of the File Maintenance Program	2-8
2-3	Typical File Maintenance Program Menu	2-9
2-4	Building a File Maintenance Program	2-11
2-5	Flow Chart of the File Inquiry Program	2-21
2-6	Building a File Inquiry Program	2-22
2-7	A Sample PROXI Report.	2-34
2-8	Building a Report Writer Program	2-38
2-9	A Sample PROXI Form	2-75
2-10	Building a Form Printing Program	2-80
3-1	The Screen Generator Module Data-Entry Screens	3-2
3-2	Literal and Variable Fields	3-3
3-3	The Three Types of Variable Fields	3-3
3-4	Screen Format Using Numbered Prompts	3-4
3-5	Screen Format Showing RECORD KEY Field	3-5
3-6	Original Screen Format	3-5
3-7	Original Screen with Overlay Screen	3-5
3-8	Information About the Current Field	3-6
3-9	Initial Step in Modifying a Field's Definition	3-38
3-10	Field Type and Entry Sequence Number	3-39
3-11	Final Step in Changing a Field's Definition	3-39
3-12	Listing of Sample Screen Format Parameters.	3-46
4-1	The File Definition Module Data-Entry Screens.	4-6
4-2	Declaratives Code (Card Format)	4-30
C-1	Record Layout	C-1
C-2	Screen Layout	C-3
C-3	Report Layout	C-7
C-4	Sample Report Produced by the SHOWLIST Program	C-11
C-5	Compilation Listing of the File Maintenance Program	C-12
C-6	Compilation Listing of the Report Writer Program	C-35
D-1	The PROXI Terminal Templates	D-1
D-2	A Typical File Maintenance Program Menu.	D-4
D-3	A Sample Screen Format	D-5
D-4	The Any Change Cycle	D-5

Chapter 1

Introduction to the PROXI™ System

The PROXI™ COBOL Code Generator is an interactive tool that helps you build business-related application programs under AOS. The PROXI™ system is easy to use. It leads you through a series of menus and data-entry screens in which you define specific aspects of the PROXI™ program you are building. After you perform all the necessary operations, you'll have a set of COBOL source files: a main program file, a number of copy files, and, optionally, your own supplementary source code files. You can compile and bind these files into an executable COBOL application program. There are four types of PROXI™ programs that you can build:

- The File Maintenance Program
- The File Inquiry Program
- The Report Writer Program
- The Form Printing Program

The PROXI system provides a main program structure that uses COPY statements to incorporate various modules of code, which perform specific functions. These PROXI copy files may also serve as source code to other PROXI programs.

PROXI Features

This section describes some of the features that make the PROXI system easy to use.

PROXI Modules

The PROXI system comprises three modules:

- the Program Generator module
- the Screen Generator module
- the File Definitions module

You use the Program Generator to build the main framework of an application program.

The Screen Generator helps you build and incorporate screen formats into a program framework.

The File Definitions module helps you build a set of copy files that contain specific information about the datafiles used by the PROXI program.

The Interactive Session

As we've mentioned, you perform all PROXI operations through an interactive session at a terminal. Starting with the main menu, you select specific operations (such as defining data files, creating data-entry screen formats, or building a program framework). Each operation presents a series of data-entry screens which allow you to provide detailed information about your program and the data it works with. The information you supply may become part of a parameter file used to build a PROXI program, or it may translate directly into coded entries for a program, depending on the operation you chose.

The PROXI system presents data-entry screens appropriate to your menu selections and to various other entries you make. These screens request specific information. They often provide you with messages listing the range of acceptable answers or the default response.

You use the terminal screen to build screen formats and print formats. The File Maintenance and File Inquiry programs, for example, employ data-entry screens to prompt the operator for input. You design these data-entry screens at the terminal, placing display and entry fields exactly as you want them to appear for the program operator.

Data Files

Each PROXI program uses a *principal data file* and up to nine additional *reference* files. The principal data file is the main source of information for a PROXI program. Reference files contain related information which the program uses to process the principal data file's records. Reference files *must* use Indexed Sequential Access Method (ISAM), (i.e., have indexed or relative organization). Using reference files gives you a great degree of flexibility and may eliminate the need for duplication of information in your database.

Depending on its type, a PROXI program can use files with sequential, indexed, or relative organization. The PROXI system does *not* support the alternate key facility or the Data Base Access Method (DBAM).

Supplementary Code

Although you can generate a complete PROXI program without adding a single statement of code, you may find it useful to expand the standard program to include special processing. The PROXI Own Code facility allows you to add procedures to a program.

File Maintenance and File Inquiry programs permit you to add Working Storage entries and a coded procedure to perform special processing. The programs use COPY statements to incorporate your code into the program framework. To execute the Procedure Division code, you must identify the code to the appropriate screen format.

You may assign a COBOL section or paragraph to be performed before or after the program processes a data-entry field in an operator screen format. This feature, for example, allows you to "translate" coded record fields into a form meaningful to the operator. You may also supply Working Storage items if they are necessary.

In some instances you *must* supply additional code. This happens only when you choose a special PROXI option such as the use of reference files or the use of accumulators. If the program will use reference files, you must provide code to perform the Open and Read procedures provided by the PROXI program. Report Writer and Form Printing programs that use any of the 99 PROXI accumulators will require you to write code that initializes and resets the accumulators.

Modular Construction

The parameter and source code copy files generated by the PROXI system may be used by other PROXI programs. A PROXI file describing a particular data file, for example, may be used by another PROXI program that uses the same data file. The PROXI system employs a set of file-naming conventions which we'll describe in detail later. These conventions permit easy referencing of specific copy files.

The Data Dictionary

As you construct your program, the PROXI system creates a Data Dictionary which contains a list of all declared data names and their pictures. The SELECT statement and FD statement you provide for each data file are the source of information for the Data Dictionary. The Data Dictionary allows the PROXI system to supply default pictures for certain data items when none exists. These data items are

- 1) The name of the data file. If the SELECT statement copy file (datafile.SL) exists, the Program Generator uses the name that appears after the SELECT keyword. If the file does not exist, the filename defaults to *datafile*.
- 2) The file status item. If datafile.SL exists, the Program Generator uses the name that appears in the FILE STATUS IS clause. If the file does not exist, the default name *datafile-STATUS* is used.
- 3) The record key item. If datafile.SL exists, the Program Generator uses the name that appears in the RECORD KEY IS clause. If the file does not exist, the default name *datafile-KEY* is used.
- 4) The record name. If the File Descriptor entries file (datafile.FD) exists, the Program Generator uses the name following the first level 01 entry for the file. If the file does not exist, the default name *datafile-RECORD* is used.

If the Program Generator uses default names, you may run into problems if you create the copy files afterwards and use data names that differ from the default names. For this reason we urge you to build all other parts of the program before calling the Program Generator. Otherwise you may find a number of undeclared data names when you attempt to compile the program.

The PROXI Programs

The PROXI system gives you four program frameworks. We'll describe each program type briefly here. Later, in the appropriate sections, we present detailed information about each of the program types.

The File Maintenance Program

The File Maintenance Program performs four functions on a principal data file. It presents the program operator with a menu listing the file maintenance functions:

- Add a new record to the data file.
- Modify (or simply display) an existing data-file record.
- Delete a data-file record.
- Call a printing program to output the data file's contents.

The operator selects a function. Then, for any function except the printing function, the operator enters one or more fields that identify the data-file record. Up to four programmer-designed screen formats handle all dialog following the file maintenance menu. After performing the requested function, the program allows the operator to process another record in the same manner. The operator may choose to cancel the function and return to the file maintenance menu to select another function.

The File Inquiry Program

The File Inquiry Program is similar to the File Maintenance Program except that it performs only one function: it displays a data-file record. There is no menu for this program. As with the File Maintenance Program, the program operator enters information that identifies the data-file record. The program then uses one or more screen formats to display the contents of the data file record.

The Report Writer Program

The Report Writer Program uses a principal data file and up to nine additional data files to produce a programmer-designed report. You specify the overall format of the report (page width, lines per page, etc.), the report title, legend and heading lines, detail and total line formats, control break fields, and line spacing.

The Form Printing Program

The Form Printing Program, like the Report Writer Program, uses a principal data file and up to nine additional data files to generate one or more forms. You may design the program to fill out preprinted forms, or to output an entire form. You define the overall layout of the form, the top-of-form lines, detail lines, page break lines, total lines, a control break field, and line spacing.

The PROXI Keyboard

In addition to the standard, labelled keys on your keyboard, the PROXI system uses special function keys. This documentation set includes a keyboard template, which you place over the row of function keys. There are two different templates - one for the D2 keyboard (model 6053) and the other for the D200 (models 6108 and 6109). You'll need only one of these templates for your terminal keyboard. The template labels the PROXI special function keys. When you want to use a particular function, simply press the appropriate key.

Figure 1-1 illustrates the PROXI terminal templates.

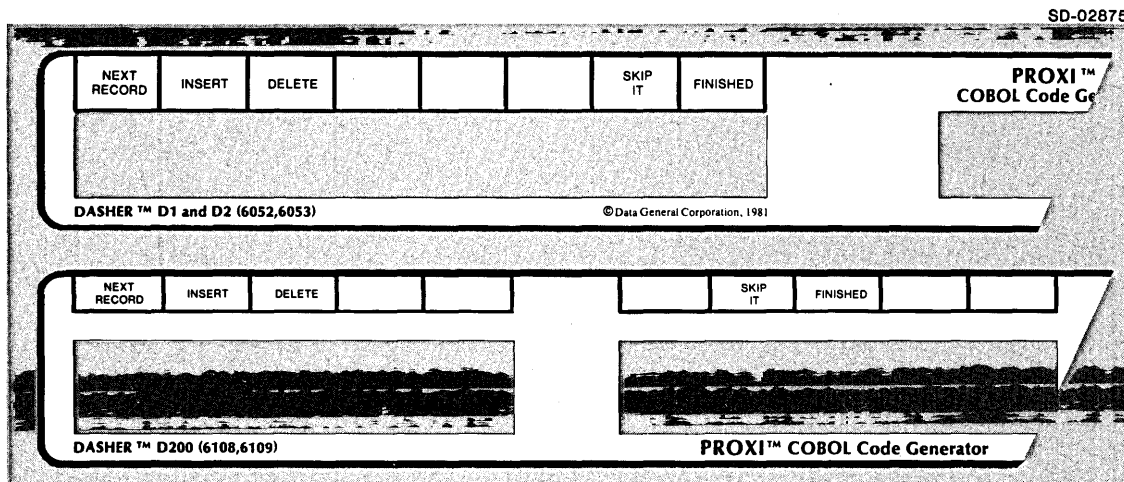


Figure 1-1. The PROXI Terminal Templates

The template will not only serve you as you perform the PROXI program generating operations; it will also help the program operator who must perform file maintenance or file inquiry operations. Be sure that the operator has a template.

The keys labelled INSERT, DELETE, SKIP IT, and FINISHED will be useful when you are building a program. The other key, NEXT RECORD, is for operator use only.

NOTE: When we refer to the DELETE key in this manual, we mean the PROXI function key, *not* the DEL key.

Table 1-1. PROXI Function Keys

Key	Action
NEXT RECORD	Operator use only. This key may be used in place of entering a RECORD KEY value. The program will then read the next sequential record in the datafile.
INSERT	Allows you to insert a serial entry into a series when changing an existing program.
DELETE	Allows you to delete a serial entry from a series when changing an existing program.
SKIP IT	Allows you to confirm remaining screens and immediately move to the COBOL code generation screen. This key is effective only when you are modifying an existing Report Writer or Form Printing program.
FINISHED	Completes your screen format. Cancels current screen and returns you to the superior level. Signals the end of serial entries.

In addition to the special PROXI function keys, you may use certain keys and key combinations to perform control and editing functions. Table 1-2 lists the control and editing keys you can use with the PROXI system.

Table 1-2. Control and Editing Characters

Key	Action
NEW LINE	Moves the cursor to the beginning of the next line.
CR	Erases the characters to the right of the cursor and moves the cursor to the beginning of the next line.
TAB	(Same as NEW LINE.)
ERASE PAGE	(Same as NEW LINE.)
→	Moves the cursor right one column.
←	Moves the cursor left one column.
↑	Moves the cursor to the beginning of the current input field.

(continues)

Table 1-2. Control and Editing Characters

Key	Action
ESC	Clears the screen to its former state. Newly created screens will be blanked. Modified screens will appear as they did after the last confirmation.
CTRL A	Moves the cursor to the end of the current field.
CTRL B	Moves the cursor to the end of the previous word in the current field.
CTRL E	Allows you to insert one character at the cursor position, moving all subsequent characters one place to the right. Note that you do not have to "close" the insertion with another CTRL E. Press CTRL E again only if you want to insert another character.
CTRL F	Moves the cursor forward to the beginning of the next word in the current field.
CTRL H	Moves the cursor to the beginning of the current field.
CTRL I	Enters the current field; it is as though you pressed NEW LINE or CR.
CTRL L	Enters the current field; it is as you though you pressed NEW LINE or CR.
CTRL U	Erases the current entry, and positions the cursor at the beginning of the field.
CTRL X	Moves the cursor right one character position.
CTRL Y	Moves the cursor left one character position.

(concluded)

How the PROXI Session Works

When you begin a PROXI session, the screen displays the main menu. You then choose to run the Program Generator, the Screen Generator, or the File Definer. Each of these modules presents another menu which lists the functions the module performs. Figure 1-2 illustrates the initial steps in a PROXI session.

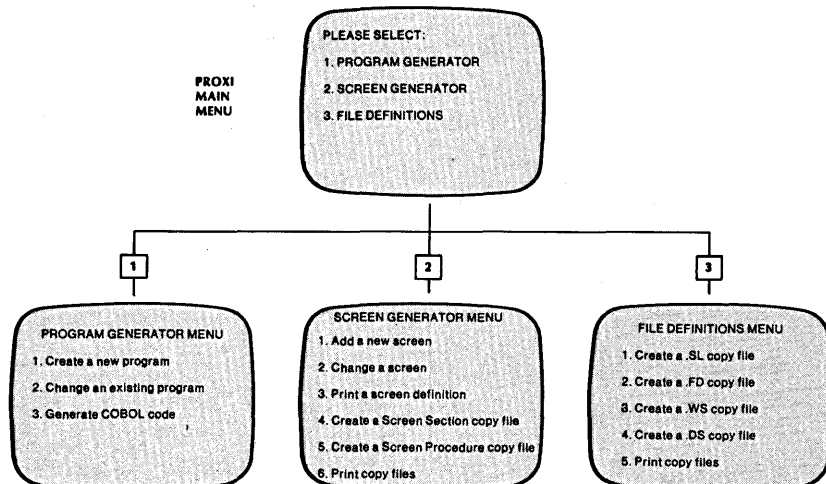


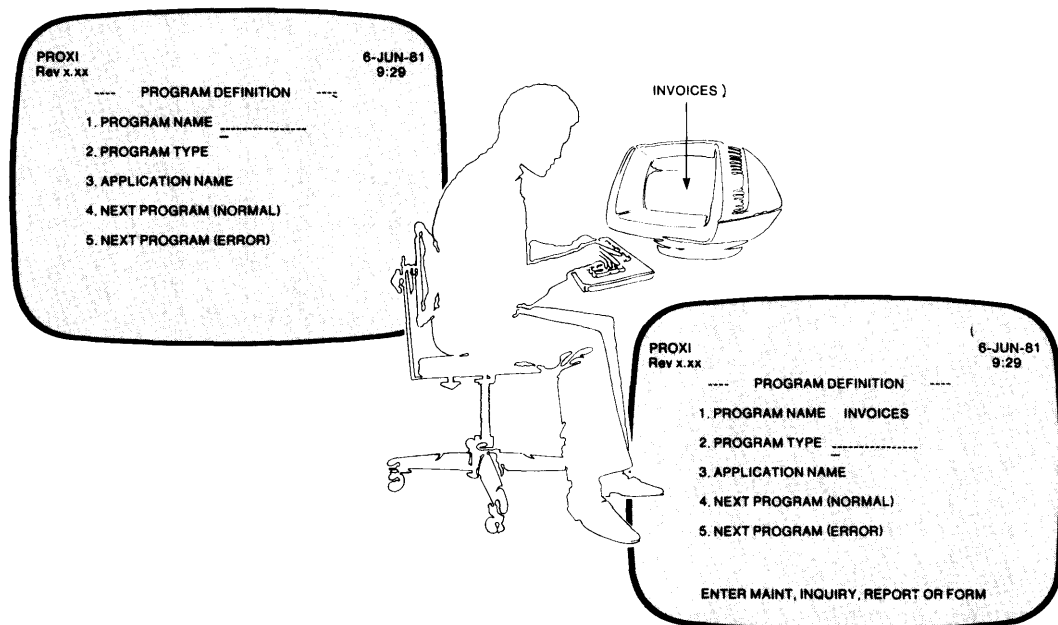
Figure 1-2. The PROXI Menus

Select a specific function from the PROXI module menu. Then you begin a corresponding series of data-entry screens. These screens prompt you for specific information that is necessary to build part of your PROXI program.

The Data-Entry Screen

The PROXI system includes many different types of data-entry screens. What we'll attempt to show here are the most important features of a PROXI screen. Once you are familiar with how these screens work, you should have no trouble using them.

In most cases, the data-entry screen lists a number of prompts or questions to which you must supply a value, a name, or a "Yes or No" response. When the system is awaiting an entry, the cursor will position itself at the beginning of the appropriate field. Enter your response; the cursor will then move to the next field (if any), or you will receive the next data-entry screen. Figure 1-3 shows what might happen as you respond to a prompt.



SD-02877

Figure 1-3. Responding to a Prompt

For certain fields the screen will display a default value. You can select the default value simply by pressing NEW LINE. The screen may also include one or more fields that display information you've already entered.

Messages

The bottom line of the PROXI screen is used for special messages. When you receive certain prompts, a message will appear at the bottom of the screen listing valid entries, the acceptable range of responses, default values, or other helpful information. If you make an invalid entry, for example, an error message appears to tell you what you did wrong and how to continue.

Screen Numbers

We've set up a numbering system to help you identify a particular data-entry screen. Please note that these numbers are used only within this manual; they do *not* appear in any of the screens you will see. We number the screens within each PROXI module:

- Program Generator: Screens PG-1 through PG-46.
- Screen Generator: Screens SG-1 through SG-16.
- File Definer: Screens FD-1 through FD-11.

The screen numbers appear in this manual above the screen illustrations. You may use these numbers to quickly locate a particular screen format.

Some fields within a screen format contain variable information. We represent a numeric variable field as 'n' or 'nnnnn'. Alphanumeric fields will appear as 'x' or 'xxxxxx'. In addition, we show your entry fields in color to distinguish them from fields displayed by the PROXI program. See Figure 1-4.

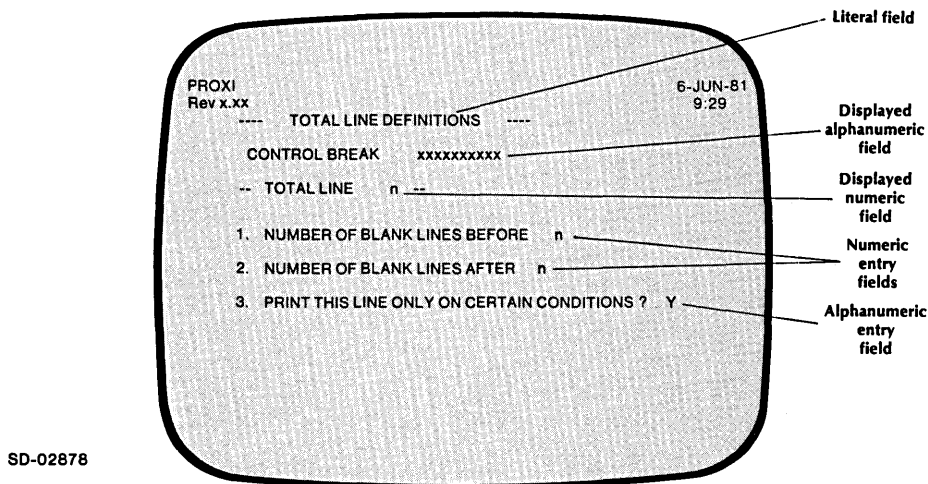


Figure 1-4. How We Represent a Screen Format

Correcting Mistakes

If, after you have entered a response to a prompt, you find that you've made a mistake and want to change your answer, you have two choices:

- Press ESC.
- Use the Any Change cycle.

When you press ESC, all entries you've made in the current screen will revert to the way they were at the start. That is, if you're filling in a new screen, all entries will disappear and you may begin over again. If you are making changes to an existing screen, the changes you've entered will be replaced by the original responses.

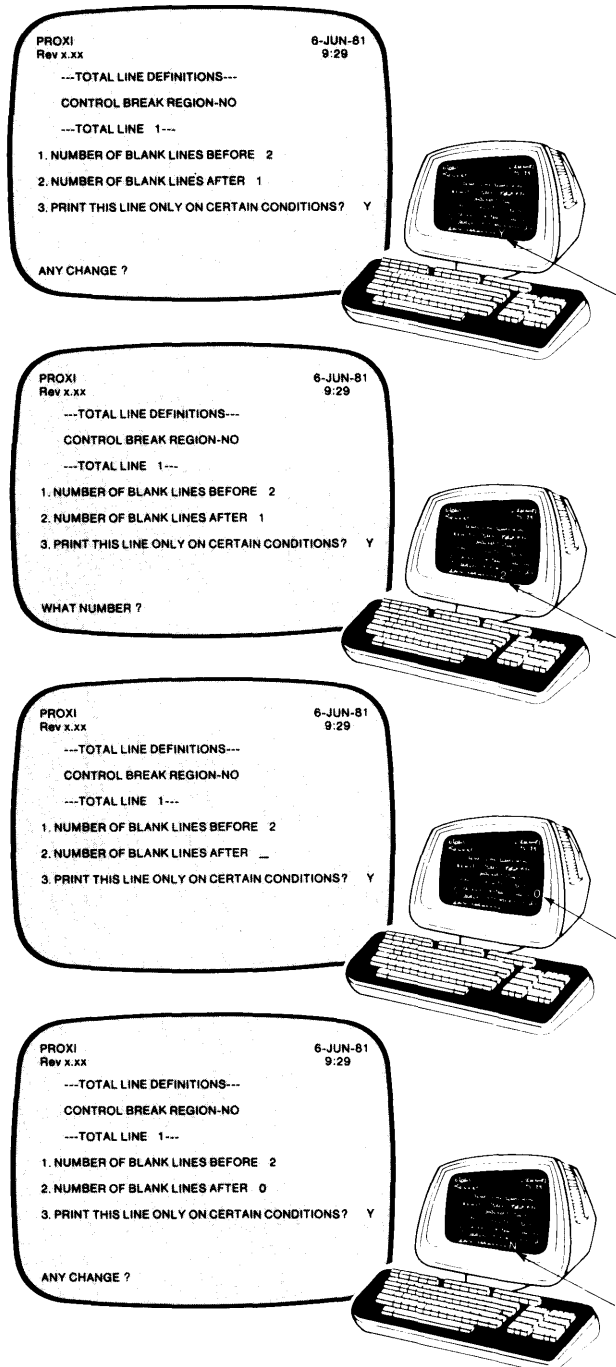
The Any Change Cycle

The Any Change cycle is a PROXI feature that asks you to confirm your entries before passing from one screen to the next. The question

ANY CHANGE ?

appears at the bottom of the screen. (See Figure 1-5.) If you are satisfied with your answers, enter N (or NEWLINE only); you'll move on to the next screen.

If you want to make a change, however, enter Y. The question *WHAT NUMBER ?* then appears. Enter the prompt number of the entry you want to change. The cursor will move to the indicated field, allowing you to make a new response. Afterwards, the Any Change question appears again. (Remember to use uppercase Y or N when responding.)



SD-02879

Figure 1-5. The Any Change Cycle

Note that certain screens may be so brief that they do not use prompt numbers. In this case you must re-enter all values if you respond Y to the Any Change question.

You may also enter the appropriate prompt number in response to the Any Change question, bypassing the What Number question. For example,

ANY CHANGE ? 2)

How To Build a Complete Program

The PROXI system has three modules:

- the File Definer
- the Screen Generator
- the Program Generator

To build a set of files that you can compile as a complete program, you do the following:

- 1) Use the File Definer to create the following files for *each* data file the program will use:

- SELECT statement copy file (datafile.SL)
- File Descriptor copy file (datafile.FD)
- Working Storage entry copy file (datafile.WS)
- Declarations Section copy file (datafile.DS)

- 2) Use the Screen Generator to create screen formats which File Maintenance and File Inquiry programs use to interact with the program operator. The File Maintenance Program already provides a menu of operations; you supply additional screens to communicate with the operator. The File Inquiry Program requires at least one data-entry screen to request RECORD KEY information from the operator. You may use up to four screen formats for each of these programs.

You create one screen format at a time through the Screen Generator. Call this module once for each screen you create.

In addition to creating the screen format you must use the Screen Generator to build two copy files for each screen: a Screen Section copy file and a Screen Procedure copy file. Be sure to perform these steps (Screen Generator menu items 4 and 5) for each screen.

- 3) Call the Program Generator to build the main structure of your PROXI program. This framework incorporates the data-file copy files and the screen format copy files into a complete unit. Use menu items 1 or 2 to produce the program parameter file. After you have completed all other steps (defining data files and building any screen formats) you can generate COBOL code (either by selecting menu item 3, or by generating the code as the last step in items 1 or 2).

After generating COBOL code you may proceed to compile it, debug it if necessary, and run the PROXI program.

NOTE: You may include segments of your own code within a PROXI program. For example, you may want to provide special processing before or after the operator enters a particular field value. If you use reference files in addition to the principal data file, you must provide code to perform the Open and Read procedures. If you use accumulators, you must supply code to initialize and reset them. Before generating the COBOL code, be sure you've provided the additional code using the section or paragraph names you specified.

The PROXI system can produce code in either card format (with line numbers) or CRT format (without numbers). The PROXI software package includes ready-made copy files that a program may require. All these copy files are in card format. If you generate a program using CRT format, you must strip the line numbers from the PROXI files that your program uses. See Appendix F for details.

Editing PROXI Files

The PROXI system builds parameter files and copy files. If you need to make changes to a program, you should carefully consider whether it's better to use a text editor or to rebuild files through the PROXI modules. You should always use the PROXI modules to alter a parameter file. After you do this, you can rebuild the associated copy files.

You may use a text editor to modify copy files. Note, however, that copy files created through parameter files may need to be edited again should you rebuild the copy file via the parameter file; your editing changes will vanish as PROXI rebuilds the copy file.

If you change information that the Data Dictionary uses, you must rebuild the program through the Program Generator (using the "Change an Existing Program" function) to avoid inconsistencies.

When You Change an Existing Program

There are a number of factors you must consider when you begin to modify an existing program. If you are making minor changes to a report layout, for example, you probably don't have too much to worry about. However, when you make significant changes such as adding another screen format or including another reference file, you must ensure that all the components of your program remain compatible.

Files and File-Naming Conventions

Your completed PROXI program consists of a main program framework plus a variety of copy files, which the program incorporates through COPY statements. In addition, the PROXI system builds one or more parameter files, which it uses to create source code for the main program framework. (The parameter files are not necessary to run your compiled PROXI COBOL program; they provide information when you build or modify PROXI copy files.)

To help identify each component of your program, the PROXI system uses a set of naming conventions. These conventions also allow more than one PROXI program to use the same component files.

The file-naming conventions use this general format:

$$\left. \begin{array}{l} \text{datafile} \\ \text{program} \\ \text{screen} \end{array} \right\} .\text{extension}$$

where:

datafile is the name of a data file used by the program.

program is the name you assign the PROXI program.

screen is the name of a screen format used by the program.

extension is a two-letter filename extension that identifies the type of file.

Table 1-3 lists the PROXI file types.

Note that PROXI programs will use these file names in program statements. Because the filenames are used within a COBOL program and also in the AOS environment, your filenames must conform to naming conventions common to each. Therefore, you cannot use the period, the underscore character, or lowercase alphabetic characters (allowed in AOS, but not COBOL), or the hyphen (allowed in COBOL, but not AOS). We recommend that you use only numeric and uppercase alphabetic characters for filenames, beginning each filename with an alphabetic character.

To refer to existing files that include the underscore or period characters in the filename, use link names.

Table 1-3. PROXI Files

Name Format	Created by	Description
program.CO	Program Generator	Program source code in card format.
program.PL	programmer*	Copy file containing Procedure Division statements.
program.PP	Program Generator	Parameter file describing the program.
program.WS	programmer*	Copy file containing Working Storage entries.
datafile.DS	File Definer	Declaratives Section copy file.
datafile.FD	File Definer	File Descriptor entries copy file.
datafile.SL	File Definer	SELECT statement copy file.
datafile.WS	File Definer	Working Storage entries copy file.
screen.PL	Screen Generator	Screen Procedure copy file.
screen.SD	Screen Generator	Screen Section copy file.
screen.SF	Screen Generator	Screen format parameter file.

* These copy files contain programmer-supplied code. The main program framework uses this naming convention in COPY statements. You do not have to follow any naming conventions for any other additional code you supply; the PROXI session will request the names of any such copy files and then build appropriate COPY statements.

Installing the PROXI System

For a description of how to install the PROXI system, refer to the Release Notice for this product. The Release Notice also lists other software you may need.

Included among the files you receive is a program that places your system installation name within the source code and listings produced by the PROXI system. To enter your installation name, enter the CLI command:

```
XEQ PROXI$91)
```

This program places the name you specify into the file FIRST\$NAME. The installation name appears in the Identification Division of PROXI source code, at the top of various listings that you may produce through the PROXI system, and at the top of most PROXI data-entry screens.

How to Begin a PROXI Session

To begin a PROXI session, type

```
PROXI)
```

The PROXI system program files must reside in the working directory or be accessible through your search list. The PROXI system uses Data General's INFOS® II System, so be sure that it is up and running.

The Main Menu

Select one of the PROXI modules.

Starting the session

```
PROXI                               6-JUN-81
Rev x.xx                             9:29

      'PROXI' - MAIN MENU

PLEASE SELECT

      1. PROGRAM GENERATOR
      2. SCREEN GENERATOR
      3. FILE DEFINITIONS
```

When the PROXI system begins executing, you will see the main menu displayed on your terminal screen. This is the origin of all your PROXI activity. The menu lists the three components of the PROXI system: the Program Generator, the Screen Generator, and the File Definer. Each of these modules performs a specific set of functions. The chapters which follow present detailed information about each module and its functions.

Here is a quick summary of each PROXI module:

1. PROGRAM GENERATOR

Select this item to build the main framework for your program - either starting from scratch, or by modifying an existing PROXI program. With this module you create the compilable code for the PROXI program. (See Chapter 2.)

2. SCREEN GENERATOR

Select this item to build screen formats for File Maintenance and File Inquiry programs. Use this module to create the screen formats and to build the necessary COBOL code that will allow your program to manage the screens. (See Chapter 3.)

3. FILE DEFINITIONS

Select this item to define each data file your program will use. This module allows you to build sets of copy files describing each data file. This set of files includes a SELECT statement, a File Descriptor (FD) entry, Working Storage items, and a Declaratives Section for the data file. (See Chapter 4.)

What Next?

After you make your selection, the PROXI system will call up the chosen module. This module will in turn present you with a menu listing the module's functions. For complete information about the PROXI modules, their menus, and their data-entry screens, turn to the appropriate tabbed section later in this manual.

End of Chapter

Chapter 2

The Program Generator Module

This chapter describes the PROXI Program Generator module. Use the Program Generator to build the main framework of an application program. You perform two steps to produce source code for a PROXI program:

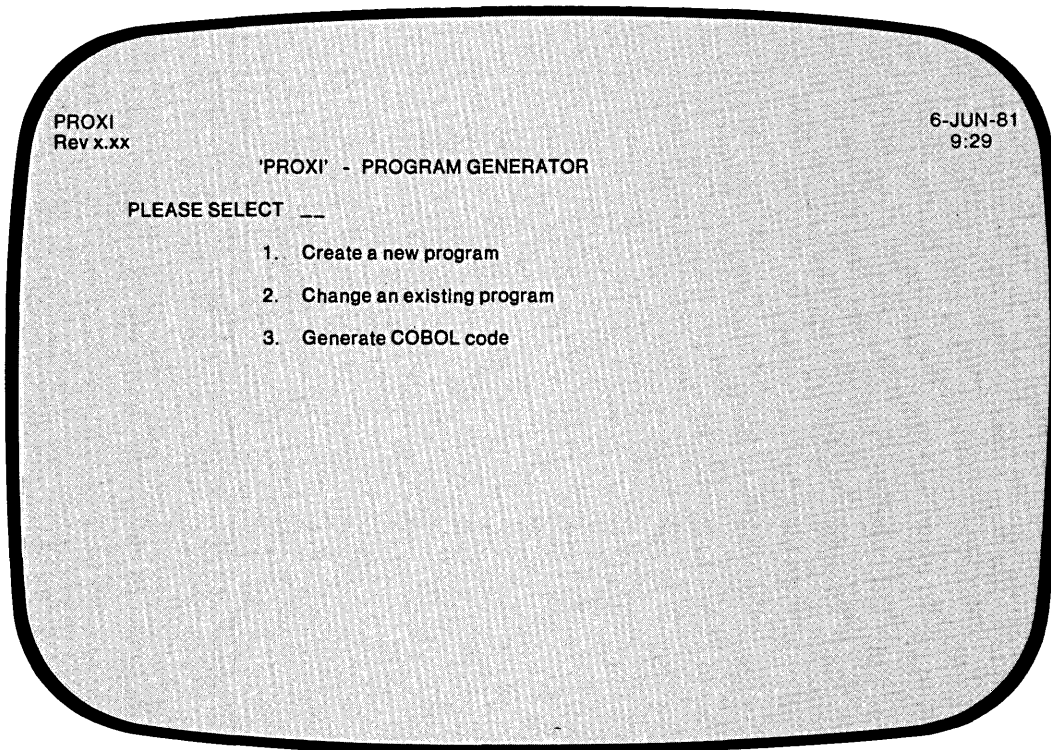
- 1) Build a program parameter file.
- 2) Request the Program Generator to generate COBOL code using the parameter file.

To build a parameter file, the Program Generator will lead you through a series of data-entry screens in which you provide specific information about the program you want to create (or modify). Your responses give the Program Generator the information it needs to construct a file it calls *program.PP*, where *program* is the name of the PROXI program you are creating.

When you request the Program Generator to produce source code from this parameter file, the Generator builds your program framework and calls it *program.CO*. This file includes all the COPY statements that will incorporate the various copy files you create using the other PROXI modules (the Screen Generator and the File Definer).

When to Generate Code

Generating COBOL code should be the last step you take to produce a complete PROXI program. The COBOL source code that the Program Generator produces contains a number of COPY statements to include necessary copy files into the program framework. You should make sure that you execute the required functions provided by the Screen Generator and File Definitions modules before generating the source code program framework that will use the copy files they create. You should also have prepared any "Own Code" files that the program will use to perform special processing.



Here is a summary of the Program Generator's functions:

1. CREATE A NEW PROGRAM

Select this item if you want to build a new parameter file for a new program. Name the program you want to create, define the type of program it will be, and then provide specific information about the program. As a final step, the Program Generator allows you to build the source code framework from the parameter file.

2. CHANGE AN EXISTING PROGRAM

Choose this item if you want to make corrections to the existing parameter file. Or choose it if you want to build a new program parameter file that is based largely on an existing one. As a final step, the Program Generator allows you to build the source code framework from the parameter file.

3. GENERATE COBOL CODE

Select this item if you want to generate a COBOL source code program framework from an existing parameter file.

What Next?

Turn to the tabbed section that matches your menu selection.

Creating a New Program

This function of the Program Generator allows you to build a new COBOL application program. The Program Generator provides the main structure for four types of business application programs:

- File Maintenance
- File Inquiry
- Report Writer
- Form Printer

The Program Generator prompts you to enter a general program definition (PG-2). After you supply the requested information, the Program Generator looks at the copy files for the SELECT statement and the FD statement. You should have created these copy files through the File Definitions module. After looking at the copy files, the Program Generator builds the Data Dictionary. If the copy files do not exist, the Program Generator supplies default data names. See Chapter 1 for details.

If the Program Generator uses default names, you may run into problems if you create the copy files afterwards and use data names that differ from the default names. For this reason, we urge you to build all other parts of the program before calling the Program Generator. Otherwise you may find a number of undeclared data names when you attempt to compile the program.

PG-2
Creating a Program

Program Generator

PROXI
Rev x.xx

6-JUN-81
9:29

---- PROGRAM DEFINITION ----

1. PROGRAM NAME
2. PROGRAM TYPE
3. APPLICATION NAME
4. NEXT PROGRAM (NORMAL)
5. NEXT PROGRAM (ERROR)

Quick Reference

Prompt	Default	Range
PROGRAM NAME	None	An AOS filename root
PROGRAM TYPE	None	MAINT, INQUIRY, REPORT, or FORM
APPLICATION NAME	None	A string containing up to 30 characters
NEXT PROGRAM (NORMAL)	LOGON	A COBOL program name
NEXT PROGRAM (ERROR)	LOGON	A COBOL program name

You selected item #1 from the Program Generator menu, "Create a New Program." The Program Generator is ready to gather information about the program you are about to build.

The Prompts

1. PROGRAM NAME

Enter the AOS filename root (up to eight characters) that you will use for this program. Your PROXI program will consist of a set of files, each of which carry the same root plus a specific two-character extension. The current operation creates a parameter file called *program.PP*. If a file already exists with this name, you'll see the message *PROGRAM ALREADY EXISTS - <CR> TO CONTINUE*. Supply a new filename or press FINISHED to return to the Program Generator menu.

2. PROGRAM TYPE

Enter one of the following to specify the type of program you want to create: MAINT (File Maintenance), INQUIRY (File Inquiry), REPORT (Report Writer), or FORM (Form Printer). A prompt at the bottom of the screen will list these options. Be sure to enter the program type exactly as it appears; use uppercase characters only.

3. APPLICATION NAME

Enter a character string (up to 30 characters) that will identify this program or the applications system it belongs to. The PROXI program will display this string on various screens it presents to the program operator.

4. NEXT PROGRAM (NORMAL)

Enter the name of the COBOL program to which control will pass upon normal termination of your PROXI program. The default response is LOGON. Normal termination for a File Maintenance program occurs after the operator presses FINISHED at the menu. A File Inquiry program terminates normally after the operator presses FINISHED at the first prompt. Report Writer and Form Printing programs terminate normally after all output is either printed or spooled.

5. NEXT PROGRAM (ERROR)

When your program attempts to open a file, an error condition may sometimes occur. Enter the name of the COBOL program to which control will pass if this should happen. The default response is LOGON.

NOTE: If your program attempts to pass control to LOGON on a normal or error termination, but LOGON.PR does not exist, the program returns you to the CLI environment.

What Next?

The next data-entry screen you see depends on the program type you specified: File Maintenance, File Inquiry, Report Writer, or Form Printing. Turn to the appropriate section for descriptions of the data-entry screens and the prompts you will receive.

Figure 2-1 illustrates how your response to the *PROGRAM TYPE* prompt determines the sequence of screens the Program Generator presents.

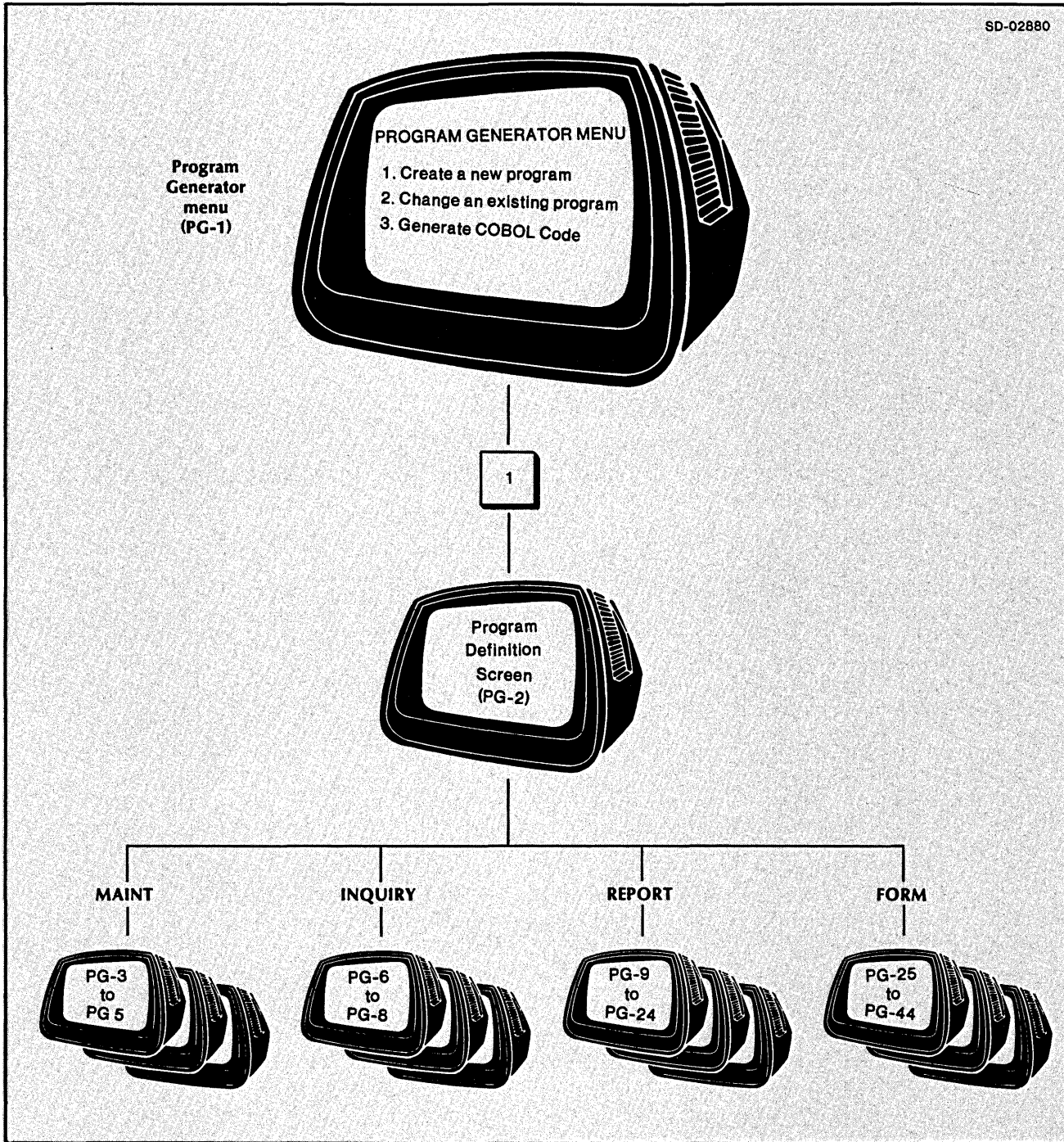


Figure 2-1. The Four PROXI Program Types

The File Maintenance Program: An Overview

The PROXI File Maintenance Program allows the operator to perform four operations on the principal data file:

- Add a record
- Change or display a record
- Delete a record
- Print the file

The principal data file must be an ISAM file (having indexed or relative organization).

After the operator makes a selection, PROXI prompts the operator to identify the data file record to be processed. Only one entry in the data-entry screen corresponds to the RECORD KEY for the data file. The PROXI program performs the chosen function on the indicated record. Then, optionally, it displays up to three additional data-entry screens before it loops to request information for another record.

The operator uses the PROXI function keys to manipulate the data entries and record display, and to signal a completed transaction or operation. The PROXI terminal template labels the function keys.

Figure 2-2 is a flow chart that illustrates the operation of the File Maintenance Program.

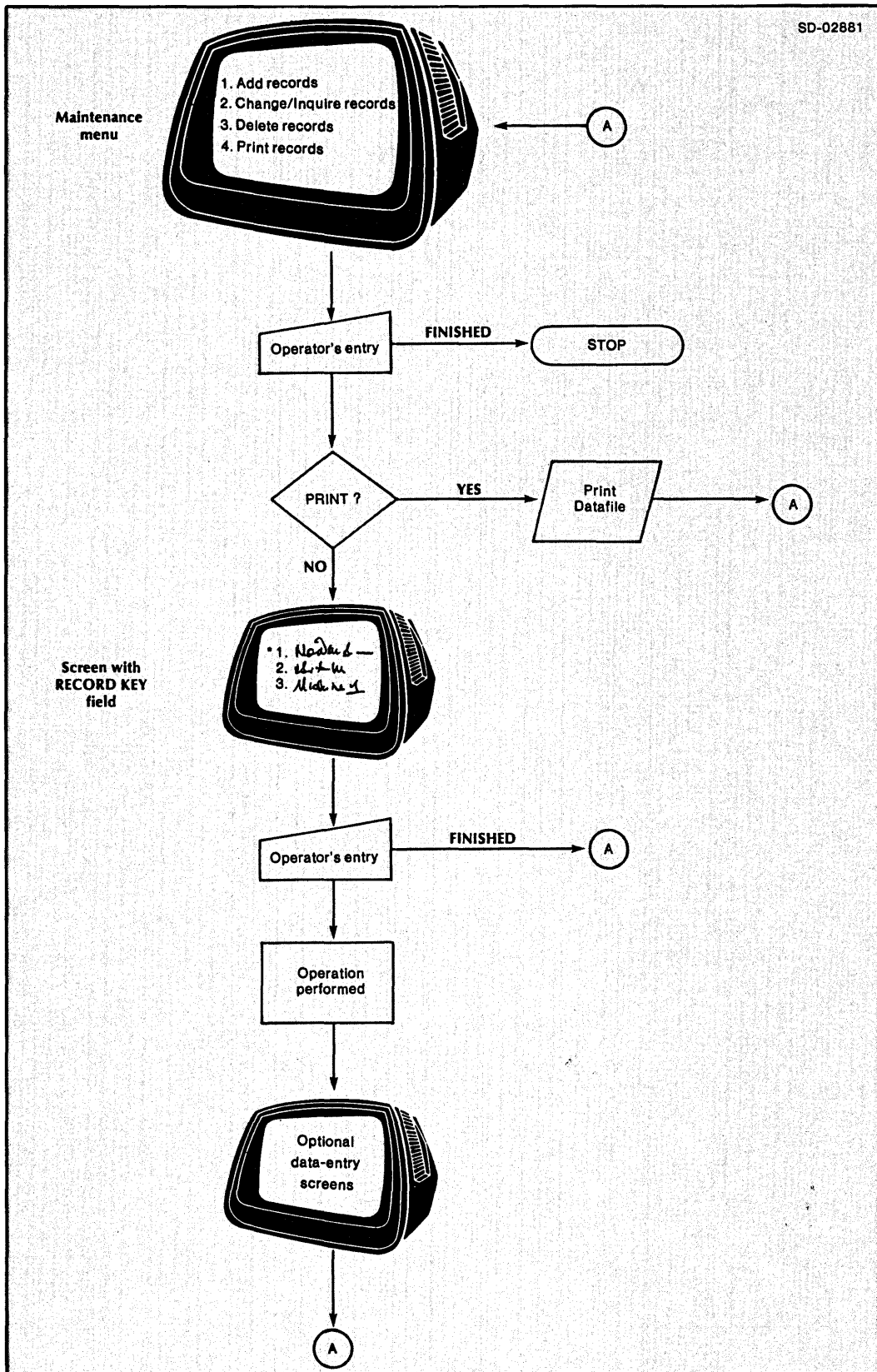


Figure 2-2. Flow Chart of the File Maintenance Program

The Maintenance Menu

The PROXI File Maintenance Program presents the operator with a menu listing four operations. When building the program, you can supply an application label (to identify the File Maintenance Program) and a record label (to refer to a data-file record). Figure 2-3 illustrates a typical File Maintenance Program menu and points out the programmer-supplied labels.

SD-02882

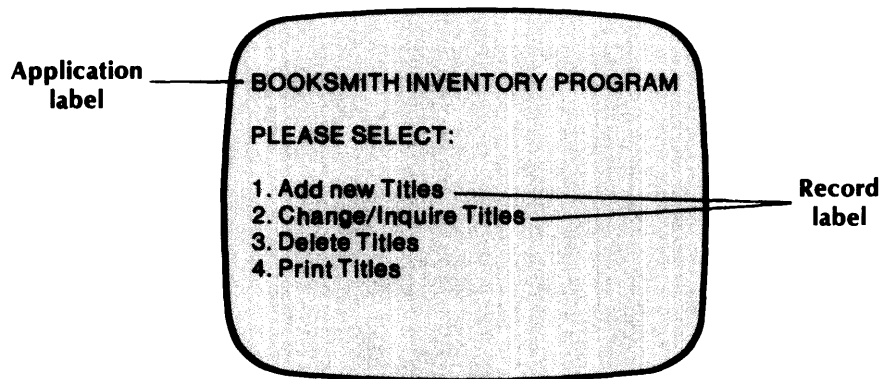


Figure 2-3. Typical File Maintenance Program Menu

Additional Operator Screens

In addition to the main menu, you may provide the operator with other screen formats to process specific information for a record. To do this you must create appropriate screen format definitions. The PROXI Screen Generator module lets you lay out a screen exactly as you want it to appear to the operator. You define all variable fields on the screen, indicating whether they are entry fields (requesting operator input), or display or format fields (providing information to the operator).

The Program Generator will ask you for the names of the screens that the File Maintenance Program will use. The main framework of the program will use COPY statements to incorporate the screen copy files into the program.

Building a File Maintenance Program

Building a File Maintenance Program is relatively easy: setting up the main program framework is a short process. Make sure that you have already provided a general program definition through the Program Definition screen (PG-2); then there are only a few remaining tasks to be done. Refer to Figure 2-4 as we describe each step.

The File Maintenance Program Parameters

You must identify the many resources that the program framework requires to complete itself. Most of the information you give helps produce a set of COPY statements that incorporate pieces of code into the File Maintenance program framework. In the data-entry screens PG-3 and PG-4 you identify

- the data file this program maintains
- the reference files
- the screen format files that interact with the operator
- any supplementary COBOL code
- the program that prints the contents of the data file

After supplying all of this information, you may generate code for the COBOL program (screen PG-5). Descriptions of screens PG-3, PG-4, and PG-5 follow at the end of this section.

In addition to building the framework of the program, you must build one or more screen formats called Screen Section and Screen Procedure copy files. You must also build a complete set of copy files that describe all the data files used by the program. These data-file copy files include a SELECT statement, a File Descriptor entry, Working Storage entries, and a Declaratives section for each data file.

Reference Files

If you intend to use reference files with this program, the Program Generator provides code to open and read these files. Note, however, that the File Maintenance program will *not* perform this code. You must supply additional "Own Code" to complete processing for the reference files. (If you intend to use reference files, be sure to respond Yes when asked if you will supply your own code.)

Remember that all reference files must be ISAM files.

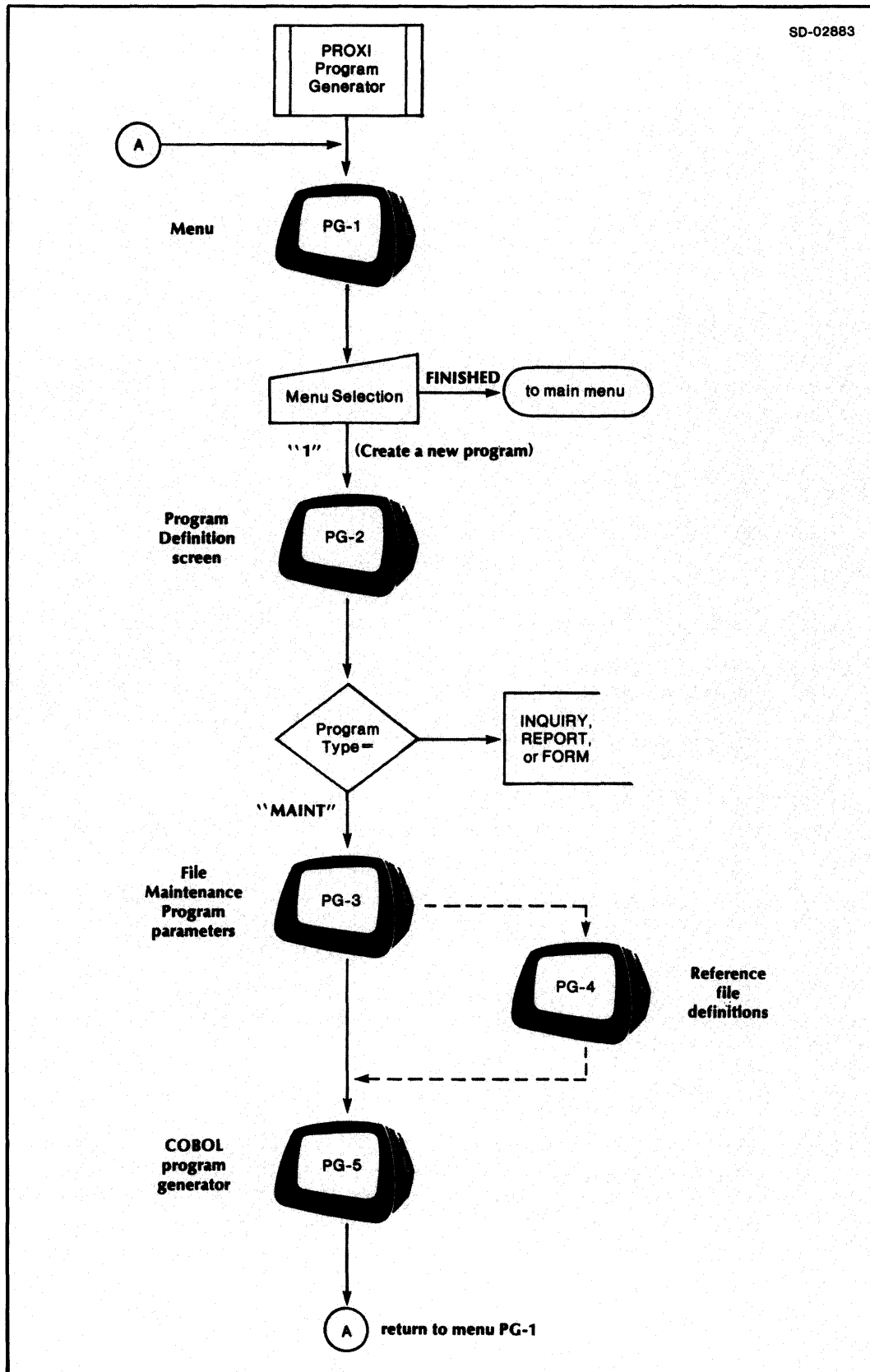


Figure 2-4. Building a File Maintenance Program

Program Screen Formats

The PROXI File Maintenance Program uses one or more screens to manage interaction with the operator. The first screen the operator sees is the maintenance menu, which we've already described. The Program Generator supplies this screen format using the labels you specify. For the File Maintenance Program to process an operator request (other than the Print function), the operator must enter a RECORD KEY field which identifies the data-file record to be processed. (You create each screen format that will appear on the terminal screen. For an explanation about how to construct a screen format, refer to the Screen Generator Overview in Chapter 3.) The screen may prompt the operator to enter data, or it may display information such as a field from a data-file record or data previously entered by the operator.

Using Your Own Code

The PROXI system allows you to give additional processing capabilities to the File Maintenance Program. To do so, you must set up copy files containing Working Storage items and Procedure Division statements. The program uses COPY statements to incorporate your code into its framework. Use the following names for these copy files:

program.WS (the Working Storage code copy file)

program.PL (the Procedure Division code copy file)

where:

`program` is the name of the File Maintenance Program that will incorporate the copy file.

The program will execute a section or paragraph contained in `program.PL` *only* when you associate the section or paragraph name with one or more fields in a screen format. Each screen format allows you to name your own code. The program will execute your code before or after processing an operator's entry to a field.

Own Code Status Flag

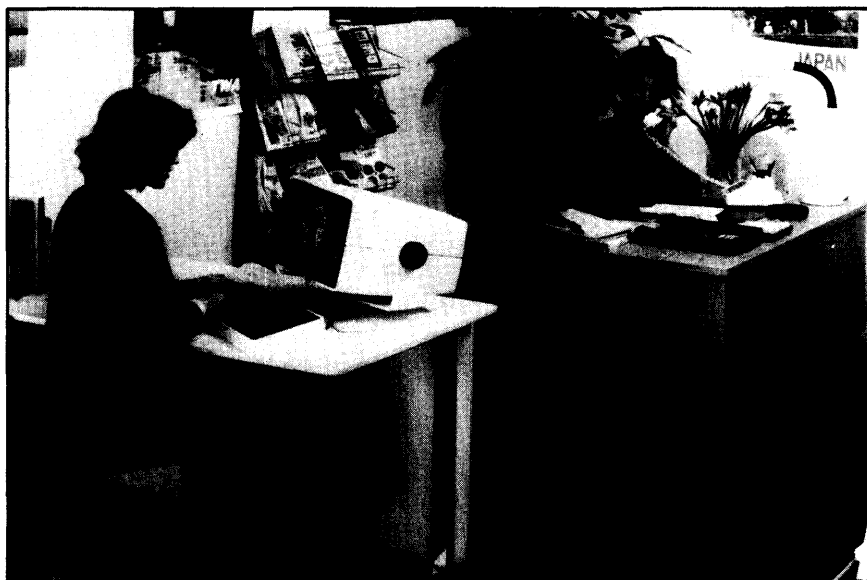
The Working Storage Section contains a data item called OC-STATUS which your code can use to affect processing of a screen field. This status flag carries the picture PIC 9.

If the Own Code executes *before* a screen field and sets OC-STATUS to 0, the File Maintenance program will process the field. If your code sets it to any other value, the program will skip processing for this field.

If the Own Code executes *after* a screen field and sets OC-STATUS to 0, the program will move on to process the next field. If your code sets it to any other value, the program will process the current field again.

Generating the Source Code

Once you define the File Maintenance Program, you may immediately generate the COBOL code for the program. You should perform this step only after you have completed all other steps in building the File Maintenance program, (such as generating screen formats and defining all data files).



PG-3 Creating a Maintenance Program

Program Generator

```

PROXI                               6-JUN-81
Rev x.xx                             9:29
---- MAINTENANCE PROGRAM PARAMETERS ----

1. FILE NAME
2. MENU ITEM NAME
3. SCREEN NAME 1                      7. OWN CODE - .WS ?
4. SCREEN NAME 2                      8. OWN CODE - .PL ?
5. SCREEN NAME 3                      9. PRINT PROGRAM
6. SCREEN NAME 4                     10. REFERENCE FILES ?
    
```

Quick Reference

Prompt	Default	Range
FILE NAME	None	An AOS filename
MENU ITEM NAME	"RECORDS"	A string containing up to 20 characters
SCREEN NAME	None	A screen format name
OWN CODE - .WS	N	Y or N
OWN CODE - .PL	N	Y or N
PRINT PROGRAM	N	A program name
REFERENCE FILES	N	Y or N

You indicated in screen PG-2 that you want to create a File Maintenance program. The Program Generator is ready to gather additional, specific information that it needs to build the program parameter file, *program.PP*. You have identified *program* in the Program Definition screen (PG-2).

After you identify all data files, the Program Generator builds a Data Dictionary which lists all declared data names and their pictures. The Program Generator uses the .SL (SELECT statement) and .FD (File Descriptor entries) files to build the Data Dictionary. If these files do not yet exist, the Program Generator uses default names for certain data items.

The Prompts

1. FILE NAME

Enter the AOS filename that identifies the data file this program will maintain. The PROXI program uses COPY statements to include the SELECT statement copy file (datafile.SL), the FD statement copy file (datafile.FD), the Working Storage entries copy file (datafile.WS), and the Declaratives Section copy file (datafile.DS). See Chapter 4 for details about building these files. (The filename you specify cannot exceed 10 characters.)

2. MENU ITEM NAME

Enter a string (up to 20 characters) for the PROXI program to use to describe the records in the data file. The default name is RECORDS. This name appears in the File Maintenance program menu. You should therefore select a name that will be helpful to the program operator.

3-6. SCREEN NAME *n*

You may provide up to four screen formats for the File Maintenance program. You specify the names of these screen formats here. The Program Generator places COPY statements in your program to incorporate the files screen.SD and screen.PL. (Chapter 3 explains how you can build these files through the Screen Generator.)

7. OWN CODE - .WS ?

Enter Y if you want to include your own code in the Working Storage Section of this program. Be sure to name your code program.WS. This is the name used by the COPY statement that incorporates your code into the program. Enter N (the default response) if you do not need to use additional Working Storage code in the program.

8. OWN CODE - .PL ?

Enter Y if you want to include your own Procedure Division code in the program. Be sure to name this code program.PL. This is the name used by the COPY statement that incorporates your code into the program. To execute this code you must associate it with a screen format. Enter N (the default response) if you do not need to use additional Procedure Division code in the program.

9. PRINT PROGRAM

Enter the name of a COBOL program to which control will pass when the operator selects the PRINT option from the File Maintenance menu. If you do not specify a print program, the File Maintenance program will ignore the operator's selection of that option. You may use this entry to have a PROXI Report Writer or Form Printing program generate output from the updated data file. The program name you specify cannot exceed eight characters.

10. REFERENCE FILES ?

Enter Y if you will supply additional data files for the program. You may use up to nine reference files in addition to the principal data file specified above. The default response is N. (If you use reference files, the Program Generator prompts you for the names of the file(s) next. See screen PG-4. The filenames cannot exceed 10 characters.

NOTE: If you use reference files, you must provide the program with additional code to perform the PROXI code that opens and reads these files. Be sure to respond Y (Yes) to the Own Code questions. (If you've already entered N to the questions, you may change your responses during the Any Change cycle.)

What Next?

If your program uses reference files, continue with the next screen description (PG-4).

If the program does not require reference files, you may move on to the COBOL Code Generation screen (PG-5).

After you identify all the data files the program uses, the Program Generator builds the Data Dictionary. It uses information you supplied in the .FD and .SL copy files for each data file. If you have not created these files, the Program Generator supplies default values for certain data items. See Chapter 1 for details.

PG-4
Creating a Maintenance Program

Program Generator

```

PROXI                               6-JUN-81
Rev x.xx                             9:29

----- MAINTENANCE PROGRAM PARAMETERS -----

1. FILE NAME             xxxxxxxx
2. MENU ITEM NAME       xxxxxx
3. SCREEN NAME 1        xxxxxx
4. SCREEN NAME 2        xxxxxx
5. SCREEN NAME 3
6. SCREEN NAME 4
7. OWN CODE - .WS ?    x
8. OWN CODE - .PL ?    x
9. PRINT PROGRAM      xxxxxxxx
10. REFERENCE FILES ?  Y

..... REFERENCE FILE DEFINITIONS .....

-- REFERENCE FILE      n --
      REFERENCE FILE NAME
  
```

Quick Reference

Prompt	Default	Range
REFERENCE FILE	None	An AOS filename

This overlay screen appears if you indicate that the File Maintenance Program uses reference files. You may specify up to nine reference files in addition to the principal data file. The Program Generator continues to prompt for a filename until you enter a ninth filename or press FINISHED.

REFERENCE FILE NAME

Enter the AOS filename that identifies the reference file. Press FINISHED at this prompt after you name the last reference file. You do not need to enter the reference files in any particular order. A reference filename cannot exceed 10 characters.

The Any Change cycle occurs after you enter each filename.

What Next?

After you specify the last reference file and confirm your entry through the Any Change cycle, press FINISHED at the first prompt (unless you have specified nine reference files). The Program Generator will then allow you to generate COBOL source code based on the parameter file you created (PG-5).

After you identify all the data files the program uses, the Program Generator builds the Data Dictionary using information you supplied in the .FD and .SL copy files for each data file. If you have not created these files, the Program Generator supplies default values for certain data items. See Chapter 1 for details.

PG-5
Creating a Maintenance Program

Program Generator

PROXI
Rev x.xx

6-JUN-81
9:29

---- COBOL PROGRAM GENERATION ----

1. PROGRAM NAME xxxxxxxx
2. GENERATE THE COBOL CODE ?
3. OUTPUT FORMAT

Quick Reference

Prompt	Default	Range
GENERATE COBOL CODE	Y	Y or N
OUTPUT FORMAT	1	1 or 2

The Program Generator displays this screen after you define the File Maintenance Program. The name of the program appears after the first prompt.

The Prompts

2. *GENERATE THE COBOL CODE ?*

Enter Y if you want to generate the code for your File Maintenance Program. *Note that Y is the default response.* You should generate the code only after you complete all other PROXI operations for the program (such as defining all data files and constructing screen formats). Enter N if you want to immediately return to the Program Generator menu, skipping the code generation step.

3. *OUTPUT FORMAT*

Enter 1 for card format (code with line numbers), or 2 for CRT format (code without line numbers). The default response is 1 (card format).

NOTE: The Any Change cycle does not allow you to change the program name. If you want to build a new program based on the parameters of this program, use function 2 of the Program Generator module, "Change an Existing Program". See the appropriate section for details.

What Next?

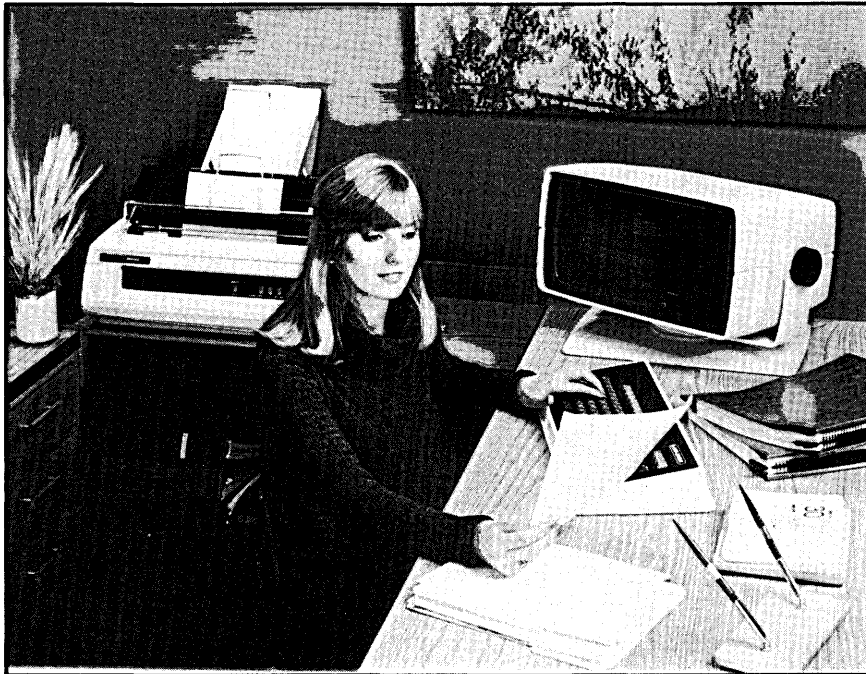
If you elect to generate COBOL code, you'll see the message

LOADING PARAMETERS FOR program - PLEASE WAIT

The Program Generator displays division and section titles on the screen as it builds each section of source code.

You return to the Program Generator menu (PG-1) whether or not you generate source code.

If you generated code, you may proceed to compile it after you end the PROXI session. Be sure that you have completed all necessary operations before attempting to compile the code. If you neglect to create the required copy files, for example, the compiler will be unable to compile the source code.



The File Inquiry Program: An Overview

The PROXI File Inquiry Program allows the operator to display record information from a principal data file and from up to nine reference files. This data file must be an ISAM file (having indexed or relative organization). The File Inquiry Program performs a subset of the File Maintenance Program functions: it does not allow the operator to add, change, or delete information from the file.

You must supply the screen formats which will guide the operator. Specifically, you must prompt the operator for the key field that you need to locate the desired record on the data file. The PROXI Screen Generator module lets you lay out a screen exactly as you want it to appear to the operator. You define all variable fields on the screen and indicate whether they are entry fields (requesting operator input), or display or format fields (providing information to the operator).

The Program Generator module asks you for the names of the screens that the File Inquiry Program will use. The main structure of the program will include COPY statements to copy the appropriate copy files into the program.

The File Inquiry Program flow chart is shown in Figure 2-5.

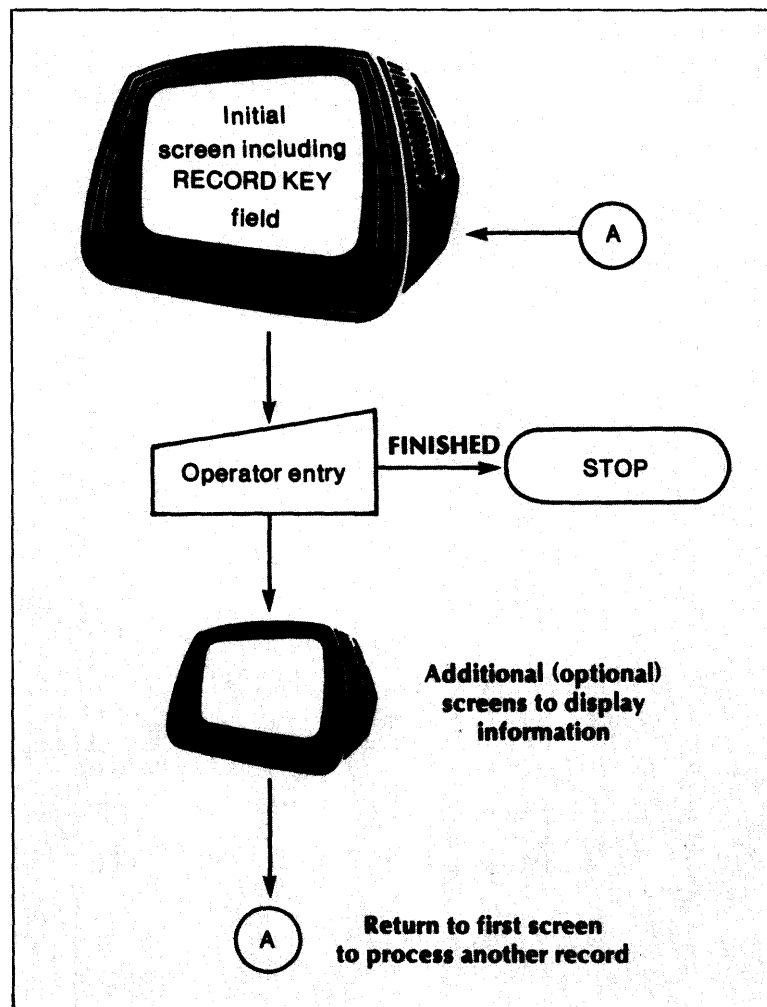


Figure 2-5. Flow Chart of the File Inquiry Program

Building a File Inquiry Program

Building a File Inquiry Program is relatively easy: it's a short process to set up the main program framework. Assuming that you have already provided a general program definition in the first data-entry screen of the Program Generator module, there are only a few remaining tasks to be done. Figure 2-6 illustrates the steps you take to build a File Inquiry Program.

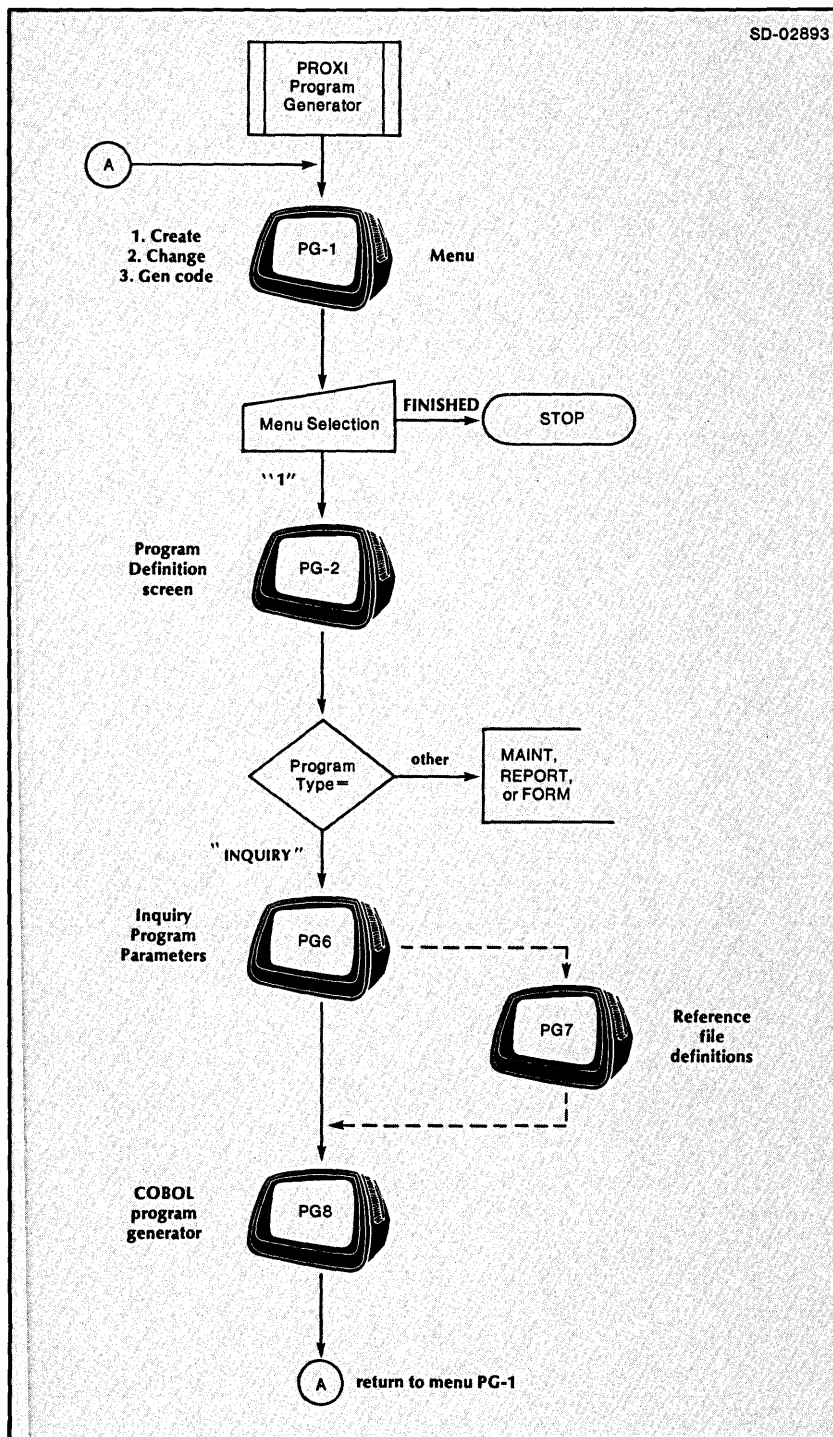


Figure 2-6. Building a File Inquiry Program

The File Inquiry Program Parameters

You must identify the resources that the program framework requires to complete itself. Most of the information that you give helps to produce a set of COPY statements. These incorporate pieces of code into the File Inquiry Program framework. In the upcoming data-entry screens you must identify:

- the data file this program examines
- any reference files
- the screen format(s) that interact with the operator
- any supplementary COBOL code

After supplying all of this information, you may generate code for the COBOL program, provided that you have taken care of all other related operations. In addition to building the framework of the program, you must build screen formats (one or more), Screen Section and Screen Procedure copy files, and a complete set of copy files that describe all the data files used by the program. These data-file copy files include a SELECT statement, a file description, a Working Storage section, and a Declaratives section for each data file. If you need supplemental code, you must provide it before you generate COBOL code for the File Inquiry Program.

Reference Files

If you intend to use reference files with this program, the Program Generator provides code to open and read these files. Note, however, that the File Inquiry Program will *not* perform this code. You must supply additional Own Code to process the reference files. If you plan to use reference files, be sure to respond Y when asked if you will supply your own code.

Remember that all reference files must be ISAM files.

Program Screen Formats

The PROXI File Inquiry Program uses one or more screens to manage interaction with the operator. You must build at least one screen format to accept a RECORD KEY field from the operator. The program uses this RECORD KEY to identify the data-file record to be examined. Screen formats may

- display information taken from a data-file record
- display information entered by the operator
- request additional information from the operator (used for display purposes only; no change to the data file occurs).

(For an explanation about how to construct a screen format, refer to the Screen Generator Overview in Chapter 3.)

Using Your Own Code

The PROXI system allows you to give additional processing capabilities to the File Inquiry Program. To do so, you must set up copy files containing Working Storage items and Procedure Division statements. The program uses COPY statements to incorporate your code into its framework. Use the following names for these copy files:

program.WS (the Working Storage code copy file)

program.PL (the Procedure Division code copy file)

where:

program is the name of the File Inquiry program that will incorporate the copy file.

The program will execute a section or paragraph contained in program.PL *only* when you associate the section or paragraph name with one or more fields in a screen format. Each screen format allows you to name your own code. The program will execute your code before or after processing an operator's entry to a field.

Own Code Status Flag

The Working Storage section contains a data item called OC-STATUS which your code can use to affect processing of a screen field. This status flag carries the picture PIC 9.

If the Own Code executes *before* a screen field and sets OC-STATUS to 0, the File Inquiry Program will process the field. If your code sets it to any other value, the program will skip processing for this field.

If the Own Code executes *after* a screen field and sets OC-STATUS to 0, the program will move on to process the next field. If your code sets it to any other value, the program will process the current field again.

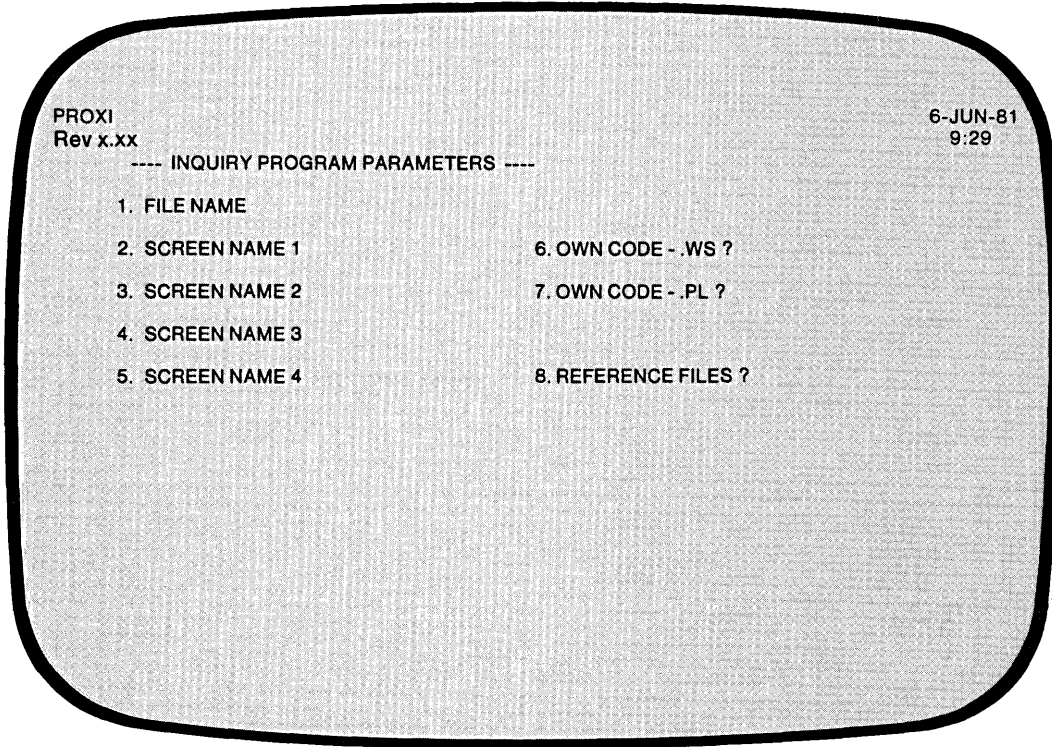
Generating the Source Code

Once you define the File Inquiry Program, the Program Generator allows you to generate the COBOL code for the program. You should perform this step only after you have completed all other steps in defining your File Inquiry Program, (such as generating screen formats and defining all data files).



PG-6
Creating a File Inquiry Program

Program Generator



Quick Reference

Prompt	Default	Range
FILE NAME	None	An AOS filename
SCREEN NAME	None	A screen format name
OWN CODE - .WS	N	Y or N
OWN CODE - .PL	N	Y or N

You indicated in screen PG-2 that you want to create a File Inquiry Program. The Program Generator uses the information you supply to build the program parameter file, program.PP.

After you identify all the data files the program uses, the Program Generator builds the Data Dictionary using information you supplied in the .FD and .SL copy files for each data file. If you have not created these files, the Program Generator supplies default values for certain data items. See Chapter 1 for details.

The Prompts

1. FILE NAME

Enter the AOS filename that identifies the data file this program will examine. The File Inquiry Program uses COPY statements to incorporate the SELECT statement copy file (datafile.SL), the FD statement copy file (datafile.FD), the Working Storage entries copy file (datafile.WS), and the Declaratives Section copy file (datafile.DS). Chapter 4 describes how you create these files using the File Definer.

2-5. SCREEN NAME *n*

The Program Generator prompts you for up to four screen format names. The File Inquiry Program uses the screen formats to interact with the operator. It uses COPY statements to incorporate the Screen Section copy file (screen.SD) and the Screen Procedure copy file (screen.PL). Chapter 3 describes how you create the screen formats and the copy files using the Screen Generator.

6. OWN CODE - .WS ?

Enter Y if you want to include your own code in the Working Storage Section of the program. Be sure to name your code "program.WS" as this is the name used in the copy file that will include your code in the program. Enter N (the default response) if you do not need to use additional Working Storage code in the program.

7. OWN CODE - .PL ?

Enter Y if you want to include your own Procedure Division code in the program. Be sure to name your code "program.PL" as this is the name used in the copy file that will include your code in the program. Enter N (the default response) if you do not need to use additional Procedure Division code in the program. To execute this code you must associate it by name with a screen format.

8. REFERENCE FILES ?

Enter Y if the program uses supplemental data files. It may use up to nine reference files in addition to the principal data file specified above. The Program Generator prompts you for the names of the reference file(s) next. See screen PG-7.

NOTE: If you use reference files, you must provide additional code to perform the PROXI code that opens and reads these files. Be sure to respond Y to the Own Code question. (If you already responded N to the question, you may change your answer during the Any Change cycle.)

Enter N if the File Inquiry Program will not need any reference files.

What Next?

If your program uses reference files, continue with the next screen description (PG-7).

If the program does not require reference files, you may move on to the COBOL Generation screen (PG-8).

PG-7
Creating a File Inquiry Program

Program Generator

PROXI
Rev x.xx

6-JUN-81
9:29

---- INQUIRY PROGRAM PARAMETERS ----

- | | | | |
|------------------|---------|----------------------|---|
| 1. FILE NAME | xxxxxxx | | |
| 2. SCREEN NAME 1 | xxxxxx | 6. OWN CODE - .WS ? | x |
| 3. SCREEN NAME 2 | xxxxxx | 7. OWN CODE - .PL ? | x |
| 4. SCREEN NAME 3 | xxxxxx | | |
| 5. SCREEN NAME 4 | xxxxxx | 8. REFERENCE FILES ? | Y |

..... REFERENCE FILE DEFINITIONS

-- REFERENCE FILE n --

REFERENCE FILE NAME

Quick Reference

Prompt	Default	Range
REFERENCE FILE	None	An AOS filename

This overlay screen appears when you indicate that the File Inquiry Program uses one or more reference files. The Program Generator continues to prompt for a name until you enter a ninth filename or press FINISHED.

After you identify all the data files the program uses, the Program Generator builds the Data Dictionary using information you supplied in the .FD and .SL copy files for each data file. If you have not created these files, the Program Generator supplies default values for certain data items. See Chapter 1 for details.

The Prompt

REFERENCE FILE NAME

Enter the AOS filename that identifies the reference file. Press FINISHED after you have named all reference files.

What Next?

After identifying all reference files, you are ready for the final screen of this series, the COBOL Program Generation screen (PG-8).

PG-8
Creating a File Inquiry Program

Program Generator

PROXI
Rev x.xx

6-JUN-81
9:29

----- COBOL PROGRAM GENERATION -----

1. PROGRAM NAME xxxxxxxx
2. GENERATE THE COBOL CODE ?
3. OUTPUT FORMAT

Quick Reference

Prompt	Default	Range
GENERATE COBOL CODE	Y	Y or N
OUTPUT FORMAT	1	1 or 2

The Program Generator displays this screen after you complete the definition of the File Inquiry Program. It supplies the program name which you gave earlier.

The Prompts

2. GENERATE THE COBOL CODE ?

Enter Y if you want to generate the code for your File Inquiry program. *Note that Y is the default response.* You should perform this operation only after you complete all other PROXI operations for this program (such as defining all data files and building screen formats). Enter N if you want to immediately return to the Program Generator menu, skipping the code generation step.

3. OUTPUT FORMAT

Enter 1 for card format (code with line numbers), or 2 for CRT format (code without line numbers).

What Next?

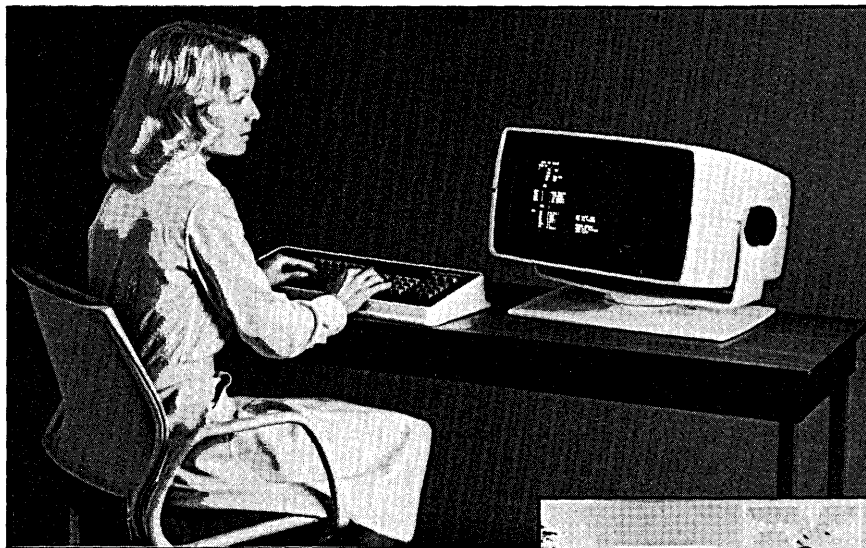
If you elect to generate COBOL code, you'll see the message

LOADING PARAMETERS FOR program - PLEASE WAIT

The Program Generator displays division and section titles on the screen as it builds each section of source code.

You return to the Program Generator menu (PG-1) whether or not you generate COBOL code.

If you generated code, you may proceed to compile it after you end the PROXI session. Be sure that you have completed all necessary operations before attempting to compile the code. If you neglect to create the required copy files, for example, the compiler will be unable to compile the source code.



The Report Writer Program: An Overview

The Report Writer Program allows you to produce printed reports using the records of a principal data file and up to nine additional reference files. You can request that the program output the reports on a printer, or spool them to a disk file for printing at a later time.

The Report

The PROXI Report does not use a standard format that you must conform to. Instead, it allows you to design your own report format within the boundaries of an 80- or 132-column page. During the interactive session with the Program Generator, you describe

- the overall layout of the report

- the title, legend, and heading lines

- the detail lines

- control breaks

- total lines

- record selection criteria

- computations on print line fields

- printing conditions

Figure 2-7 illustrates a sample report. Refer to this example as we describe the various features of the PROXI report.

Legend Lines (3)		Title		Heading Lines (3)		
SEMI-MONTHLY SALES REPORT						
AUCHTERARDEW IMPORTS OF PERTH, LTD. 51st Division Street Isle of Skye						
Region	Agent	Expected Sales	Actual Sales	% + or -		
12	2871	10,380	10,257	-1.18		
12	2904	12,100	12,547	+3.69		
12	3021	11,500	11,075	-3.70		
12	3087	13,250	14,227	+7.37		
12	3184	12,550	13,125	+4.58		
12	3279	14,325	12,186	-14.93		
12	Totals:	74,105	73,417	-0.93		
15	2831	22,875	23,548	+2.94		
15	2893	24,565	25,234	+2.72		
15	3608	18,450	21,340	+15.66		
15	4592	22,500	19,382	-13.86		
15	4688	21,750	20,805	-4.34		
15	5034	22,950	24,105	+5.03		
15	5321	19,750	19,976	+1.14		
15	Totals:	152,840	154,390	+1.01		
18	4872	13,450	13,240	-1.56		
18	5125	12,800	14,250	+11.33		
18	5318	14,350	15,075	+5.05		
18	7398	15,250	14,995	-1.67		
18	7473	16,000	15,440	-3.50		
18	7503	15,450	16,300	+5.50		
18	Totals:	87,300	89,300	+2.29		
21	3971	10,075	11,300	+12.16		
21	3984	11,650	12,565	+7.85		
21	4021	10,500	10,875	+3.57		
21	5067	12,350	14,129	+14.40		
21	6284	9,550	11,125	+16.49		
21	7419	13,185	12,956	-1.74		
21	Totals:	67,310	72,950	+8.38		
25	4920	21,535	22,647	+5.16		
25	5813	19,865	21,430	+7.88		
29		Totals:	120,490	124,720	+3.51	
Final Totals:		656,215	673,898	+2.69		

Repeated Legend Line (1)		
Expected Sales	Actual Sales	% + or -
24,250	23,340	-3.75
22,500	24,312	+8.05
20,620	22,765	+10.40
21,950	22,155	+0.93
23,450	22,472	-4.17
154,170	150,121	+3.21
18,450	18,240	-1.14
22,800	24,250	+6.36
23,920	24,375	+1.90
19,870	21,045	+5.91
17,450	19,230	+10.20
18,000	17,580	-2.33
PERTH, LTD.		
Grand Total		

SD-02886

Figure 2-7. A Sample PROXI Report

The Title

The report's title appears on the first page only. It is centered at the top of the page (on line 3). The length of the title depends on the width of the report. An 80-column report can have a title 60 characters long. A 132-column report may use a title 90 characters in length.

Legend Lines

Legend lines appear under the title on the first page (beginning on line 5). Legend lines are not centered; they begin at the left margin. You may specify up to nine such lines and optionally request that the first *n* legend lines appear on subsequent pages.

Heading Lines

Heading lines appear on every page immediately below any legend lines. You may enter up to nine heading lines on a page.

Detail Lines

A detail line is the basic information unit of a report. Each detail line may contain up to 99 print fields. The print field may display a literal or report a record field value, a computation result, or a constant.

In the simplest case, one record from the principal data file produces one detail line. Every time the program reads a record, it generates another detail line for the report.

If you want to print more information (or format it differently) you can establish multiple detail lines for each data-file record. The program will then produce two or more detail lines for each record it reads from the principal data file.

To provide even more flexibility, the Report Writer Program allows you to specify conditions for printing a detail line. You set up from one to 99 logical tests to be performed. If the result of these tests is true, the program will print the corresponding detail line; if the result is false, the line is not printed.

Suppose you need to print different groups of detail lines depending on certain information. The Report Writer Program allows you to establish separate, conditionally printed groups of one or more detail lines, each group having its own set of criteria. Now, for example, you can print a group of three detail lines under one set of conditions, and print a group of two detail lines under another set of conditions. This "Record Selection" feature also allows you to selectively process records within the principal data file. Records not passing any set of criteria are ignored.

Within a group of detail lines controlled by record selection criteria, you may further impose printing conditions for a particular detail line.

Conditional Fields

The Report Writer Program allows you to set up conditionally printed fields within a detail line. To do this you must:

- 1) Define one detail line containing *only* the conditional field(s). When you define this line you enter an asterisk (*) following the BLANK LINES AFTER prompt in screen PG-17. You must also supply the printing criteria that will apply to both this field *and* the line that will contain the field.
- 2) Define another detail line containing *only* the field(s) that you want to appear each time the line is printed. You may specify printing conditions for this line as well.

Let's look at an example. We define one detail line that represents the conditional field. It looks like this:

OVERDRAFT

The condition we specify for this field is

IF CURRENT-BALANCE < 0.00

Next we define the part of the detail line that will always appear. It includes the following fields: DATE, TRANSACTION-AMT, TRANS-CODE, and CURRENT-BALANCE.

XXXXXXXX	\$\$\$,\$\$9.99	xxx	-\$,\$\$\$,\$\$9.99
----------	-----------------	-----	---------------------

When processing the data-file records, the Report Writer Program will consider these two lines as a single detail line and print the first portion only when conditions permit. A portion of the resulting report might look like this:

04-MAY-81	\$10.00	DEP	\$44.42	
06-MAY-81	\$47.89	CHK	-\$3.47	OVERDRAFT

Control Breaks and Total Lines

You define one or more control break fields for the report. When the program detects a change in the value of a control break field, it will generate total lines associated with that control break field. You must define the control breaks beginning with the lowest level (the most specific) up through the highest level (the most general). A control break at a higher level automatically generates a control break and total lines at each lower level.

An end-of-file condition on the principal data file will cause a control break at the highest level (and all lower levels).

Total lines may include literals, data-item values, computation results, and constants.

Accumulators and Computations

Each print field has an associated data item or constant. You may perform simple calculations using the field value and a specified accumulator. There are 99 accumulators available for your use; they are named A01 through A99. If you use any of these accumulators, you must provide your own code for initializing and resetting them.

Automatic Totaling

The Report Writer Program offers an automatic field-totaling facility. You do not have to manipulate values and accumulators; the program can do it for you automatically. You may request automatic totaling for any numeric field in a detail or total line.

If you want automatic totaling for a field, simply enter the code T at the COMPUTATION prompt for that field. The program will then add the contents of the field to the auto-total accumulator reserved for that field. The program can supply up to 99 auto-total accumulators.

When a control break occurs, the program will automatically print the accumulator value using the field's column position and picture. (You may want to assign the field's picture with the total in mind.) Be sure that you provide enough fields in the total line definition to handle the automatic totals.

The program automatically sets the appropriate accumulators to zero when multiple control break fields are in effect.

Logical Testing

The Report Writer Program allows you to specify record selection criteria and printing conditions for detail lines. You may specify from one to 99 logical tests that the program will use to determine if it will print the corresponding line. If the result of the test series is false, the program will not print the line.

Each test consists of four parts arranged in the following format:

$$\left. \begin{array}{l} \text{IF} \\ \text{AND} \\ \text{OR} \end{array} \right\} \text{field-name} \left. \begin{array}{l} \text{EQ} \\ \text{NE} \\ \text{LT} \\ \text{LE} \\ \text{GT} \\ \text{GE} \end{array} \right\} \left\{ \begin{array}{l} \text{field-name} \\ \text{literal} \end{array} \right\}$$

The first test always begins with IF. You'll receive prompts for the remaining three parts of the first test.

The program will apply the following hierarchy of evaluation when you include more than one test:

1. Tests joined by AND
2. Tests joined by OR
3. Nested tests joined by IF

Report Input

As we mentioned earlier, you may use up to nine reference files in addition to the principal data file.

Sorting

If you are working with a principal data file with indexed or relative organization (ISAM), you may request the Report Writer Program to sort the file on up to ten key fields. (An ISAM file is ordered "sequentially" according to RECORD KEY values.) Sorting on a different key field allows you to generate a different set of reports based on other record data.

You cannot sort a file that has sequential organization.

Range Selection

You may allow the operator to select one or more particular ranges of records within the principal data file, to be used as input to the report. The operator specifies a start key and an ending key to identify the range of records to be processed.

Building a Report Writer Program

To build a Report Writer Program you must

- Use the PROXI File Definitions module to define all data files that your program will use.
- Create the program framework through the PROXI Program Generator module. This chapter explains how to do this.

The flow chart in Figure 2-8 diagrams the steps you take to build a Report Writer program.

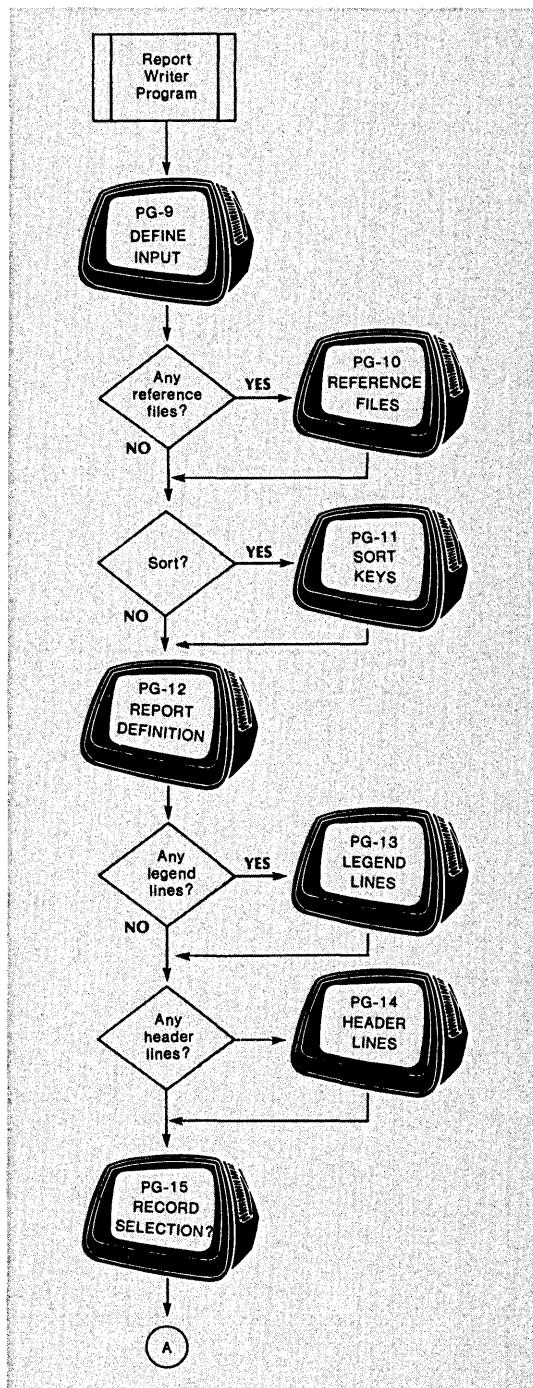


Figure 2-8. Building a Report Writer Program (continues)

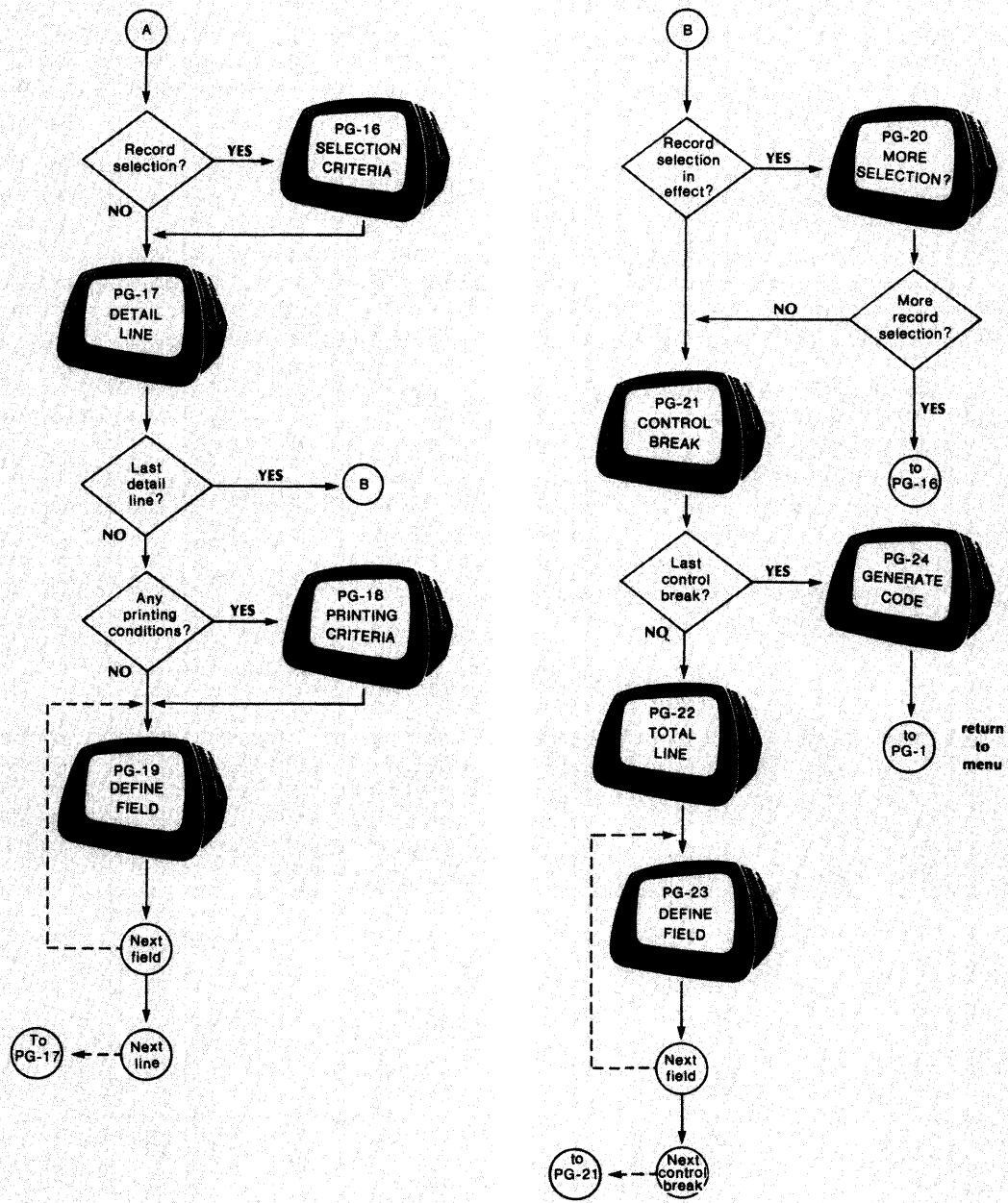


Figure 2-8. Building a Report Writer Program (concluded)

The Interactive Session

As you can see in Figure 2-8, you begin by defining the input to the Report Writer program. You name the principal data file (PG-9), and optionally, any reference files (PG-10), and provide sort keys if you want the principal data file sorted (PG-11).

After defining the input, you begin to lay out your report format (PG-12). In addition, you can allow the operator to select a particular range of records from the principal data file. You also name the disk file to which output will be spooled if the printer is not available. You specify the width of the report, supply its title, legend (PG-13), and header lines (PG-14).

Before gathering information about the detail lines, the Program Generator asks if you want to set up selection criteria for data-file records (PG-15). This feature allows you to specify up to 99 logical tests to be performed to determine if a record should be processed (PG-16). You may define more than one detail line to be associated with a set of selection criteria. If necessary, you may establish separate sets of selection criteria, each set controlling one or more detail lines (PG-20).

The Program Generator then asks about a detail line (PG-17). You specify the number of blank lines to precede and follow the detail line. You may also specify printing conditions for the line; again, you may provide up to 99 logical tests to determine if the report will include a detail line (PG-18). You then describe each field within the detail line, giving its column position, length, the data item or constant to be printed in this field, and whether you will perform any calculations on the field value (PG-19). The PROXI system provides the Column 999 field which is a nonprinting field in which you can perform calculations.

You have defined all fields for all detail lines, and all detail lines for all sets of selection criteria. Now you define the control break fields (PG-21) and their associated total lines (PG-22). As with the detail lines, describe each field within the total line (PG-23). After defining all total lines and all control breaks, you arrive at the final screen (PG-24). The Program Generator allows you to generate a COBOL code framework for the Report Writer Program. Perform this step only after completing all necessary file definitions.

Refer to the screen descriptions that follow for details about any of the steps in building a Report Writer Program.

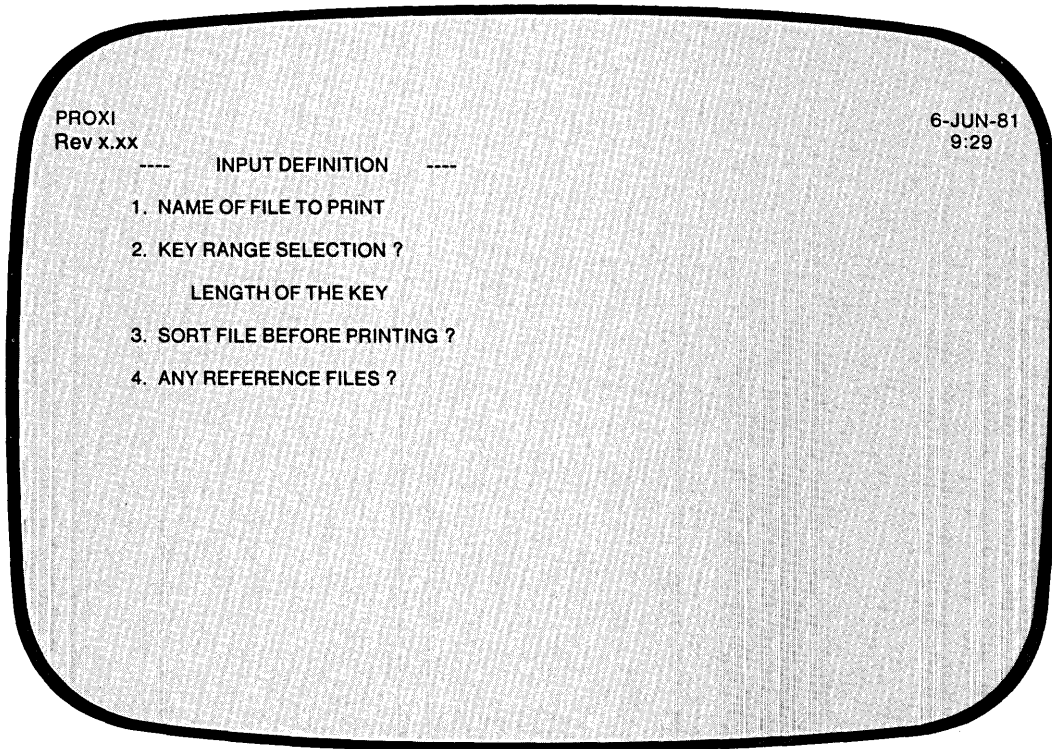
The PROXI Function Keys

In the screen descriptions that follow we often refer to the function keys FINISHED, INSERT, and DELETE. You will not use the INSERT or DELETE keys when you are building a new program; these keys provide special functions when you are modifying an existing program. We've mentioned them in this section because you may also refer to this section for reference information while you're changing a program. If you are not changing a program, but creating a new one, you may use only the FINISHED key.



PG-9
Creating a Report Writer Program

Program Generator



Quick Reference

Prompt	Default	Range
NAME OF FILE	None	An AOS filename
KEY RANGE SELECTION	N	Y or N
LENGTH OF KEY	None	1 - 99
SORT FILE	N	Y or N
ANY REFERENCE FILES	N	Y or N

You indicated in screen PG-2 that you want to build a Report Writer Program. The Program Generator is ready to gather information about the principal data file which the program will use to generate the report.

After you identify the principal data file (and any reference files) the Program Generator builds the Data Dictionary using the SELECT and FD statements for all data files.

After you identify all the data files the program uses, the Program Generator builds the Data Dictionary using information you supplied in the .FD and .SL copy files for each data file. If you have not created these files, the Program Generator supplies default values for certain data items. See Chapter 1 for details.

The Prompts

1. NAME OF FILE TO PRINT

Enter the name of the principal data file. The Report Writer Program uses COPY statements to include the files datafile.SL (the SELECT statement), datafile.FD (the File Descriptor entries), datafile.WS (the Working Storage entries), and datafile.DS (the Declaratives Section) into its structure. Chapter 4 explains how to create these files using the PROXI File Definitions module.

2. KEY RANGE SELECTION ?

Enter Y if you want to allow the operator to select one or more specific ranges of records as input to the Report Writer Program. Enter N (the default response) if you will not allow range selection. The Program Generator skips the next prompt if you respond N.

LENGTH OF KEY

Enter an integer in the range 1 through 99 to define the length of the data file's RECORD KEY. There is no default answer. The Program Generator uses this key length of build the Key Range Selection screen format for the operator. (If you want to modify your response during the Any Change cycle, enter 2 as the item to be changed).

3. SORT FILE BEFORE PRINTING ?

If you want to sort the principal data file before processing it, answer Y; the Program Generator requests sort key information shortly. If you do not require sorting, answer N (the default). The file must have indexed or relative organization; you cannot sort a sequentially organized file.

4. ANY REFERENCE FILES ?

If you want to include additional data files to process the principal data file, answer Y. Answer N (the default) if the program will not use any reference files.

What Next?

If you indicate that the program uses reference files, the Program Generator prompts you for information about each of these files (PG-10).

If you answer N to the reference file question, but Y to the sort question, the Program Generator prompts you to enter sort keys (PG-11).

If this program does not use reference files and does not require sorting, you may begin to define the overall layout of the report (PG-12).

PG-10
Creating a Report Writer Program

Program Generator

```

PROXI                                     6-JUN-81
Rev x.xx                                  9:29

----- INPUT DEFINITION -----

1. NAME OF FILE TO PRINT          xxxxxxxx
2. KEY RANGE SELECTION ?         x
   LENGTH OF THE KEY              nn
3. SORT FILE BEFORE PRINTING ?   x
4. ANY REFERENCE FILES ?        Y

***** REFERENCE FILE DEFINITIONS *****
-- REFERENCE FILE  n --
5. NAME OF REFERENCE FILE
6. FIELD USED TO ACCESS FILE
    
```

Quick Reference

Prompt	Default	Range
REFERENCE FILE NAME	None	An AOS filename, or the FINISHED, INSERT, or DELETE key
FIELD USED TO ACCESS	None	A field name or constant

This overlay screen appears only if you indicate that the Report Writer program will use reference files.

You may name up to nine reference files to be used by the Report Writer Program. The Program Generator displays the reference file number as it prompts you to enter a name. It will continue to prompt for a filename until you name a ninth reference file or press FINISHED.

The Prompts

5. NAME OF REFERENCE FILE

Enter the AOS filename that identifies the data file. There is no default response. Press FINISHED after you have entered the names of all reference files.

6. FIELD USED TO ACCESS FILE

Enter the name of the field in the principal data file that corresponds to this reference file's RECORD KEY. There is no default answer.

What Next?

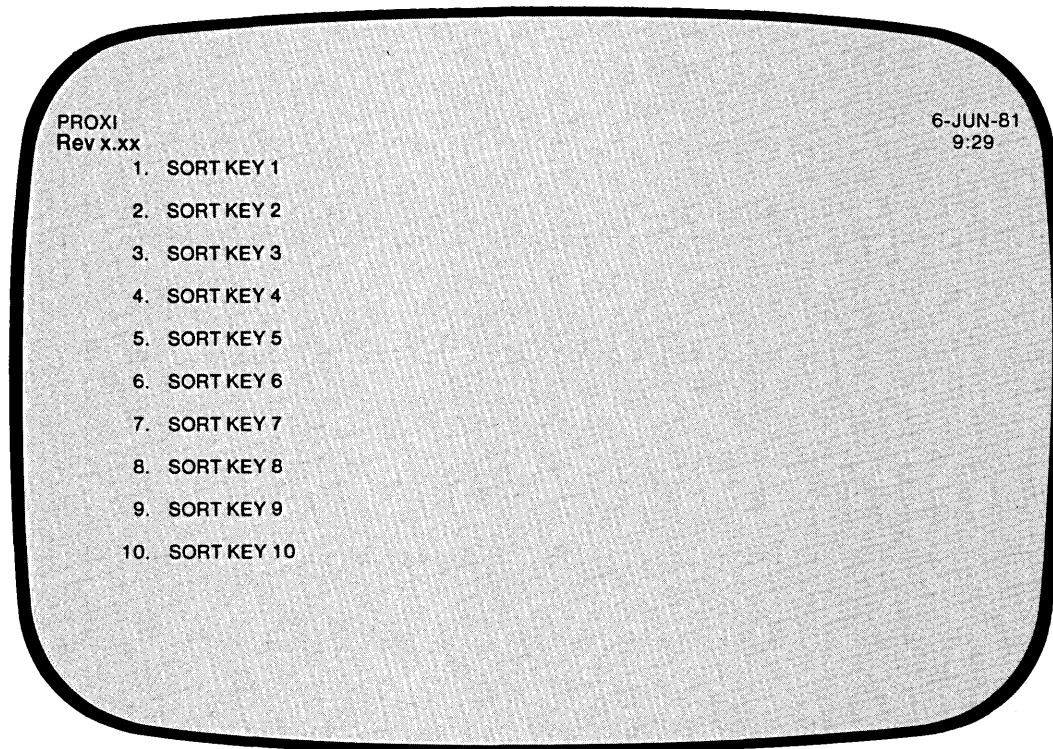
If you requested sorting for the principal data file, continue with the next screen description (PG-11).

If the principal data file does not require sorting, you begin to define the overall layout of the report (PG-12).

After you identify all the data files the program uses, the Program Generator builds the Data Dictionary using information you supplied in the .FD and .SL copy files for each data file. If you have not created these files, the Program Generator supplies default values for certain data items. See Chapter 1 for details.

PG-11
Creating a Report Writer Program

Program Generator



Quick Reference

Prompt	Default	Range
SORT KEY	None	A field name

You'll see this data-entry screen if you request sorting of the principal data file. You may enter up to ten sort keys. Begin with the highest level key and end with the lowest. Each sort key must be a field in the principal data file.

Note that the sum of the lengths of the sort keys you specify (and therefore the principal data file's RECORD KEY) *must* be less than 100 bytes. You cannot sort a file unless it has relative or indexed organization.

The Prompt

SORT KEY n

Enter the field name in the principal data file on which you want a sort performed. Remember to start with the highest (broadest) level and then move toward the lowest (most specific) level.

For example:

SORT KEY 1 DEPARTMENT-NO)

SORT KEY 2 SECTION-NO)

SORT KEY 3 PROJECT-NO)

SORT KEY 4 EMPLOYEE-NO)

You may enter up to ten sort keys. The Program Generator will continue to prompt for a key until you enter a tenth key or press FINISHED.

What Next?

The Program Generator begins a series of data-entry screens that describe the format of the report you want to print (PG-12).

PG-12
Creating a Report Writer Program

Program Generator

```

PROXI                               6-JUN-81
Rev x.xx                             9:29
----- REPORT DEFINITION -----

1. PRINT OR SPOOL
2. SPOOL FILE NAME
3. PAGE WIDTH (NO. OF COLUMNS)
4. NO. OF PRINT LINES PER PAGE
5. IS RUN TIME TO BE PRINTED ?
6. REPORT TITLE - 1 - 30
                  31 - 60
                  61 - 90
7. DO YOU HAVE LEGEND LINES ?      HOW MANY FROM PAGE 2
8. DO YOU HAVE HEADING LINES ?
  
```

Quick Reference

Prompt	Default	Range
PRINT OR SPOOL	P	P, S, or O
SPOOL FILE NAME	None	An AOS filename
PAGE WIDTH	132	132 or 80
PRINT LINES PER PAGE	60	0 - 99
PRINT RUNTIME	N	Y or N
REPORT TITLE	Blanks	A character string
ANY LEGEND LINES	N	Y or N
LINES FROM PAGE 2	0	0 - 9
ANY HEADING LINES	N	Y or N

The Program Generator is ready to gather information about the format of the report you want to produce. You define the overall layout of the report with this data-entry screen. Subsequent screens prompt you for specific information.

The Prompts

1. PRINT OR SPOOL

Enter one of the following letters to specify how the program will output the report:

- P = Write to the printer, if possible.
- S = Spool output to disk.
- O = Prompt the operator to select Print or Spool.

The default response is P (Print). If you specify P and the printer is not available, the program will send a message to the operator asking whether to retry printing or to direct program output to a spool file.

2. SPOOL FILENAME

Enter the name of the disk file to which the program will direct output if the spool option is selected. (If you specify a name less than nine characters long, the Program Generator appends the operator's terminal number to the filename to make it unique.)

3. PAGE WIDTH

Enter 80 or 132. This entry defines the number of columns per line of your report. The default width is 132.

4. NO. OF PRINT LINES PER PAGE

Enter an integer in the range 0 through 99. This number indicates the number of print lines permitted for the report. Exceeding this number will cause a form feed. Remember that the number of lines per page includes the title, legend, and header information. The default value is 60 lines per page.

5. IS RUNTIME TO BE PRINTED ?

Enter Y if you want the program to place the date and time at the top of each page of the report. Enter N (the default) if you do not want runtime information printed.

6. REPORT TITLE

Enter the title you want to appear on the first page of the report. The program will center the title; you do not have to enter any leading blanks. If you selected a 132-column report, your title may run up to 90 characters. For an 80-column report, the title cannot exceed 60 characters.

Enter the title in 30-character fields. The program ignores trailing blanks when centering the title.

7. DO YOU HAVE LEGEND LINES ?

Enter Y if you want to include up to nine legend lines following the title. These lines appear on the first page of the report and optionally may appear on subsequent pages. Enter N (the default) if you will not use any legend lines in this report.

HOW MANY FROM PAGE 2

If the program uses legend lines, it optionally may print them on pages after the title page. Indicate how many of the legend lines (starting with the first one) you want to appear. Enter an integer from 0 to the number of legend lines you specified (up to 9). The default response is 0. (If you want to change your response to this question during the Any Change cycle, enter 7 as the prompt number.)

8. DO YOU HAVE HEADING LINES ?

Enter Y if the report uses headings. The heading lines appear after any legend lines. Each page of the report includes the heading lines. Enter N (the default) if the report does not use heading lines.

What Next?

If you indicate that the report uses legend lines, continue with the next screen description (PG-13).

If the report includes heading lines, but *not* legend lines, move on to the heading line description screen (PG-14).

If your report uses neither legend nor heading lines, move ahead to begin the definition of the detail lines (PG-15).

Creating a Report Writer Program

```

PROXI                                     6-JUN-81
Rev x.xx                                  9:29
----  LEGEND AND HEADER DEFINITIONS  ----
      --  LEGEND LINE  n  --

|-----|-----|-----|-----|-----|-----|
| 1       2       3       4       5       6       |
|1234567890123456789012345678901234567890123456|
|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|
|<-----|-----|-----|-----|-----|-----|
|< 7       8       9       0       1       2       3  |
|<7890123456789012345678901234567890123456789012|
|-----|-----|-----|-----|-----|-----|
|<-----|-----|-----|-----|-----|-----|

```

Quick Reference

Key	Action
	Overwrites a character and shifts subsequent characters one column to the left.
\	Inserts a space and shifts subsequent characters one column to the right.

The Program Generator is ready for you to enter a legend line. This screen repeats until you define a ninth legend line, or press FINISHED.

Legend lines appear on the page exactly as you enter them; they are not centered.

You cannot enter blank legend lines. If you enter a pair of NEW LINEs the Program Generator assumes you have completed all entries (as if you had pressed FINISHED).

You may use all PROXI cursor controls when entering the legend line. (See Table 1-2.) You may also use the two editing characters shown in the Quick Reference section for screen PG-13.

The legend line screen divides the line into two 66-character fields. If you are using an 80-column format, remember not to exceed 80 characters when entering a legend line; the Program Generator will truncate excess characters.

Note that once you move to the second segment of the line you cannot return to the first part unless you use the Any Change cycle or press ESC. Control characters (such as CTRL A) operate in relation to the beginning of the current segment only.

If you are changing an existing report, you may use the INSERT and DELETE keys to add or remove a legend line. Press the appropriate key when the Any Change question appears.

What Next?

After you complete the last legend line, the Program Generator moves on to the next screen. If you indicated that this report uses heading lines, continue with the next screen description (PG-14).

If the report does not use any heading lines, the Program Generator begins a series of screens in which you describe this report's detail lines (PG-15).

The Program Generator is ready for you to enter a header line. It repeats this screen until you have defined a ninth header line or have pressed FINISHED.

Header lines appear exactly as you enter them; they are not centered on the page. Be sure to line up column headings with the fields they describe.

You cannot enter blank header lines. If you enter two NEW LINES the Program Generator assumes you have completed your entries (as if you had pressed FINISHED).

You may use all PROXI cursor controls when entering a header line. (See Table 1-2.) You may also use the editing characters shown in the Quick Reference section on the previous page.

The header line screen divides the line into two 66-character fields. If you are using an 80-column format, remember not to exceed 80 characters when entering a header line; the Program Generator truncates excess characters.

Note that once you move to the second segment of the line you cannot return to the first part unless you use the Any Change cycle or press ESC. Control characters (such as CTRL A) operate in relation to the beginning of the current segment only.

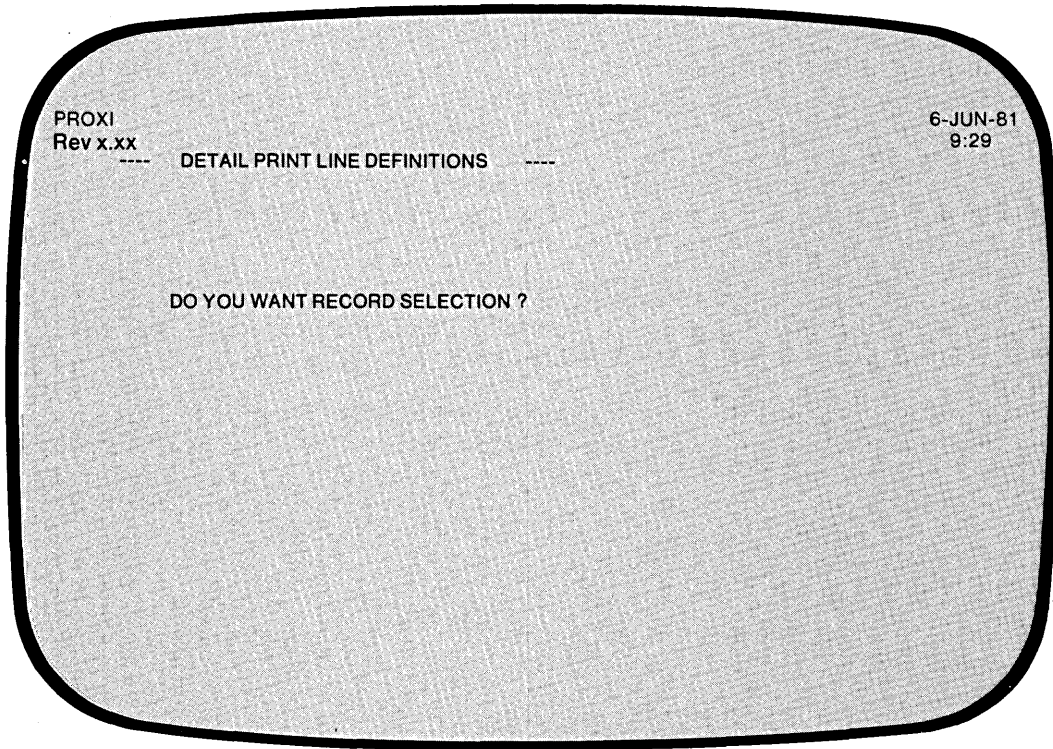
If you are changing an existing report, you may use the INSERT and DELETE keys to add or remove a header line. Press the appropriate key when the Any Change question appears.

What Next?

After you define all heading lines, the Program Generator requests information about this report's detail lines (PG-15).

PG-15
Creating a Report Writer Program

Program Generator



Quick Reference

Prompt	Default	Range
RECORD SELECTION	N	Y or N

The Program Generator presents this screen after you provide all necessary information about the report's legend and header lines. This data-entry screen is the first in a series of screens that you use to describe the detail lines for this report.

The Program Generator allows you to establish separate groups of detail lines. A set of record selection criteria controls each group. You may enter from one to 99 logical tests that will control a group of detail lines. If the result of the tests is true, the program will print the associated detail lines.

After you define a set of selection criteria and the related detail line(s), the Program Generator again asks if you want record selection. You may define another set of criteria to govern another group of detail lines. (Refer to screen PG-20).

The Record Selection feature not only allows you to manage separate groups of details lines; it also permits you to selectively process records in the data file. The program ignores records that do not pass any of the selection criteria.

The Prompt

DO YOU WANT RECORD SELECTION ?

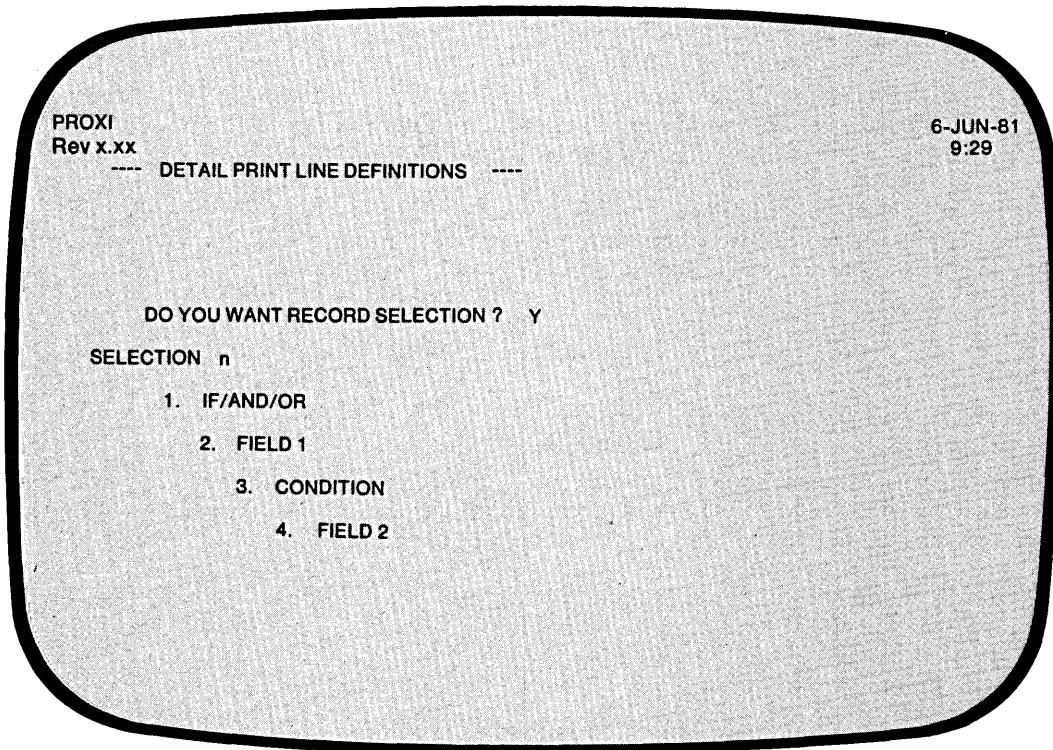
Enter Y if you want the program to test each data record against one or more logical tests (which you will define later). Each record that passes the test(s) is reported in the corresponding detail lines.

Enter N if you do not want to use selection criteria. In this case all data-file records will use the same detail line(s).

What Next?

If you request record selection, the Program Generator displays an overlay screen to gather the logical tests (PG-16).

If you do not want record selection, you immediately begin a series of screens with which you describe each detail line for this report (PG-17).



Quick Reference

Prompt	Default	Range
IF/AND/OR	None	IF, AND, OR, or the INSERT, DELETE, or FINISHED key
FIELD 1	None	A field name
CONDITION	EQ	EQ, NE, LT, LE, GT, or GE
FIELD 2	None	A field name or constant

This overlay screen appears if you indicate that you want the program to apply selection criteria to data-file records when building detail lines. You may enter up to 99 logical tests for record selection. The screen repeats until you enter 99 tests or press FINISHED at the first prompt following the final test entry.

The result of these test series must be TRUE, otherwise the program will not process the record.

The Prompts

1. IF/AND/OR

Specify the appropriate logical operator for the indicated test. (The Program Generator supplies "IF" for the first test.) After you have entered all the logical tests, press FINISHED.

2. FIELD 1

Enter the first field for this logical test. You may specify any field defined in the principal data file or a reference file. There is no default entry.

3. CONDITION

Enter one of the following conditional operators:

EQ (equal to)
NE (not equal to)
LT (less than)
LE (less than or equal to)
GT (greater than)
GE (greater than or equal to)

The default response is EQ.

4. FIELD 2

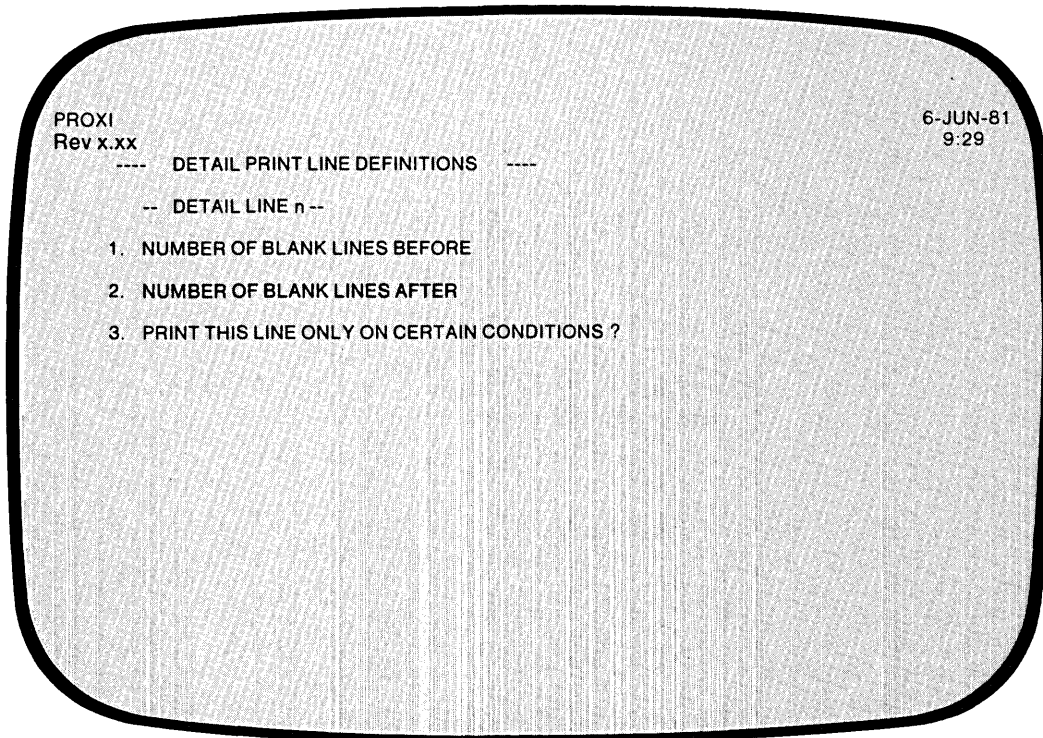
Enter a field name or literal value to be compared with FIELD 1. The field name may refer to a field in the principal data file, a reference file, or a Working Storage data name. (Be sure to enclose non-numeric literals in quotation marks.) There is no default.

What Next?

This screen repeats, allowing you to specify up to 99 logical tests for controlling record selection. After you enter the final test, press FINISHED at the first prompt. The Program Generator then prompts you to define the detail lines associated with this set of selection criteria (PG-17).

PG-17
Creating a Report Writer Program

Program Generator



Quick Reference

Prompt	Default	Range
BLANK LINES BEFORE	None	0 - 9, P, T, or the FINISHED, INSERT, or DELETE key
BLANK LINES AFTER	0	0 - 9, P, T, or *
CONDITIONAL PRINTING	N	Y or N

The Program Generator is ready to define a specific detail line. This screen permits you to specify line feeds before and after the detail line. You may also establish up to 99 conditional tests to be performed to determine whether or not to print the detail line. You return to this screen after completing the definition of each detail line. To end the cycle, press FINISHED at the first prompt.

The Prompts

1. NUMBER OF BLANK LINES BEFORE

Enter a digit (0 through 9), P, or T. The digit specifies the number of line feeds the program will enter before printing this detail line. (Take into account the blank lines printed after the previous detail line, if any.) The letter T signifies a top-of-page (form feed) operation. The letter P also inserts a form feed, but resets the page number counter to one. There is no default entry.

Press FINISHED at this prompt after you've defined all detail lines in this group.

2. NUMBER OF BLANK LINES AFTER

Indicate how many lines you want the program to print after this detail line. The rules given for the previous prompt apply with two exceptions: 1) The default response is 0. 2) You may also enter an asterisk (*), which identifies this field as one to be conditionally printed as part of the next print line.

3. PRINT THIS LINE ONLY ON CERTAIN CONDITIONS ?

Enter Y if you want this detail line to appear only if certain conditions are met. You may specify up to 99 logical tests to be performed to determine whether or not this line is to be printed. If you answer Y, you will define the logical tests shortly. Enter N (the default) if you do not want conditional testing for this line.

What Next?

If you want to specify conditions for printing, continue with the next screen description (PG-18).

If you do not want conditional testing for this detail line, move on to the screen with which you define each field within this detail line (PG-19).

Press FINISHED if

- you have already defined all detail lines for this report. The Program Generator then requests information about control breaks (PG-21).
- you have defined all detail lines to be associated with a particular set of record selection criteria. The Program Generator then allows you to set up another set of selection criteria that governs one or more additional detail lines (PG-20).

PG-18
Creating a Report Writer Program

Program Generator

```

PROXI
Rev x.xx
-----
DETAIL PRINT LINE DEFINITIONS
-----
--
DETAIL LINE n
--
1. NUMBER OF BLANK LINES BEFORE
2. NUMBER OF BLANK LINES AFTER
3. PRINT THIS LINE ONLY ON CERTAIN CONDITIONS ?
CONDITION n
1. IF/AND/OR
2. FIELD 1
3. CONDITION
4. FIELD 2
    
```

6-JUN-81
9:29

Quick Reference

Prompt	Default	Range
IF/AND/OR	None	IF, AND, OR, or the FINISHED, INSERT, or DELETE key
FIELD 1	None	A field name
CONDITION	EQ	EQ, NE, LT, LE, GT, or GE
FIELD 2	None	A field name or constant

This overlay screen appears if you indicate that you want the detail lines printed only on certain conditions. You may specify up to 99 logical tests which the program will perform for the detail line. If the result of the tests is TRUE, the line will be printed.

The Prompts

1. IF/AND/OR

Specify the appropriate logical operator for the indicated test. (The Program Generator supplies "IF" for the first test.) After you enter all the logical tests, press FINISHED.

2. FIELD 1

Enter the first field for this logical test. You may specify any field defined in the principal data file or a reference file. There is no default entry.

3. CONDITION

Enter one of the following conditional operators:

EQ (equal to)
NE (not equal to)
LT (less than)
LE (less than or equal to)
GT (greater than)
GE (greater than or equal to)

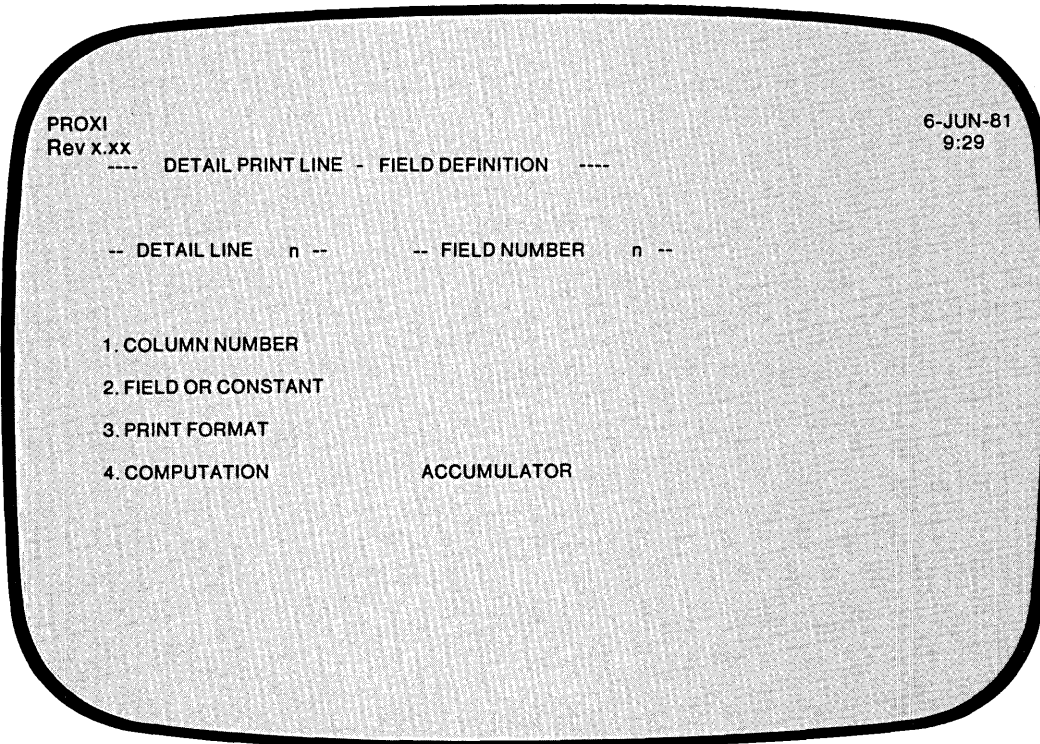
The default response is EQ.

4. FIELD 2

Enter a field name or literal value to be compared with FIELD 1. The field name may refer to a field in the principal data file, a reference file, or a Working Storage data name. (Be sure to enclose non-numeric literals in quotation marks.) There is no default.

What Next?

This completes your definition of the printing of this line. Now the PROXI Program Generator presents a series of data-entry screens in which you describe each field within the detail line (PG-19).



Quick Reference

Prompt	Default	Range
COLUMN NUMBER	None	1 - 132 or 999
FIELD OR CONSTANT	None	A field name or constant
PRINT FORMAT	(See description)	A PICTURE clause
COMPUTATION	No computation	(See Table 2-1)
ACCUMULATOR	None	A01 - A99

Table 2-1. Computation Symbols

Symbol	Computation
=	Axx = data item
+	Axx = Axx + data item
-	Axx = Axx - data item
*	Axx = Axx * data item
/	Axx = Axx / data item
%	Axx = 100 * Axx / data item
T	Axx = automatic total + data item

The Program Generator is ready to gather information about each field within this detail line. You may define up to 99 fields. This screen repeats until you have defined all fields. Press FINISHED to end the series.

If the detail line includes multiple print fields, specify the fields from left to right (in ascending column positions). You may enter Column 999 fields at any time provided that order is consistent with the operation to be performed.

The Prompts

1. COLUMN NUMBER

Enter the column position (1-132) of the field's first character. You may also specify 999 for a nonprinting field.

2. FIELD OR CONSTANT

Identify the data item or constant associated with this field. The data item may appear in the principal data file, a reference file, Working Storage, or a literal. If you enter a non-numeric constant, be sure to enclose it with quotation marks.

3. PRINT FORMAT

Enter a PICTURE clause for this field. If the field displays a non-numeric literal, the default print format is an alphanumeric field of the same length. If the field reports a data item for which a Data Dictionary entry exists, the default print format is the data item's defined picture.

If you intend to use automatic totaling for this field, be sure that use automatic totaling for this field, be sure that the picture is large enough to contain the total field.

If you specify an alphanumeric field, the Program Generator skips the next two prompts.

4. COMPUTATION

To perform a computation using this field's data item, enter the appropriate computation symbol (listed in Table 2-1). If you make no entry, no computation occurs.

If you request automatic totaling (by entering T), the Program Generator skips the next prompt.

ACCUMULATOR

Enter the name of the accumulator to be used for this computation. There are 99 PROXI accumulators (A01 through A99) that you may use. Remember to prefix the number with the letter A. You must provide your own code for initializing and resetting these accumulators. If you request automatic totaling, the Program Generator skips this prompt because the program will use a special set of totaling accumulators.

You need not specify computation for printing fields. To bypass these prompts, simply press NEW LINE at the COMPUTATION and ACCUMULATOR prompts. Only Column 999 fields require entries for these two prompts.

What Next?

After you define each field within this detail line, press FINISHED at the first prompt. The Program Generator then allows you to define another detail line (PG-18).

PG-20
Creating a Report Writer Program

Program Generator

```
PROXI                               6-JUN-81  
Rev x.xx                             9:29  
-----  DETAIL PRINT LINE DEFINITIONS  -----  
  
--  DETAIL LINE   n  --  
  
ANY MORE RECORD SELECTIONS ?
```

Quick Reference

Prompt	Default	Range
RECORD SELECTIONS	N	Y or N

As we explained earlier (see PG-15), you may define multiple sets of record selection criteria, each set governing a group of detail lines. If you want to establish another set of record selection criteria, answer Y. The Program Generator then repeats the series of screens in which you specify the selection tests and the associated detail line(s).

Answer N if you do not want to define another set of selection criteria.

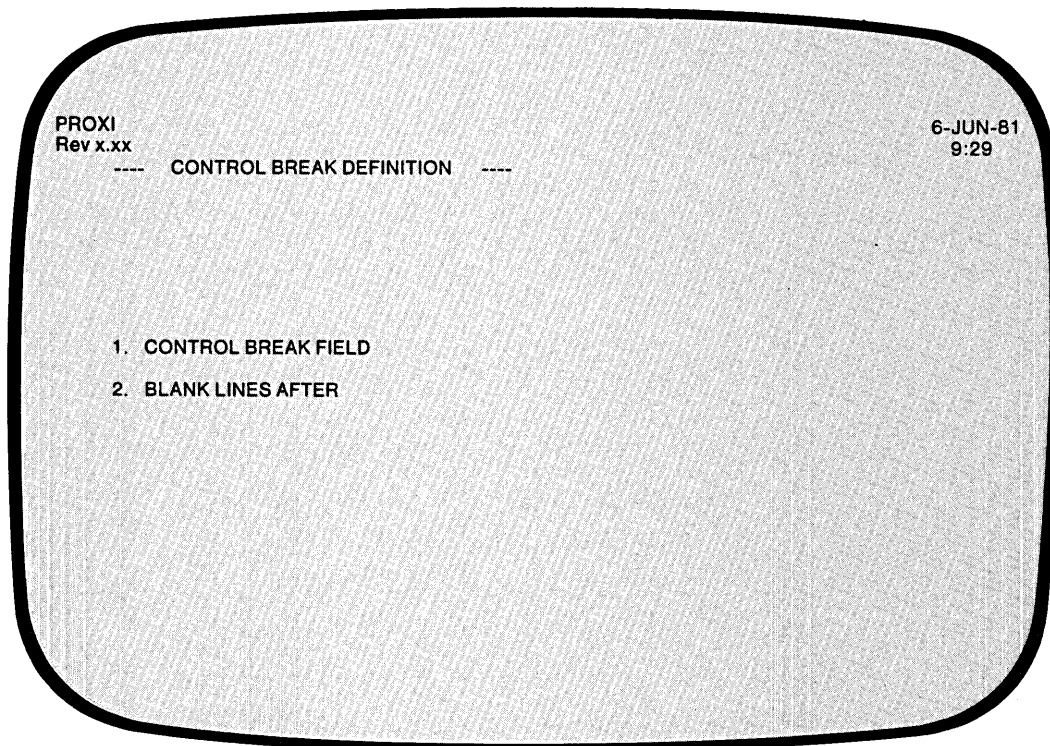
What Next?

If you answer N, the Program Generator begins a series of screens in which you describe control breaks and total lines. Continue with the next screen description (PG-21).

If you want to specify additional record selection criteria and related detail lines, answer Y. The Program Generator presents an overlay screen, as before, to gather the logical tests to be applied to each data-file record (PG-16).

PG-21
Creating a Report Writer Program

Program Generator



Quick Reference

Prompt	Default	Range
CONTROL BREAK FIELD	None	A field name, EOF, or the FINISHED, INSERT, or DELETE key
BLANK LINES AFTER	None	0 - 9, T or P

This screen appears after you complete the definitions of all print lines and their fields. The Program Generator is ready to set up control break processing. You may request up to nine control break levels for this report.

The program monitors the control break fields and prints a total line when a change occurs in the field. A higher level control break automatically generates a control break (and total line) for each lower level.

When you define a control break, the Program Generator allows you to specify total lines for that break level. After completing all subsequent screens, you will see this screen again. You may then define the next control break level and its associated total lines.

Enter the control breaks in order, beginning with the lowest level (the least comprehensive) through the highest level (which is often the end-of-file condition).

After you define all control breaks and total lines, press FINISHED at the this screen's first prompt.

The Prompts

1. CONTROL BREAK FIELD

Enter the name of a field in the principal data-file record, or EOF to indicate the end of the principal data file (or the end of the operator-selected KEY RANGE). Press FINISHED if you have defined all control breaks and their accompanying total lines. There is no default entry.

2. BLANK LINES AFTER

Enter a digit (0 through 9), P, or T. A digit indicates the number of blank lines to be printed after the total line(s) for this control break. The letter T requests a page break (form feed) after the total lines for this control break. A P requests a form feed and also a resetting of the page counter to one.

What Next?

If you have just defined a control break, continue with the next data-entry screen (PG-22), which begins a series of screens that request information about total lines.

Press FINISHED at the first prompt to indicate that you have completed definitions of all control breaks and associated total lines. Then you move on to the COBOL Program Generator screen (PG-24).

PG-22
Creating a Report Writer Program

Program Generator

```
PROXI  
Rev x.xx  
----- TOTAL LINE DEFINITIONS -----  
CONTROL BREAK xxxxxxxx  
  
-- TOTAL LINE n --  
  
NUMBER OF BLANK LINES BEFORE
```

6-JUN-81
9:29

Quick Reference

Prompt	Default	Range
BLANK LINES BEFORE	None	0 - 9, T, P, or the FINISHED key

You can set up spacing or form feeds before this total line. You may enter up to nine blank lines before a total line. PROXI also permits you to enter a page break (form feed) before the total line and to optionally reset the page counter. The total line number and the current control break field appear above the prompt.

The Prompt

NUMBER OF BLANK LINES BEFORE

Enter a digit (0 through 9), P, or T. The digit indicates the number of lines to skip before printing this total line. The letter T requests a page break (form feed) before printing the total. The letter P requests a form feed accompanied by a resetting of the page number counter to one. There is no default.

What Next?

If you press FINISHED to indicate that you've defined the last total line for this control break field, the Program Generator allows you to define the next control break field, if any (PG-21).

If you complete this screen, the Program Generator moves on to let you describe each field within this total line (PG-23).

PG-23
Creating a Report Writer Program

Program Generator

```

PROXI
Rev x.xx
6-JUN-81
9:29
---- TOTAL PRINT LINE - FIELD DEFINITION ----
CONTROL BREAK xxxxxxxx
-- TOTAL LINE n -- FIELD NUMBER n --

1. COLUMN NUMBER
2. FIELD OR CONSTANT
3. PRINT FORMAT
4. COMPUTATION ACCUMULATOR
    
```

Quick Reference

Prompt	Default	Range
COLUMN NUMBER	None	1 - 132, or 999
FIELD OR CONSTANT	None	A field name or constant
PRINT FORMAT	(See description)	A PICTURE clause
COMPUTATION	No computation	(See Table 2-2)
ACCUMULATOR	None	A01 - A99

Table 2-2. Computation Symbols

Symbol	Computation
=	Axx = data item
+	Axx = Axx + data item
-	Axx = Axx - data item
*	Axx = Axx * data item
/	Axx = Axx / data item
%	Axx = 100 * Axx / data item
T	Axx = automatic total + data item

The Program Generator is ready to gather specific layout information for this total line. This screen requests information about a particular field within the total line. You may define up to 99 separate fields for each total line.

If the total line includes multiple print fields, specify the fields from left to right (in ascending column position).

The Prompts

1. COLUMN NUMBER

Enter the column position of the field's first character. You may also specify 999 for a nonprinting field. You may enter a Column 999 field at any time provided that the sequence of computations is consistent with the operation to be performed.

2. FIELD OR CONSTANT

Identify the data item or constant associated with this field. The data item may appear in the principal data file, a reference file, or Working Storage. It may also be a literal or constant. If you enter a non-numeric constant, be sure to enclose it with quotation marks.

3. PRINT FORMAT

Enter a PICTURE clause for this field. If the field displays a non-numeric literal, the default print format is an alphanumeric field of the same length. If the field reports a data item for which a Data Dictionary entry exists, the default print format is the data item's defined picture.

If you intend to use automatic totaling, be sure to provide a picture large enough to contain the resulting total.

4. COMPUTATION

To perform a computation using this field's data item, enter the appropriate computation symbol (listed in Table 2-2). If you make no entry, no computation occurs.

ACCUMULATOR

Enter the name of the accumulator to be used for this computation. There are 99 accumulators (A01 through A99) you may use. Remember to prefix the number with the letter A when making your entry. You must provide your own code for initializing and resetting these accumulators. If you request automatic totaling, the Program Generator skips this prompt because the program uses a special set of totaling accumulators.

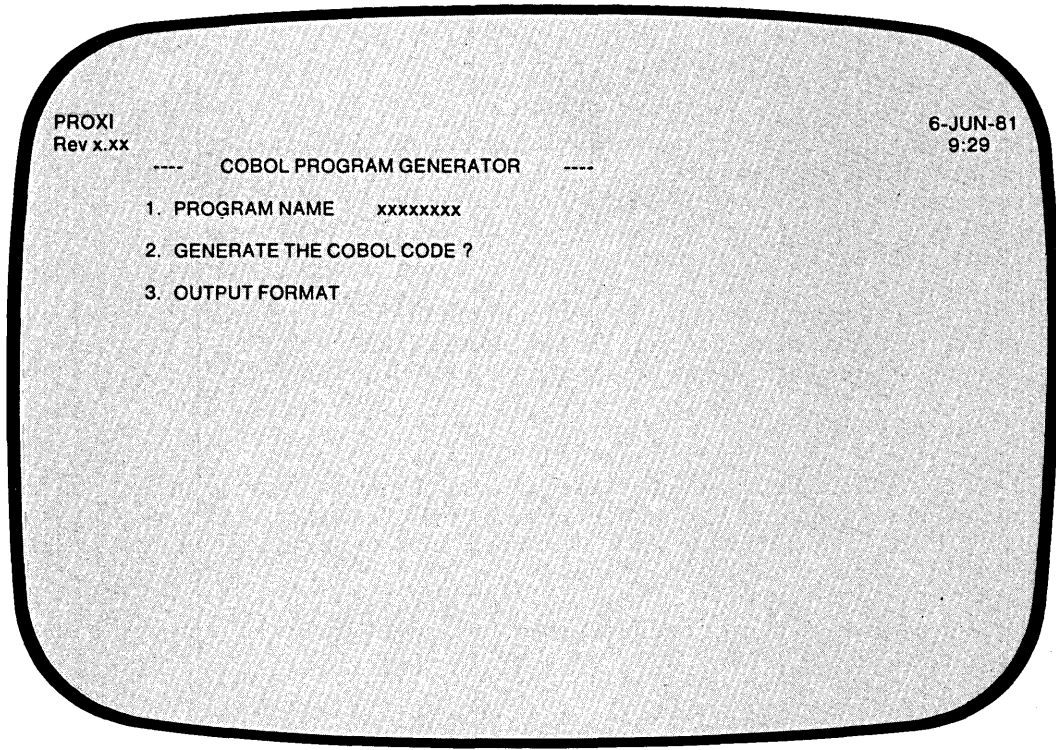
You need not specify computation for printing fields. To bypass these prompts, simply press NEW LINE at the COMPUTATION and ACCUMULATOR prompts. Only Column 999 fields require entries for these prompts.

What Next?

This screen repeats, allowing you to define each field within this total line. Press FINISHED at the first prompt after you have defined all the fields. The Program Generator then allows you to define the next total line, if any (PG-22).

PG-24
Creating a Report Writer Program

Program Generator



Quick Reference

Prompt	Default	Range
GENERATE COBOL CODE	Y	Y or N
OUTPUT FORMAT	1	1 - 2

The Program Generator displays this screen after you've defined all total lines and details lines for the report writer program. It supplies the program name which you gave earlier.

The Prompts

2. GENERATE THE COBOL CODE ?

Enter Y if you want to generate the code for your Report Writer program. You should perform this operation only after you complete all other PROXI operations for this program (such as defining data files). Enter N if you want to return to the Program Generator menu, skipping the code generation step.

3. OUTPUT FORMAT

Enter 1 for card format (code with line numbers), or 2 for CRT format (code without line numbers).

What Next?

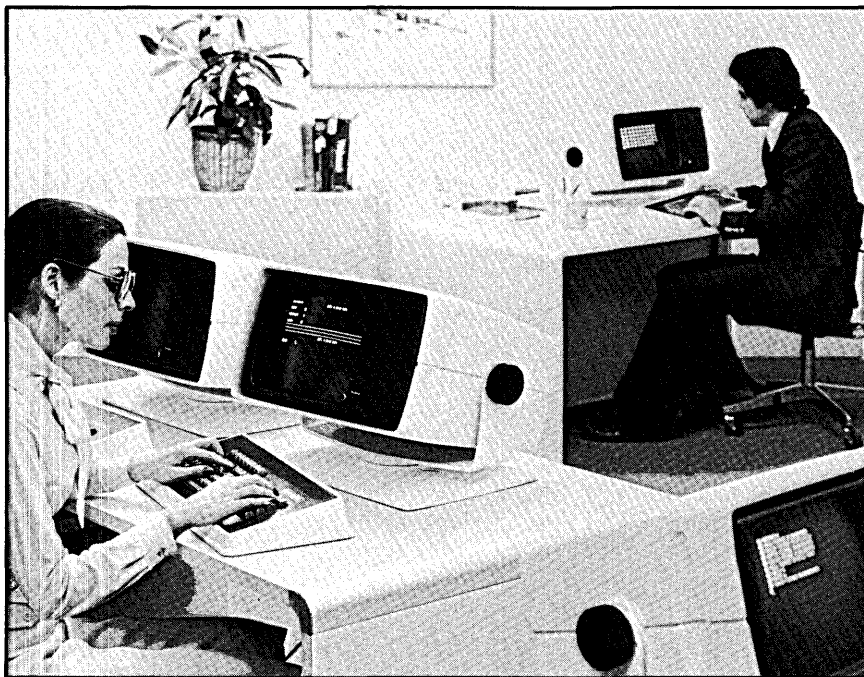
If you request code generation, you'll receive the message

LOADING PARAMETERS FOR program - PLEASE WAIT

Then, as the Program Generator builds each section of source code, you'll see the division and section titles displayed on the screen.

You now return to the Program Generator menu (PG-1), whether or not you generated COBOL code.

If you generated code, you may proceed to compile it after you end the PROXI session. Be sure that you have completed all necessary operations before attempting to compile the code. If you neglect to create the required copy files, for example, the compiler will be unable to compile the source code.



The Form Printing Program: An Overview

The Form Printing Program allows you to produce a series of single- or multiple-page forms using the records of a principal data file and up to nine reference files. You can request that the program output the forms on a printer, or spool them to a disk file for printing at a later time. This program is designed to complete preprinted forms (such as invoices) or produce formatted output (such as mailing labels).

The Form

The PROXI system does not force you to conform to a standard format. Instead, it allows you to design your own format for a variety of uses, from invoices to mailing labels. During the interactive session with the Program Generator, you describe

- top-of-form lines
- record selection procedures
- detail line formats
- computations
- a control break and its total lines
- page numbering
- page break lines
- printing criteria

Figure 2-9 illustrates a sample PROXI form. Refer to this example as we describe the features of the PROXI form.

Product No.	Description	Qty.	Price each	Total
BL808	LUCKENBOOTH BROOCH, sterling	1	\$ 35.00	\$ 35.00
HD127	SKEAN DHUB, imitation cairngorm	1	\$ 65.50	\$ 65.50
HG107	CABER	1	\$ 75.00	\$ 75.00
	*****SPECIAL HANDLING CHARGE			\$ 9.00
PD375	FULL DIRK, hand carved	1	\$ 305.00	\$ 305.00
SP418	SPORRAN, hand tooled, runic design	1	\$ 38.50	\$ 38.50
SUBTOTAL:				\$ 526.00
Less credit vouchers, discount:				\$ 50.00
Vermont 3% sales tax:				\$ 0.00
Postage and handling:				\$ 7.50
TOTAL:				\$ 483.50

SD-02888

Figure 2-9. A Sample PROXI Form

Conditional Printing

You may specify printing conditions for top-of-form lines, detail lines, page break lines, and total lines. The program will perform up to 99 logical tests on data items to determine whether or not to print the line.

Print Fields

You may specify up to 99 separate printing fields within a detail line. Associated with each print field is a data item or constant. The program will print the value of the data item or constant in the associated field. You may perform various computations on field values prior to printing.

Conditional Fields

The Form Printing Program allows you to set up conditionally printed fields within a top-of-form, detail, page break, or total line. (We describe these line types below.) To do this you must:

- 1) Define one line containing *only* the conditional field(s). When you define this line, you enter an asterisk (*) following the BLANK LINES AFTER prompt. (Refer to screens PG-28, PG-33, PG-37, or PG-41.) You must also supply the printing criteria that will apply to both this field *and* the line that will contain the field.
- 2) Define another line containing *only* the field(s) that you want to appear each time the line is printed. You may specify printing conditions for this line as well.

Let's look at an example. We define one detail line that represents the conditional field. It looks like this:

SHIPPED SEPARATELY

The condition we specify for this field is

IF ORDER-ITEM GE 9000

Next we define the part of the detail line that will always appear. It includes the following fields: ORDER-ITEM, DESC, PRICE, QTY, and ITEM-TOTAL.

9999 xxxxxxxxxxxxxxxx \$ \$9.99 999 \$ \$ \$9.99

When processing the data-file records, the Form Printing Program will consider these two lines as a single line (in this case a detail line), printing the first portion only when conditions permit. A portion of the resulting form might look like this:

2381	Flashes, pr.	\$ 8.00	1	\$ 8.00	
4238	Jabot	\$ 36.95	1	\$ 36.95	
9127	Haggis	\$ 24.00	1	\$ 24.00	SHIPPED SEPARATELY

Top-of-Form Lines

You may include up to nine top-of-form lines on the first page of a form. These lines may print heading information if you are not using preprinted forms, or they may refer to the information to follow. The Form Printing Program prints these lines after it has read all records pertaining to the first detail line. This allows you to include information specific to the first detail line in the top-of-form lines. (This could be information such as the name and address of the customer who made an order.) You may specify conditional printing criteria; otherwise, all top-of-form lines will appear at the top of each form.

Detail Lines

A detail line is the basic information unit of a form. Each detail line may contain up to 99 print fields. These fields display a literal or report record field values, the result of a computation, or a constant.

In the simplest case, one record from the principal data file produces one detail line. Every time the program reads a record, it generates a detail line for the form.

If you want to print more information (or arrange it differently), you can establish multiple detail lines for each data-file record. The program will then produce two or more detail lines as it processes each record.

To provide even more flexibility, the Form Printing Program allows you to specify conditions for printing a detail line. Now you can print a detail line only if certain conditions are true.

Suppose you need to print different groups of detail lines, depending on the information taken from the data-file record or resulting from a computation. The Form Printing Program allows you to establish separate conditionally-printed groups of detail lines. A set of record selection criteria governs each group. This feature allows you to print a group of three detail lines under one set of conditions, and two detail lines under another set of conditions. When you use record selection criteria, you can selectively process records in the principal data file. The program will ignore records that do not pass any of the selection criteria.

With each detail line you have the option of advancing to the next page. If you choose to advance, either you may request the program to reset the page counter to 1, or you may continue incrementing the page counter.

Page Break Lines

A page break line appears at the bottom of all but the last page of a multiple-page form. The program prints the page break line if it has printed the last allowable detail line and is not a control break (in which case it would print a total line). The program prints these lines before it reads records for the next detail line. Therefore you may include information from the current detail line as well as subtotals up to this point.

Control Break and Total Lines

You define one control break field for the form. When the program detects a change in the value of the control break field, it generates one or more total lines. Several records may supply information to the same form, depending on the control break field.

An end-of-file condition on the principal data file causes a control break, generating total lines for the last form.

As with detail lines, you may specify blank lines before (but not after) a total line, and you may specify computations for field values.

Accumulators and Computations

Each print field has an associated data item or constant. You may perform simple calculations using the field value and a specified accumulator. You may use any of 99 accumulators, named A01 through A99. If you do use any of these accumulators, you must provide your own code for initializing and resetting them.

Automatic Totaling

The Form Printing Program offers an automatic field totaling facility. You do not have to manipulate values and accumulators; the program can do it for you automatically. You may request automatic totaling for any numeric field in a detail or total line.

If you want automatic totaling for a field, simply enter the code T at the COMPUTATION prompt for that field. The program will then add the contents of the field to the auto-total accumulator reserved for that field. The program can supply up to 99 auto-total accumulators.

When a control break occurs, the program automatically prints the accumulator value using the field's column position and PICTURE. (You may want to assign the field's PICTURE with the total in mind.) Be sure that you have provided enough fields in the total line definition to handle the automatic totals.

The program automatically sets the appropriate accumulators to zero when multiple control break fields are in effect.

Page Numbering

The Form Printing Program automatically numbers the pages of a form. It sets the page counter to 1 for each new form, and increments as each page is printed. You may reset the page counter to 1 within a multiple-page form when you request a detail line to be printed at the top of the next page.

Logical Testing

The Form Printing Program allows you to specify record selection criteria for detail lines and printing conditions for top-of-form, detail, page break, and total lines. You may specify from one to 99 logical tests that the program will use to determine if it will print the appropriate line. If the result of the test series is false, the program will not print the line.

Each test consists of four parts arranged in the following format:

$$\left. \begin{array}{l} \text{IF} \\ \text{AND} \\ \text{OR} \end{array} \right\} \text{field-name} \left. \begin{array}{l} \text{EQ} \\ \text{NE} \\ \text{LT} \\ \text{LE} \\ \text{GT} \\ \text{GE} \end{array} \right\} \left. \begin{array}{l} \text{field-name} \\ \text{literal} \end{array} \right\}$$

The first test always begins with IF. You'll receive prompts for the remaining three parts of the first test.

The program will apply the following hierarchy of evaluation when you include more than one test:

1. Tests joined by AND
2. Tests joined by OR
3. Nested tests joined by IF

Form Input

As we mentioned earlier, you may use up to nine reference files in addition to the principal data file.

Range Selection

You may allow the operator to select a particular range of records within the principal data file, to be used as input to the program. The operator specifies a start key and an ending key to identify the range of records to be processed.

Building a Form Printing Program

To build a Form Printing Program you must

- Use the PROXI File Definitions module to define all data files to be used by the program.
- Create the main program framework through PROXI's Program Generator module. This chapter explains how to do this.

The flow chart in Figure 2-10 diagrams the steps you take to build a Form Printing Program.

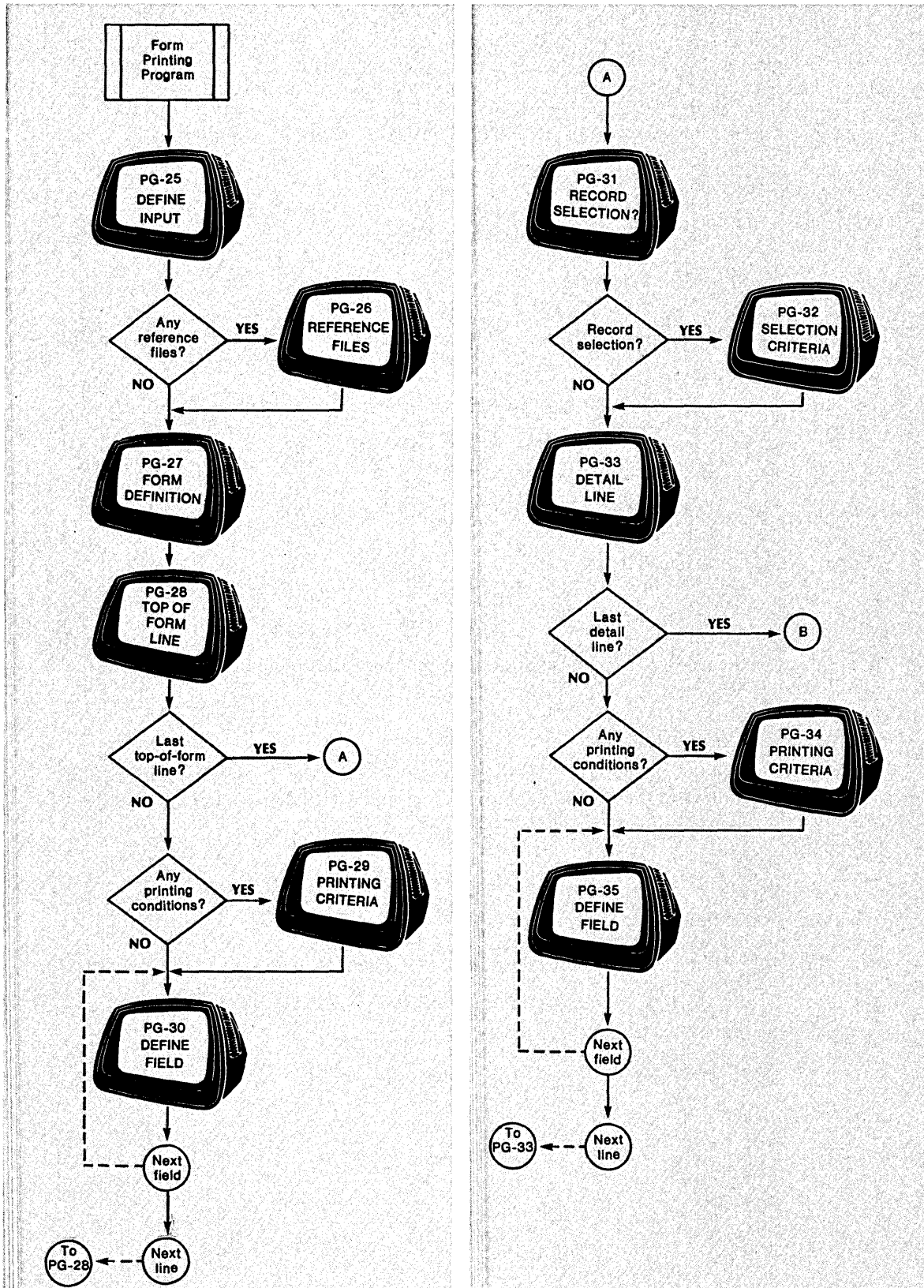


Figure 2-10. Building a Form Printing Program (continues)

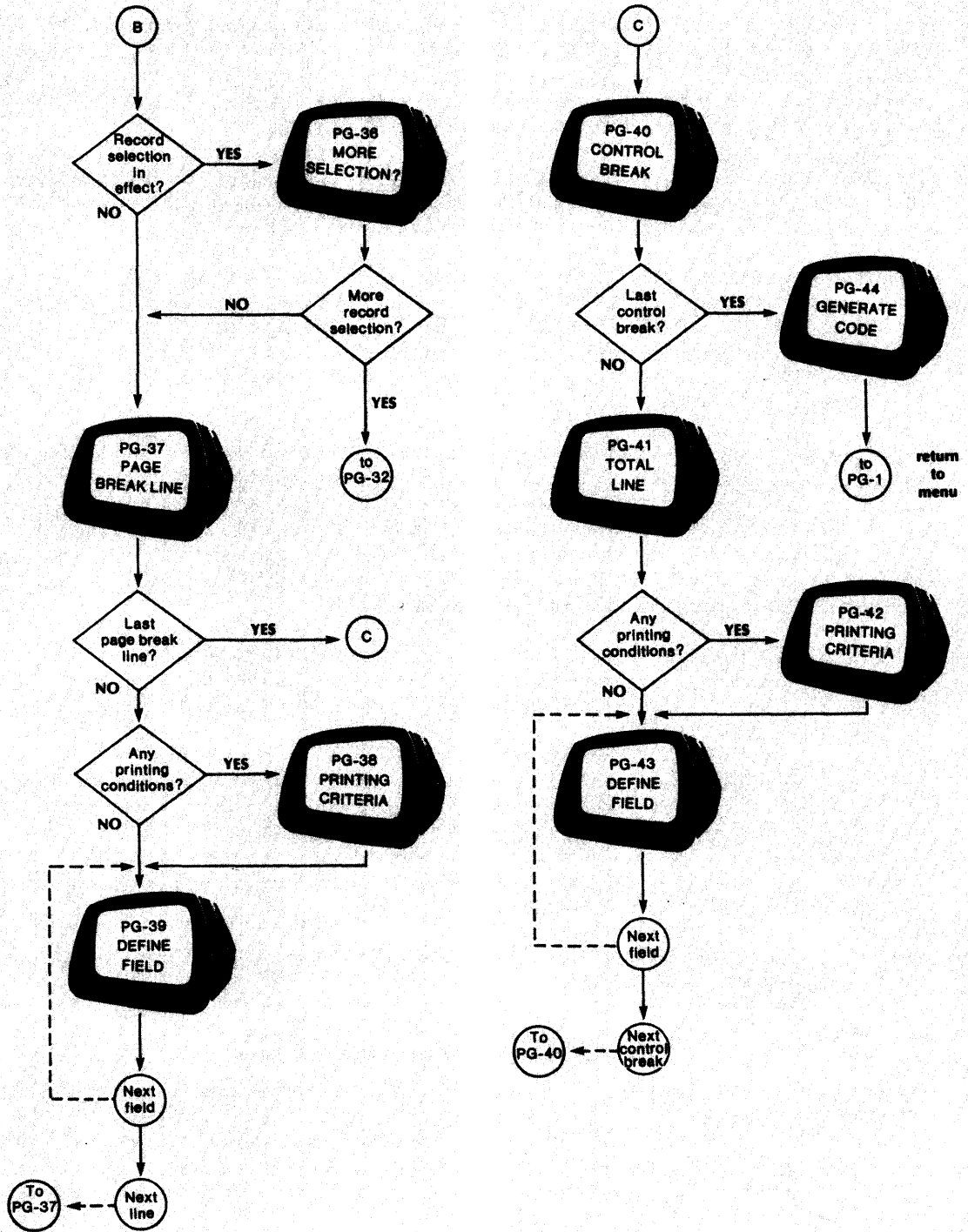


Figure 2-10. Building a Form Printing Program (concluded)

The Interactive Session

As you can see in Figure 2-10, you begin by defining the input to the File Printing Program. You name the principal data file (PG-25), and, optionally, any reference files (PG-26).

After identifying the input to the program, you begin to lay out your form (PG-27). In addition, you can allow the operator to select a particular range of records from the principal data file. You also name the disk file to which output will be spooled if the printer is not available. You specify the number of detail lines per page, the line on which the first detail line will appear, and the last line of the page available for printing.

The next step is to define the top-of-form lines that will appear on the first page (PG-28). You may supply printing criteria for these lines (PG-29). The Program Generator requests information about each field within each top-of-form line (PG-30).

Before gathering information about the detail lines, the Program Generator asks if you want to set up selection criteria for data-file records (PG-31). This feature allows you to specify up to 99 logical tests to be performed to determine if a record should be processed (PG-32). You may define one or more detail lines to be associated with this set of selection criteria. If necessary, you may establish independent sets of selection criteria, with each set controlling one or more detail lines (PG-36).

The Program Generator then asks you to describe each detail line (PG-33). You specify the number of blank lines to precede and to follow the detail line. You may also specify printing conditions, providing up to 99 logical tests to determine if the program will print the detail line (PG-34). You then describe each field within the detail line; give its column position, length, the data item or constant to be printed in this field, and whether or not you will perform any calculations on this field value (PG-35). In the PROXI system, the Column 999 is a non-printing field in which you can perform calculations.

After defining the detail lines, you describe the page break lines (PG-37). As with the top-of-form and detail lines, you may specify printing conditions for page break lines (PG-38). The Program Generator requests specific information about each field within each page break line (PG-39).

After defining all fields for all of the above lines, you define the control break field (PG-40) and its associated total lines (PG-41). You may specify printing criteria for each total line (PG-42). Again, you must provide definitions for all fields within all total lines (PG-43). Finally you arrive at the last screen (PG-44). Now you may generate the COBOL code framework for the Form Printing Program. You should perform this step only after completing all necessary file definitions.

Refer to the screen descriptions that follow for details about any of the steps you take in building a Form Printing Program.

The PROXI Function Keys

In the screen descriptions that follow we often refer to the function keys FINISHED, INSERT, and DELETE. Please note that only the FINISHED key applies to building a program. The INSERT and DELETE keys provide special functions when you are modifying an existing program. We've mentioned them in this section because you will probably refer to the information here when you are changing a program. So, if you are creating a program, ignore any reference to the INSERT or DELETE keys.



PROXI
Rev x.xx

6-JUN-81
9:29

---- INPUT DEFINITION ----

1. NAME OF FILE TO PRINT
2. KEY RANGE SELECTION ?
LENGTH OF THE KEY
3. ANY REFERENCE FILES ?

Quick Reference

Prompt	Default	Range
NAME OF FILE	None	An AOS data-file name
KEY RANGE SELECTION	N	Y or N
LENGTH OF THE KEY	None	1 - 99
ANY REFERENCE FILES	N	Y or N

You indicated that you want to build a Form Printing Program. The Program Generator is ready to gather information about the principal data file.

After you identify all the data files the program uses, the Program Generator builds the Data Dictionary using information you supplied in the .FD and .SL copy files for each data file. If you have not created these files, the Program Generator supplies default values for certain data items. See Chapter 1 for details.

The Prompts

1. NAME OF FILE TO PRINT

Enter the name of the principal data file. The Form Printing Program uses COPY statements to include datafile.SL (the SELECT statement, datafile.FD (the File Descriptor entries), datafile.WS (the Working Storage entries), and datafile.DS (the Declaratives Section) into its structure. Chapter 4 describes how to create these files using the File Definitions module.

2. KEY RANGE SELECTION ?

Enter Y if you will allow the operator to select a particular range of records to be processed by the Form Printing Program.

If you do not want to give the operator this choice, respond N (which is the default answer). The Program Generator then skips the next prompt.

LENGTH OF THE KEY

Enter an integer in the range 1 through 99 to define the length of the data file's RECORD KEY. There is no default answer. The Program Generator uses your response to build KEY RANGE SELECTION screen formats for the program operator.

3. ANY REFERENCE FILES ?

If the program uses additional data files to process the principal data file, answer Y.

Answer N (the default) if the program does not use any reference files.

What Next?

If you indicate that the program uses reference files, the Program Generator prompts you for information about each of these files (PG-26).

If you answer N to the reference file question, move on to the Form Definition screen (PG-27).

PG-26
Creating a Form Printing Program

Program Generator

```

PROXI                                     6-JUN-81
Rev x.xx                                9:29
----- INPUT DEFINITION -----
1. NAME OF FILE TO PRINT                xxxxxxxx
2. KEY RANGE SELECTION ?                x
   LENGTH OF THE KEY                    nn
3. ANY REFERENCE FILES ?                Y

..... REFERENCE FILE DEFINITIONS .....
-- REFERENCE FILE      n --
4. NAME OF REFERENCE FILE
5. FIELD USED TO ACCESS FILE
    
```

Quick Reference

Prompt	Default	Range
NAME	None	An AOS data-file name, or the FINISHED, INSERT, or DELETE key
ACCESS FIELD	None	A field name

The Program Generator requests information about each reference file. This overlay screen appears only if you answered Y to the question, ANY REFERENCE FILES. Skip this screen description if you answered N.

You may enter up to nine reference files. The reference file number appears above the prompts. The Program Generator continues to prompt you for reference file information until you enter nine reference file names or press FINISHED at the prompt.

After you identify all the data files the program uses, the Program Generator builds the Data Dictionary using information you supplied in the .FD and .SL copy files for each data file. If you have not created these files, the Program Generator supplies default values for certain data items. See Chapter 1 for details.

The Prompts

4. NAME OF REFERENCE FILE

Enter the AOS filename that identifies the data file. There is no default entry. Press FINISHED at this prompt after you have entered all reference file information.

5. FIELD USED TO ACCESS FILE

Enter the name of the field in the principal data file that corresponds to this reference file's RECORD KEY. There is no default answer.

What Next?

After you name all reference files, the Program Generator asks you to define the overall layout of the form (PG-27).

PROXI
Rev x.xx

6-JUN-81
9:29

---- FORM DEFINITION ----

1. PRINT OR SPOOL
2. SPOOL FILE NAME
3. START OF DETAIL LINE #
4. NO. OF DETAIL LINES PER PAGE
5. START OF PAGE BOTTOM

Quick Reference

Prompt	Default	Range
PRINT OR SPOOL	P	P, S, or O
SPOOL FILE NAME	None	An AOS filename
START DETAIL LINE #	0	0 - 99
NO. DETAIL LINES	0	0 - 99
START PAGE BOTTOM	None	0 - 99

This screen appears after you define the data files that the Form Printing Program uses. The Program Generator is ready to gather information about the overall layout of the form.

The Prompts

1. PRINT OR SPOOL

Enter one of the following letters to specify the program's output method:

P = Write to the printer, if possible.

S = Spool output to disk.

O = Prompt the operator to select Print or Spool.

The default response is P(rint). If you specify P and the printer is not available, the program sends a message to the operator asking whether to retry printing or to direct output to a spool file.

2. SPOOL FILE NAME

Enter the name of the disk file to which the program will direct output when spooling. If you specify a filename less than nine characters long, the program appends the operator's terminal number to the filename.

3. START OF DETAIL LINE #

Enter the line number where you want the form's first detail line to appear. Note that this position is not affected by top-of-form lines. You may enter an integer from 0 through 99. The default value is 0.

4. NO. OF DETAIL LINES PER PAGE

Enter a number from 0 through 99 to indicate the maximum number of detail lines you will allow per page. The default number is 0.

5. START OF PAGE BOTTOM

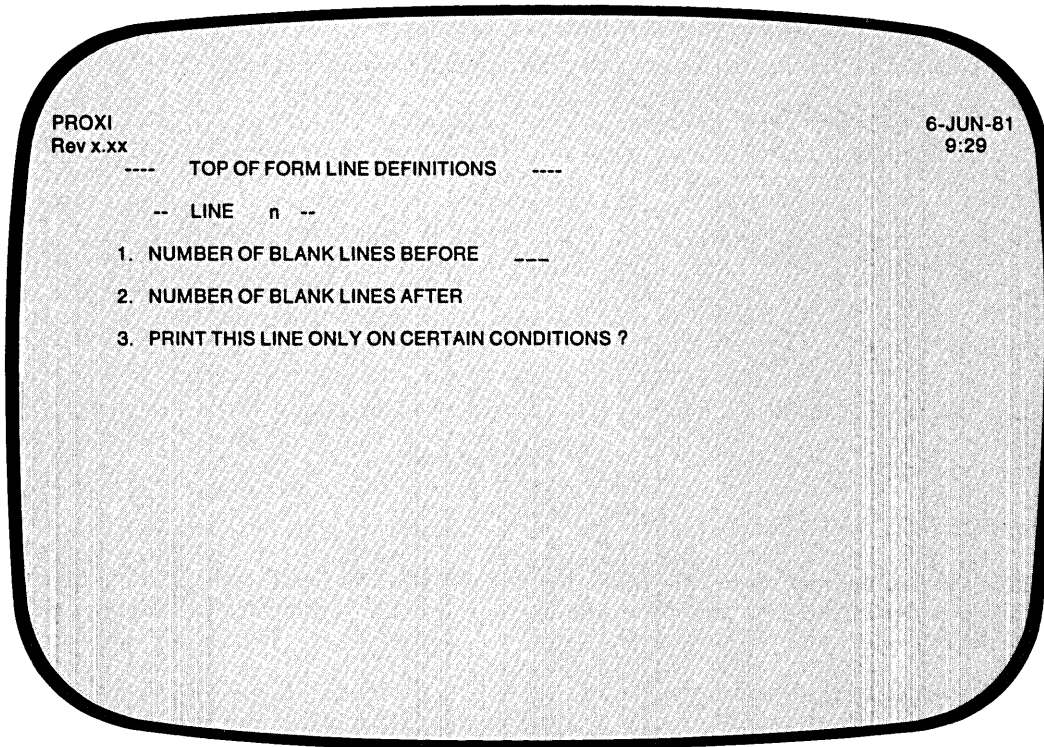
Enter the line number where you want the program to begin printing page break lines. The program generates page break lines when the number of detail lines exceeds the maximum you set for the page. Specify a number from 0 through 99. There is no default response.

What Next?

The Program Generator now presents the first in a series of data-entry screens with which you define and lay out the top-of-form lines (PG-28).

PG-28
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
BLANK LINES BEFORE	None	0 - 9, *, or the FINISHED, INSERT, or DELETE key
BLANK LINES AFTER	0	0 - 9 or *
CONDITIONS	N	Y or N

The Program Generator is ready to gather information about the lines that appear at the top of the form (i.e., the first page only). You may include up to nine top-of-form lines. Because the program has already read information for the first detail line, you may include this information in the top-of-form lines.

The Program Generator returns to this screen after you define each field within the current top-of-form line. Press FINISHED at the first prompt after you have defined all top-of-form lines.

The Prompts

1. NUMBER OF BLANK LINES BEFORE

Enter a digit (0 through 9) to specify the number of blank lines that will precede this top-of-form line. There is no default entry. Recall where you planned the first detail line to appear. Make sure that the top-of-form lines and the spacing between them will fit in the available area.

2. NUMBER OF BLANK LINES AFTER

Indicate the number of blank lines that will follow this top-of-form line. Instead of entering a number, you may enter an asterisk (*) to indicate that this line is a conditional field that belongs to the next print line.

3. PRINT THIS LINE ONLY ON CERTAIN CONDITIONS ?

Enter Y if you want this total line to appear only under certain conditions. If you entered an asterisk after the last prompt, you should respond Y here.

Enter N (the default) if you do not want conditional printing of this line.

What Next?

If you indicated that you want to provide conditional printing criteria, continue with the next screen description (PG-29).

If you do not want conditional printing for this line, move on to define each field within this line (PG-30).

If you press FINISHED at the first prompt to indicate that you've defined all top-of-form lines for this form, the Program Generator begins a series of screens about this form's detail lines (PG-31).

PG-29
Creating a Form Printing Program

Program Generator

```

PROXI                                     6-JUN-81
Rev x.xx                                  9:29
---- TOP OF FORM LINE DEFINITIONS ----
-- LINE  n --
1. NUMBER OF BLANK LINES BEFORE      x
2. NUMBER OF BLANK LINES AFTER       x
3. PRINT THIS LINE ONLY ON CERTAIN CONDITIONS ?  Y
CONDITION  n
    1. IF/AND/OR
    2. FIELD 1
    3. CONDITION
    4. FIELD 2
    
```

Quick Reference

Prompt	Default	Range
IF/AND/OR	None	IF, AND, OR, or the FINISHED, INSERT, or DELETE key
FIELD 1	None	A data-file field
CONDITION	EQ	EQ, NE, LT, LE, GT, or GE
FIELD 2	None	A field name or constant

This overlay screen allows you to enter up to 99 logical tests that will determine whether or not the program prints this top-of-form line. After you have entered and verified the final test of the series, press FINISHED. The program will print the line if the result of the tests is TRUE.

The Prompts

1. IF/AND/OR

Specify the appropriate logical operator for the indicated test. (The Program Generator supplies "IF" for the first test.) Press FINISHED at this prompt after you have entered all the logical tests.

2. FIELD 1

Enter the first field for this logical test. You may specify any field defined in the principal data file or a reference file. There is no default entry.

3. CONDITION

Enter one of the following conditional operators:

EQ (equal to)
NE (not equal to)
LT (less than)
LE (less than or equal to)
GT (greater than)
GE (greater than or equal to)

The default response is EQ.

4. FIELD 2

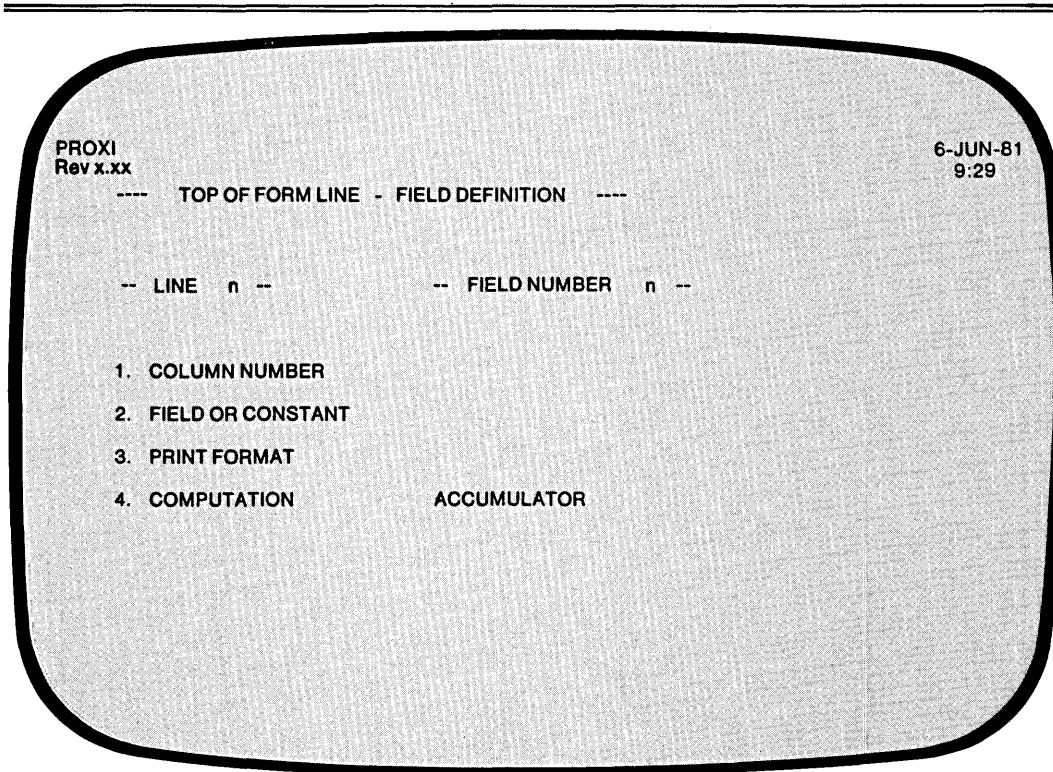
Enter a field name or literal value to be compared with FIELD 1. The field name may refer to a field in the principal data file, a reference file, or a Working Storage data name. (Be sure to enclose non-numeric literals in quotation marks.) There is no default.

What Next?

After you define all printing conditions for this top-of-form line, press FINISHED at the first prompt. The Program Generator then begins a series of screens to define each field within this top-of-form line (PG-30).

PG-30
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
COLUMN NUMBER	None	1 - 132, 999 or the FINISHED, INSERT, or DELETE key
FIELD OR CONSTANT	None	A data item or constant
PRINT FORMAT	(See description)	A PICTURE clause
COMPUTATION	No computation	(See Table 2-3)
ACCUMULATOR	None	A01 - A99

Table 2-3. Computation Symbols

Symbol	Computation
=	Axx = data item
+	Axx = Axx + data item
-	Axx = Axx - data item
*	Axx = Axx * data item
/	Axx = Axx / data item
%	Axx = 100 * Axx / data item

This data-entry screen appears after you define a top-of-form line. The Program Generator is ready to collect details about each field within the line. You will see the top-of-form line number and the field number displayed on the screen. If this line includes more than one print field, define each field beginning with the leftmost. You may enter Column 999 fields at any time, provided that the order is consistent with the operations they perform.

The Prompts

1. COLUMN NUMBER

Enter the column number (1-132) where the field begins (i.e., the column position of the field's first character). Enter 999 for a nonprinting field in which you perform a computation.

2. FIELD OR CONSTANT

Identify the data item or constant associated with this field. You may use a field item from the principal data file, a reference file, or Working Storage, or a literal. If you enter a non-numeric constant, be sure to enclose it with quotation marks.

3. PRINT FORMAT

Enter a PICTURE clause for this field. If this field displays a non-numeric literal, the default print format is an alphanumeric field of the same length. If the field reports a data item for which a Data Dictionary entry exists, the default print format is the data item's defined picture.

If you define an alphanumeric field, the Program Generator skips the next two prompts.

4. COMPUTATION

To perform a computation using this field's data item, enter the appropriate computation symbol (listed in Table 2-3). If you make no entry, no computation occurs.

ACCUMULATOR

Enter the name of the accumulator to be used for this computation. There are 99 accumulators (A01 through A99). Remember to prefix the number with the letter A. You must provide your own code for initializing and resetting these accumulators.

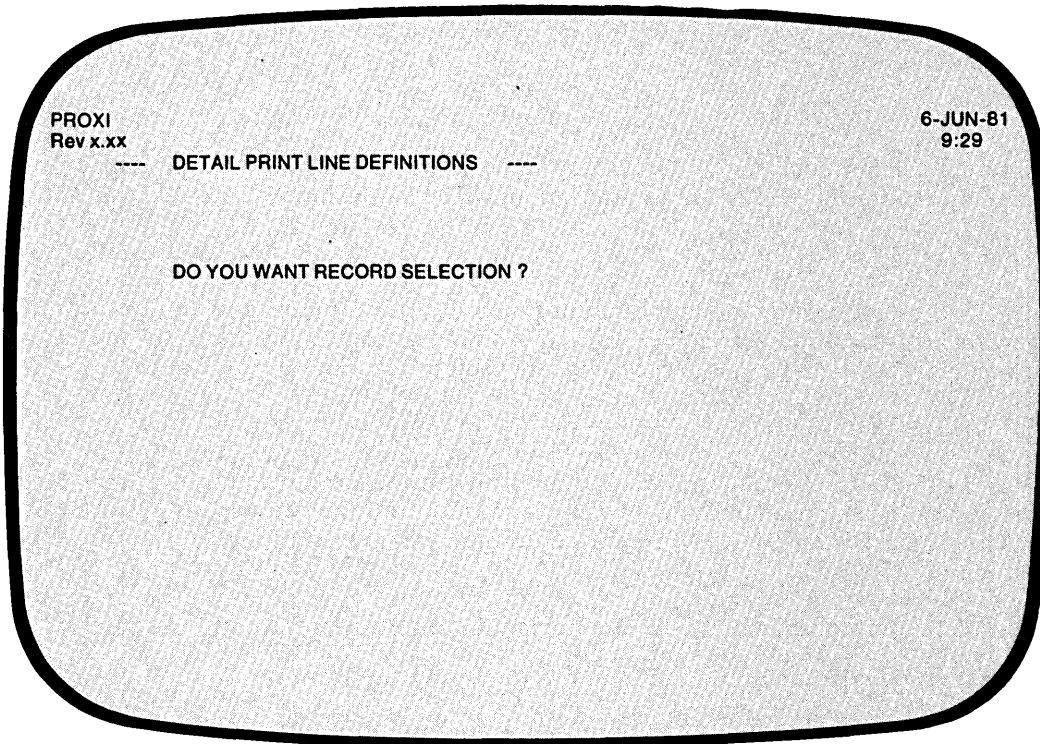
You need not specify computation for printing fields. To bypass these prompts, simply press NEW LINE at the COMPUTATION and ACCUMULATOR prompts. Only Column 999 fields require entries for these prompts.

What Next?

This screen repeats allowing you to define up to 99 fields within this top-of-form line. Press FINISHED after you have defined the last field. The Program Generator then allows you to define the next top-of-form line, if any (PG-28).

PG-31
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
RECORD SELECTION	N	Y or N

You encounter this screen after defining all top-of-form lines and their fields.

This is the first data-entry screen in a series that defines your form's detail lines. You indicate whether or not you will impose selection criteria for data-file records. You may specify up to 99 logical tests that will govern the printing of a group of one or more detail lines. After you define all the detail lines in a group, the Program Generator allows you to set up another set of criteria to govern another group of detail lines. (See PG-36).

You may also use record selection to selectively process data-file records by ignoring those records that do not pass any of the established criteria. (See PG-36).

The Prompt

DO YOU WANT RECORD SELECTION ?

Enter Y to establish a set of criteria that govern the printing of a group of detail lines. If you do not want to use record selection, answer N.

What Next?

If you chose record selection, continue with the next screen description (PG-32). Otherwise, move on to the detail line definition screen (PG-33).

PG-32
Creating a Form Printing Program

Program Generator

PROXI
Rev x.xx

6-JUN-81
9:29

---- DETAIL PRINT LINE DEFINITIONS ----

DO YOU WANT RECORD SELECTION ? Y

SELECTION n

1. IF/AND/OR
2. FIELD 1
3. CONDITION
4. FIELD 2

Quick Reference

Prompt	Default	Range
IF/AND/OR	None	IF, AND, OR, or the FINISHED, INSERT, or DELETE key
FIELD 1	None	A data-file field
CONDITION	EQ	EQ, NE, LT, LE, GT, or GE
FIELD 2	None	A field name or constant

This overlay screen allows you to provide record selection criteria for one or more detail lines. After you enter each logical test, the selection count increases by one and the fields clear so that you may enter the next logical test. You may specify up to 99 tests to be performed on the data-file records. To terminate the series of tests, press FINISHED at the first prompt.

The Prompts

1. IF/AND/OR

Specify the appropriate logical operator for the indicated test. (The Program Generator supplies "IF" for the first test.) After you have entered all the logical tests, press FINISHED.

2. FIELD 1

Enter the first field for this logical test. You may specify any field that is defined in the principal data file or in a reference file. There is no default entry.

3. CONDITION

Enter one of the following conditional operators:

EQ (equal to)
NE (not equal to)
LT (less than)
LE (less than or equal to)
GT (greater than)
GE (greater than or equal to)

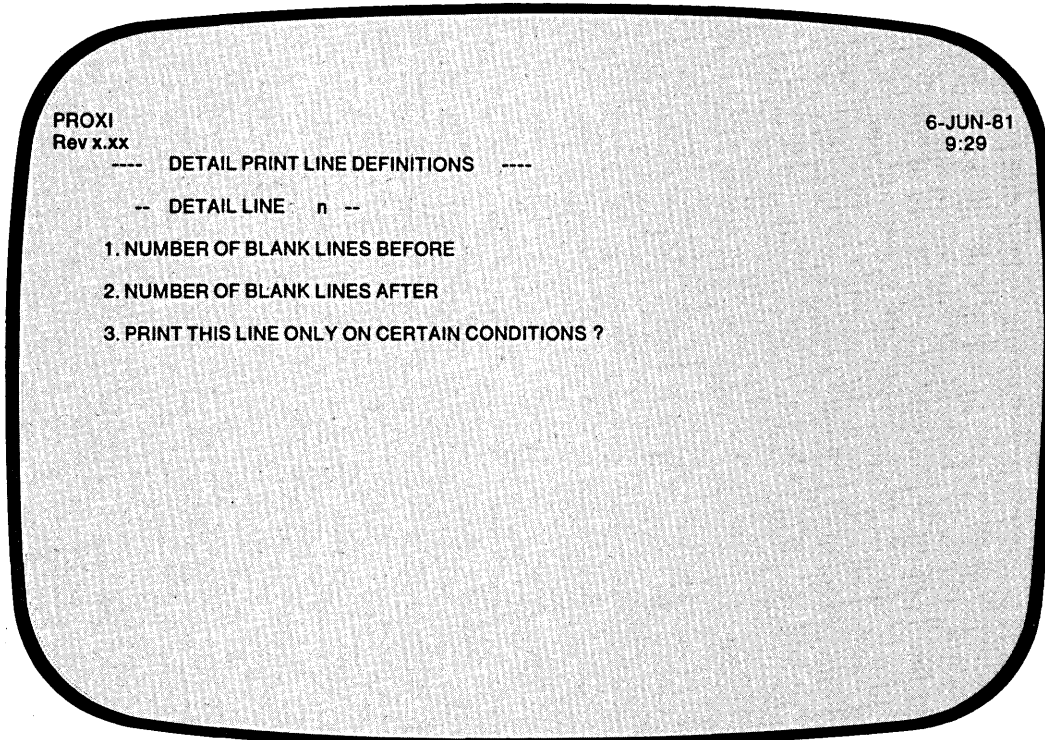
The default response is EQ.

4. FIELD 2

Enter a field name or literal value, to be compared with FIELD 1. The field name may refer to a field in either the principal data file, a reference file, or a Working Storage data name. (Be sure to enclose non-numeric literals in quotation marks.) There is no default.

What Next?

The next step is to define the detail lines associated with these selection criteria. Continue with the next screen (PG-33).



Quick Reference

Prompt	Default	Range
BLANK LINES BEFORE	None	0 - 9, P, T, or the FINISHED, INSERT, or DELETE key
BLANK LINES AFTER	0	0 - 9, P, T, or *
PRINT CONDITIONS	N	Y or N

The Program Generator is ready to define a specific detail line. This screen permits you to specify blank lines before and after the detail line. You may also impose printing conditions for this line.

After you complete the definition of a detail line, the Program Generator displays this screen again, allowing you to define another line. To end this cycle, press FINISHED at the first prompt.

The Prompts

1. NUMBER OF BLANK LINES BEFORE

Enter a digit (0 through 9), P, or T. The digit specifies the number of blank lines that will precede this detail line.

Enter the letter T to advance to the top of the next page before printing the detail line. Enter P if you want the program to advance to the top of the next page *and* to reset the page counter to one. There is no default entry. Press FINISHED at this prompt after you've defined all detail lines.

2. NUMBER OF BLANK LINES AFTER

Indicate the number of blank lines that the program will print following this detail line. You may also enter the letter T or P as we described in the previous prompt. The default response is 0.

You may also enter an asterisk (*) to indicate that this line is a conditional field that belongs to the next print line.

3. PRINT THIS LINE ONLY ON CERTAIN CONDITIONS ?

Enter Y if you want this detail line to appear only under certain conditions.

Enter N (the default) if you do not want conditional printing of this line.

What Next?

If you chose conditional printing for this detail line, continue with the next screen (PG-34).

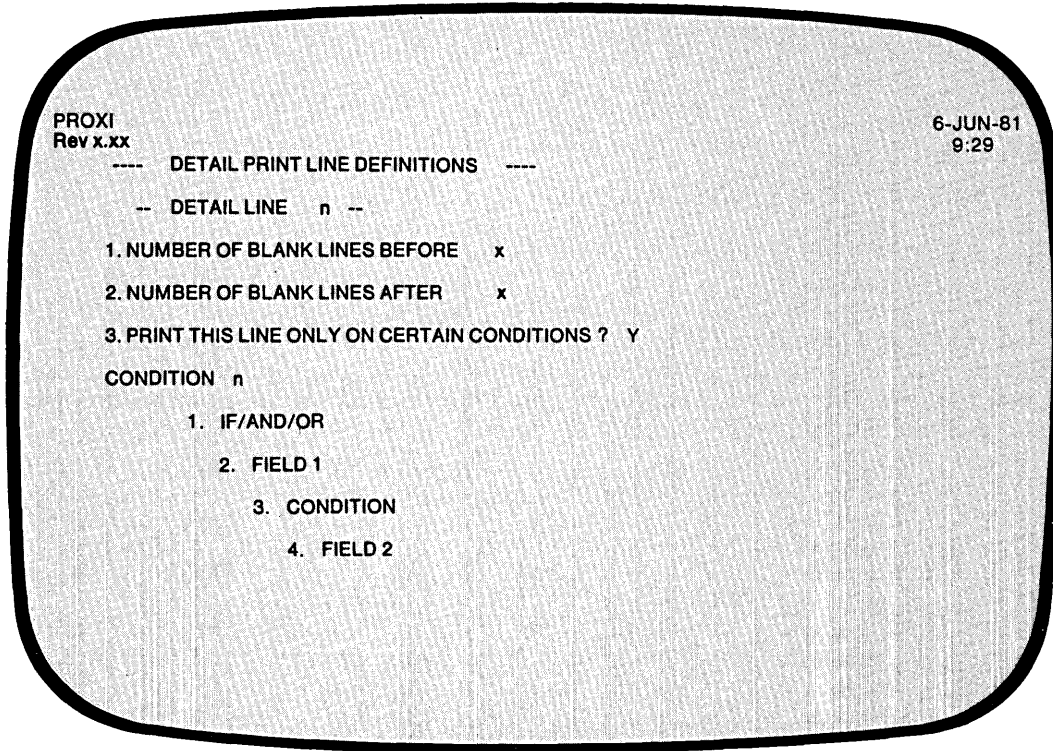
If you are defining a line without conditional testing, move on to the detail line field description screen (PG-35).

If you press FINISHED to indicate that you've defined all detail lines, and if the program does not use record selection criteria, then the Program Generator requests information about page break lines (PG-37).

If you press FINISHED to indicate that you've defined all detail lines governed by the current set of selection criteria, the Program Generator allows you to define another set of selection criteria for another group of detail lines (PG-36).

PG-34
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
IF/AND/OR	None	IF, AND, OR, or the FINISHED, INSERT, or DELETE key
FIELD 1	None	A data-file field
CONDITION	EQ	EQ, NE, LT, LE, GT, or GE
FIELD 2	None	A field name or constant

This overlay allows you to define printing conditions for a detail line. You may enter up to 99 logical tests. Press FINISHED at the first prompt to terminate the test series.

The Prompts

1. IF/AND/OR

Specify the appropriate logical operator for the indicated test. Note that the Program Generator supplies "IF" for the first test. After you have entered all the logical tests, press FINISHED.

2. FIELD 1

Enter the first field for this logical test. You may specify any field defined in the principal data file or a reference file. There is no default entry.

3. CONDITION

Enter one of the following conditional operators:

EQ (equal to)
NE (not equal to)
LT (less than)
LE (less than or equal to)
GT (greater than)
GE (greater than or equal to)

The default response is EQ.

4. FIELD 2

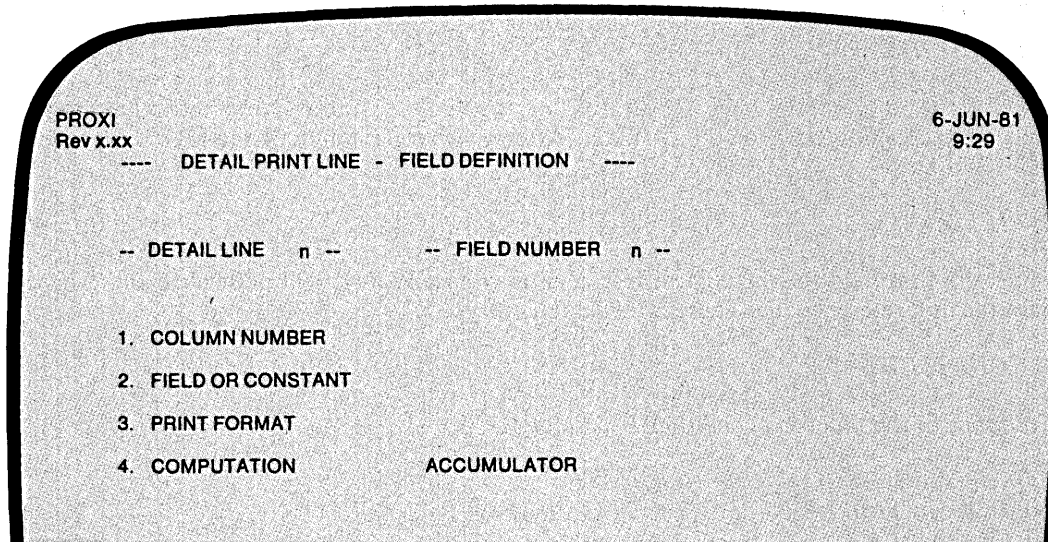
Enter a field name or literal value, to be compared with FIELD 1. The field name may refer to a field in either the principal data file, a reference file, or a Working Storage data name. (Be sure to enclose non-numeric literals in quotation marks.) There is no default.

What Next?

The Program Generator begins a series of screens with which you define each field in the detail line. Continue with the next screen description (PG-35).

PG-35
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
COLUMN NUMBER	None	1 - 132, 999, or the FINISHED, INSERT, or DELETE key
FIELD OR CONSTANT	None	A data item or constant
PRINT FORMAT	(See description)	A PICTURE clause
COMPUTATION	No computation	(See Table 2-4)
ACCUMULATOR	A01 - A99	none

Table 2-4. Computation Symbols

Symbol	Computation
=	Axx = data item
+	Axx = Axx + data item
-	Axx = Axx - data item
*	Axx = Axx * data item
/	Axx = Axx / data item
%	Axx = 100 * Axx / data item
T	Axx = automatic total + data item

The Program Generator is ready to gather specific layout information about this detail line. You may define up to 99 separate fields for this detail line.

If this line includes more than one print field, define them in order, beginning with the leftmost. You may define a Column 999 field at any time, provided that the order is consistent with the operation to be performed.

The Prompts

1. COLUMN NUMBER

Enter the column number where the field begins (i.e., the column position of the field's first character). You may also enter 999 for a nonprinting field that performs a computation.

2. FIELD OR CONSTANT

Identify the data item or constant associated with this field. The field item may appear in the principal data file or a reference file. If you enter a non-numeric constant, be sure to enclose it with quotation marks.

3. PRINT FORMAT

Enter a PICTURE clause for this field. If the field displays a non-numeric literal, the default print format is an alphanumeric field of the same length. If the field reports a data item for which a Data Dictionary entry exists, the default print format is the data item's defined picture.

If you intend to use automatic totaling, be sure to make the picture large enough to contain the resulting total; otherwise, you should use an accumulator to generate the total.

4. COMPUTATION

To perform a computation using this field's data item, enter the appropriate computation symbol (listed in Table 2-4). If you make no entry, no computation occurs.

ACCUMULATOR

Enter the name of the accumulator to be used for this computation. There are 99 accumulators (A01 through A99). Remember to prefix the number with the letter A. You must provide your own code for initializing and resetting these accumulators. If you are using automatic totaling, the Program Generator skips this prompt because the program uses a special set of totaling accumulators.

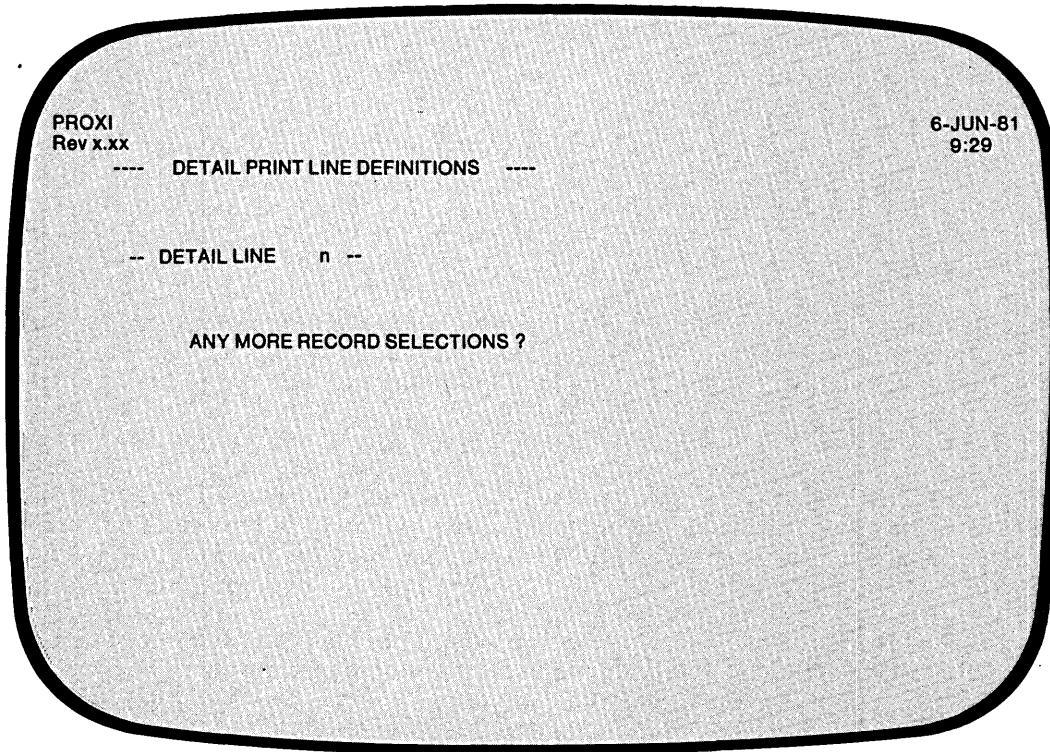
You need not specify computation for printing fields. To bypass these prompts, simply press NEW LINE at the COMPUTATION and ACCUMULATOR prompts. Only Column 999 fields require entries for these prompts.

What Next?

This screen repeats, allowing you to define up to 99 fields within this detail line. Press FINISHED after you've defined the last field. Then the Program Generator allows you to define the next detail line, if any (PG-33).

PG-36
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
RECORD SELECTIONS	N	Y or N

As we explained earlier (screen PG-31), you may define more than one set of record selection criteria.

The Prompt

ANY MORE RECORD SELECTIONS ?

Answer Y if you want to define another set of criteria to manage a group of detail lines.

Answer N if you do not want to define another set of selection criteria.

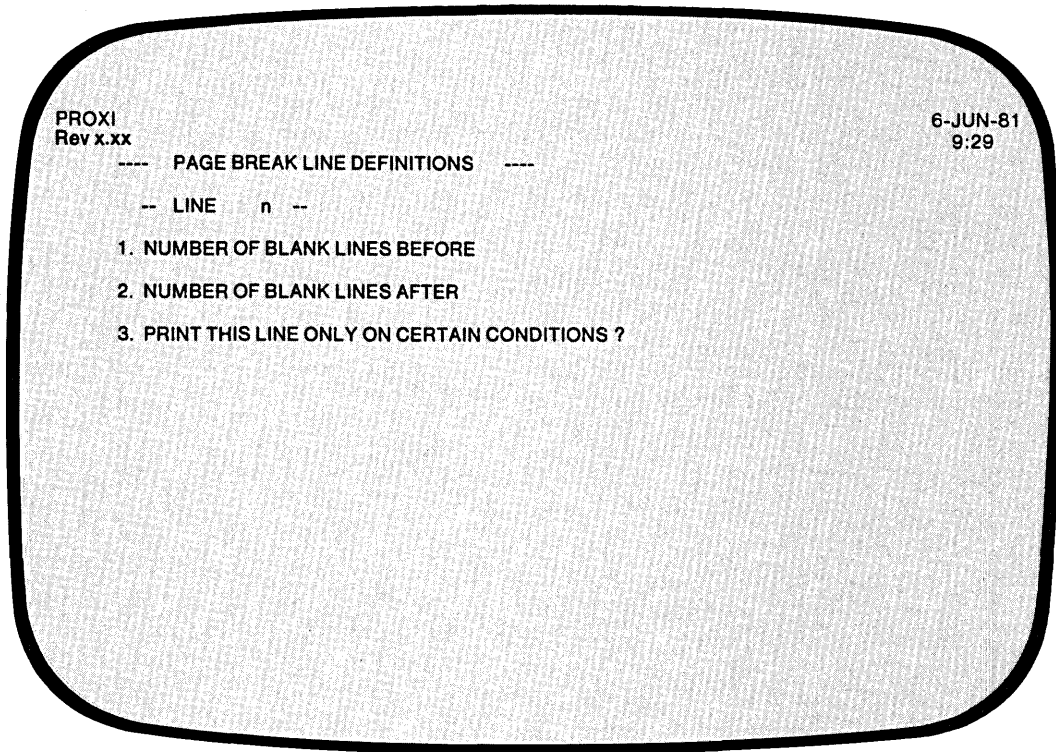
What Next?

If you chose to define another set of selection criteria, the Program Generator asks you to enter one or more logical tests (PG-32).

If you have completed definitions of all detail lines for this form, the Program Generator requests information about the page break lines (PG-37).

PG-37
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
BLANK LINES BEFORE	None	0 - 9, or the FINISHED, INSERT, or DELETE key
BLANK LINES AFTER	0	0 - 9, or *
PRINT CONDITIONS	N	Y or N

Page break lines appear at the bottom of a form page when the information (detail lines) continues onto a subsequent page. The Program Generator allows you to specify up to nine such lines. Press FINISHED at the first prompt after you have defined all page break lines.

A page break line may include information from the previous detail line's data-file records as well as any field subtotals to this point.

The Prompts

1. NUMBER OF BLANK LINES BEFORE

Enter a digit (0 through 9) to specify the number of blank lines that will precede this page break line. There is no default entry. Press FINISHED at this prompt after you've specified all page break lines.

2. NUMBER OF BLANK LINES AFTER

Indicate the number of blank lines to follow this page break line. You may also enter an asterisk (*) to indicate that this line is a conditional field belonging to the next line. The default number of blank lines is 0.

3. PRINT THIS LINE ONLY ON CERTAIN CONDITIONS ?

Enter Y if you want this page break line to appear only under certain conditions. You may specify up to 99 logical tests that determine whether or not this line is to be printed. If you answer Y, the Program Generator prompts you for the tests. Enter N (the default) if you do not want conditional printing of this line.

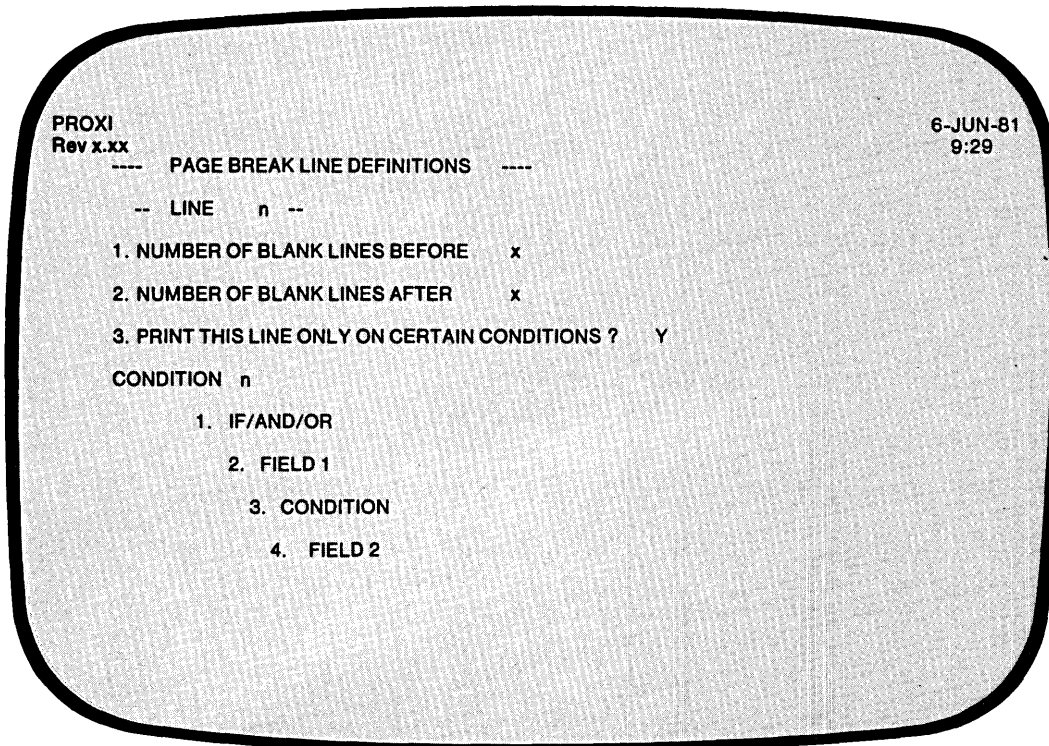
What Next?

If you press FINISHED at the first prompt (indicating that you have defined all page break lines), the Program Generator requests information about the control break field (PG-40).

If you request conditional printing for this page break line, continue with the next screen description (PG-38); otherwise, move on to the screen in which you define this line's fields (PG-39).

PG-38
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
IF/AND/OR	None	IF, AND, OR, or the FINISHED, INSERT, or DELETE key
FIELD 1	None	A data-file field
CONDITION	EQ	EQ, NE, LT, LE, GT, or GE
FIELD 2	None	A field name or constant

This overlay screen appears if you indicate that you want to provide conditional printing criteria for this page break line. It allows you to create a series of up to 99 logical tests.

The Prompts

1. IF/AND/OR

Specify the appropriate logical operator for the indicated test. Note that the Program Generator supplies "IF" for the first test. After you have entered all the logical tests, press FINISHED.

2. FIELD 1

Enter the first field for this logical test. You may specify any field defined in the principal data file or in a reference file. There is no default entry.

3. CONDITION

Enter one of the following conditional operators:

EQ (equal to)
NE (not equal to)
LT (less than)
LE (less than or equal to)
GT (greater than)
GE (greater than or equal to)

The default response is EQ.

4. FIELD 2

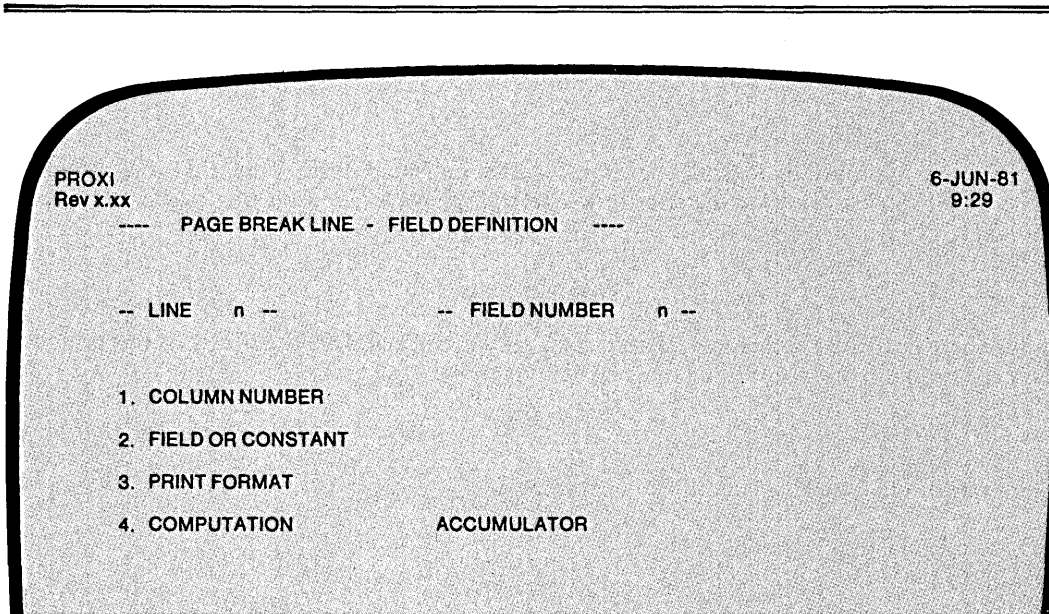
Enter a field name or literal value, to be compared with FIELD 1. A field name may identify a field in the principal data file, a reference file, or a Working Storage data name. (Be sure to enclose non-numeric literals in quotation marks.) There is no default.

What Next?

Press FINISHED at the first prompt after you have entered all the logical tests. The Program Generator then requests information about the fields within this page break line (PG-39).

PG-39
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
COLUMN NUMBER	None	1 - 132, 999, or the FINISHED, INSERT, or DELETE key
FIELD OR CONSTANT	None	A data item or constant
PRINT FORMAT	(See description)	A PICTURE clause
COMPUTATION	No computation	(See Table 2-5)
ACCUMULATOR	None	A01 - A99

Table 2-5. Computation Symbols

Symbol	Computation
=	Axx = data item
+	Axx = Axx + data item
-	Axx = Axx - data item
*	Axx = Axx * data item
/	Axx = Axx / data item
%	Axx = 100 * Axx / data item

The Program Generator is ready to gather specific layout information for this page break line. This allows you to define each field within the page break line. You may define up to 99 separate fields for each line.

If this line includes more than one print field, define the fields in order, beginning with the leftmost. You may enter a Column 999 field at any time provided that the order is consistent with the operation it performs.

The Prompts

1. COLUMN NUMBER

Enter the column position (1-132) of the field's first character. Enter 999 for a nonprinting field to be used for computation.

2. FIELD OR CONSTANT

Identify the data item or constant associated with this field. You may use a field item from the principal data file, a reference file, or Working Storage, or a literal. If you enter a non-numeric constant, be sure to enclose it with quotation marks.

3. PRINT FORMAT

Enter a PICTURE clause for this field. If this field displays a non-numeric literal, the default print format is an alphanumeric field of the same length. If the field reports a data item for which a Data Dictionary entry exists, the default print format is that data item's defined picture.

4. COMPUTATION

To perform a computation using this field's data item, enter the appropriate computation symbol (listed in Table 2-5). If you make no entry, no computation occurs.

ACCUMULATOR

Enter the name of the accumulator used for this computation. There are 99 accumulators (A01 through A99). Remember to prefix the number with the letter A. You must provide your own code for initializing and resetting the accumulator.

You need not specify computation for printing fields. To bypass these prompts, simply press NEW LINE at the COMPUTATION and ACCUMULATOR prompts. Only Column 999 fields require entries for these prompts.

What Next?

This screen allows you to describe each field within the specified page break line. Press FINISHED at the first prompt after you have defined each field in the current line. The Program Generator then allows you to define another page break line, if any (PG-37).

PG-40
Creating a Form Printing Program

Program Generator

PROXI
Rev x.xx

6-JUN-81
9:29

---- CONTROL BREAK DEFINITION ----

CONTROL BREAK FIELD

Quick Reference

Prompt	Default	Range
CONTROL BREAK FIELD	None	A field name

This screen appears after you define all page break lines and their fields.

The control break allows you to perform a single level of totaling. A change in the contents of the specified field causes the program to generate total lines (completing the form) and to begin printing the next form.

The Prompt

CONTROL BREAK FIELD

Enter the name of a field in the principal data file's record. There is no default response.

What Next?

The Program Generator presents a series of data-entry screens with which you define the total lines generated by the control break (PG-41).

If you do not define a control break, you move immediately to the COBOL program generation screen (PG-44).

PG-41
Creating a Form Printing Program

Program Generator

```

PROXI
Rev x.xx
6-JUN-81
9:29
---- TOTAL LINE DEFINITIONS ----
CONTROL BREAK  xxxxxxxxxxxx
-- TOTAL LINE  n --
1. NUMBER OF BLANK LINES BEFORE
2. NUMBER OF BLANK LINES AFTER
3. PRINT THIS LINE ONLY ON CERTAIN CONDITIONS ?

```

Quick Reference

Prompt	Default	Range
BLANK LINES BEFORE	None	0 - 9, or the FINISHED, INSERT, or DELETE key
BLANK LINES AFTER	0	0 - 9, or *
PRINT CONDITIONS	N	Y or N

The Program Generator is now ready to gather information about a total line.

The Prompts

1. NUMBER OF BLANK LINES BEFORE

Enter a digit (0 through 9) to specify the number of blank lines preceding this total line. There is no default.

Press FINISHED at this prompt after you've defined all total lines.

2. NUMBER OF BLANK LINES AFTER

Indicate the number of blank lines to follow this total line. You may also enter an asterisk to indicate that this line is a conditional field belonging to the next line. The default response is 0.

3. PRINT THIS LINE ONLY ON CERTAIN CONDITIONS ?

Enter Y if you want this total line to appear only under certain conditions. You may specify up to 99 logical tests that determine whether or not this line is to be printed.

Enter N (the default) if you do not want conditional printing for this line.

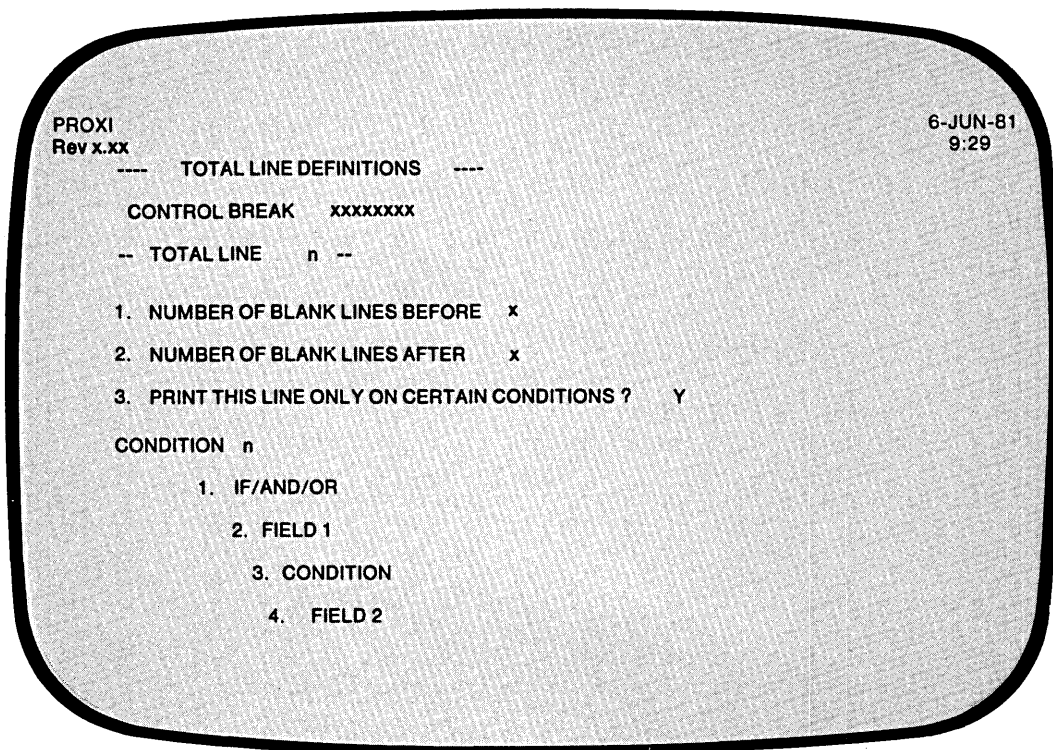
What Next?

If you press FINISHED at the first prompt to indicate that you've already defined all total lines for this form, the Program Generator screen appears (PG-44).

If you chose conditional printing for this line, continue with the next screen format (PG-42). Otherwise, move on to the screen with which you define this line's fields (PG-43).

PG-42
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
IF/AND/OR	None	IF, AND, OR, or the FINISHED, INSERT, or DELETE key
FIELD 1	Current field	A data-file field
CONDITION	EQ	EQ, NE, LT, LE, GT, or GE
FIELD 2	None	A field name or constant

This overlay screen allows you to define the printing conditions for this total line. As you enter each condition, the Program Generator increments the condition number and allows you to enter the next condition. You may specify up to 99 logical tests for this total line.

The Prompts

1. IF/AND/OR

Specify the appropriate logical operator for the indicated test. (The Program Generator supplies "IF" for the first test.) After you have entered all the logical tests, press FINISHED.

2. FIELD 1

Enter the first field for this logical test. The default field name is the current control break field (displayed in the upper part of the screen). Specify any field defined in the principal data file or in a reference file.

3. CONDITION

Enter one of the following conditional operators:

EQ (equal to)
NE (not equal to)
LT (less than)
LE (less than or equal to)
GT (greater than)
GE (greater than or equal to)

The default response is EQ.

4. FIELD 2

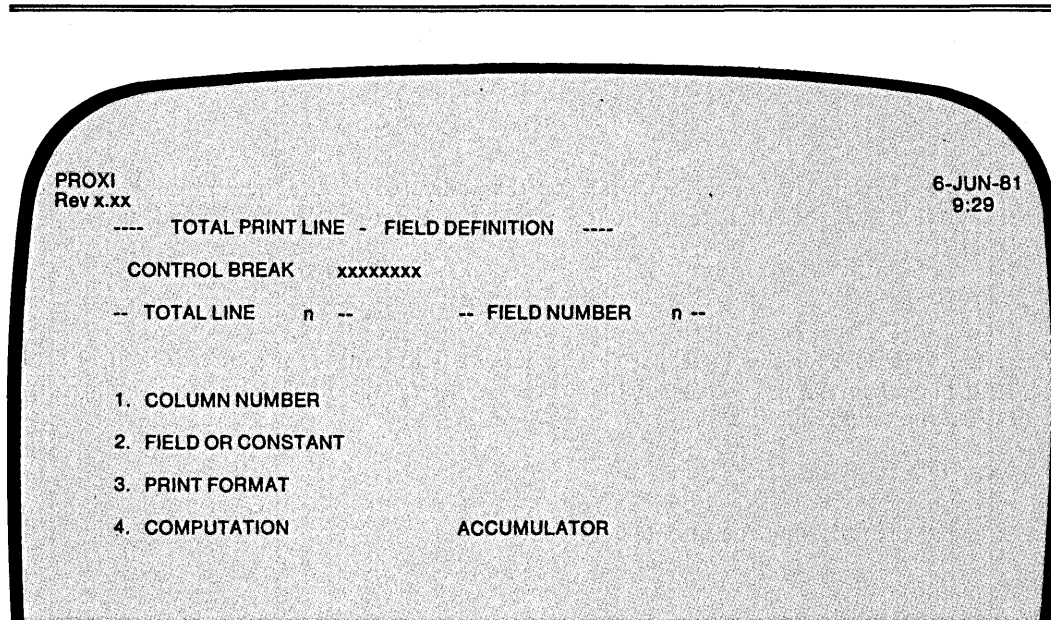
Enter a field name or literal value, to be compared with FIELD 1. The field name may refer to a field in the principal data file, a reference file, or a Working Storage data name. (Be sure to enclose non-numeric literals in quotation marks.) There is no default.

What Next?

After you have supplied all the printing criteria for this line, the Program Generator requests information about the field(s) in this line. Continue with the next screen (PG-43).

PG-43
Creating a Form Printing Program

Program Generator



Quick Reference

Prompt	Default	Range
COLUMN NUMBER	None	0 - 132, 999, or the FINISHED, INSERT, or DELETE key
FIELD OR CONSTANT	None	A data item or constant
PRINT FORMAT	(See description)	A PICTURE clause
COMPUTATION	No computation	(See Table 2-6)
ACCUMULATOR	None	A01 - A99

Table 2-6. Computation Symbols

Symbol	Computation
=	Axx = data item
+	Axx = Axx + data item
-	Axx = Axx - data item
*	Axx = Axx * data item
/	Axx = Axx / data item
%	Axx = 100 * Axx / data item
T	Axx = automatic total + data item

The Program Generator is ready to gather specific layout information for this total line. You may define up to 99 separate fields for each line. If the line contains more than one field, define the fields in order beginning with the leftmost. You may define Column 999 fields at any time provided that the order is consistent with the operation performed.

The Prompts

1. COLUMN NUMBER

Enter the column position (1-132) of the field's first character. You may also enter 999 to specify a nonprinting field to use for computation.

2. FIELD OR CONSTANT

Identify the data item or constant associated with this field. The field item may appear in the principal data file or in a reference file. If you enter a non-numeric constant, be sure to enclose it with quotation marks.

3. PRINT FORMAT

Enter a PICTURE clause for this field. If the field displays a non-numeric literal, the default print format is an alphanumeric field of the same length. If the field reports a data item for which a Data Dictionary entry exists, the default print format is the data item's defined picture.

4. COMPUTATION

To perform a computation using the field's data item, enter the appropriate computation symbol (listed in Table 2-6). If you make no entry, no computation occurs.

ACCUMULATOR

Enter the name of the accumulator to be used for this computation. There are 99 accumulators (A01 through A99). Remember to prefix the number with the letter A. You must provide your own code for initializing and resetting these accumulators. If you request automatic totaling, the Program Generator skips this prompt because the program uses a special set of totaling accumulators.

You need not specify computation for printing fields. To bypass these prompts, simply press NEW LINE at the COMPUTATION and ACCUMULATOR prompts. Only Column 999 fields require entries for these prompts.

What Next?

This screen repeats, allowing you to define more than one field in the specified total line. Press FINISHED at the first prompt to end the series; you then return to the total line definition screen (PG-41).

PG-44
Creating a Form Printing Program

Program Generator

PROXI
Rev x.xx

6-JUN-81
9:29

--- COBOL PROGRAM GENERATION ---

1. PROGRAM NAME xxxxxxxx
2. GENERATE THE COBOL CODE ?
3. OUTPUT FORMAT

Quick Reference

Prompt	Default	Range
GENERATE COBOL CODE	Y	Y or N
OUTPUT FORMAT	1	1 or 2

This screen appears after you've defined all total lines and detail lines for the Form Printing Program. The program name you supplied earlier appears after the first prompt.

The Prompts

2. GENERATE THE COBOL CODE

Enter Y to generate the code for your Form Printing Program. You should perform this operation only after you have completed all other PROXI operations for this program (such as defining data files).

Enter N if you want to skip the code generation step and return to the Program Generator menu immediately.

3. OUTPUT FORMAT

Enter 1 for card format (code with line numbers), or 2 for CRT format (code without line numbers). The default response is 1 (card format).

What Next?

If you choose to generate COBOL code, you receive the message

LOADING PARAMETERS FOR program - PLEASE WAIT

As the Program Generator builds each section of source code, it displays the appropriate division and section title on the screen. You may proceed to compile the code after you end the PROXI session.

You return to the Program Generator menu (PG-1) after this screen (whether or not you generated code).



Changing an Existing Program

You've selected item #2 from the Program Generator menu.

You may modify an existing PROXI program to update or correct it, or to build a new program based on the old one. When you select this function of the Program Generator you are actually building a new parameter file. When you generate COBOL code using this parameter file, the PROXI system creates a new set of compilable source code. Note that if the new program uses different screen formats or data files than did the original, you must use the PROXI Screen Generator and File Definer to build any related copy or parameter files before the new program will execute successfully.

The Sequence of Screens

```
PROXI                               6-JUN-81
Rev x.xx                             9:29
----- PROGRAM DEFINITION -----
1. PROGRAM NAME
2. PROGRAM TYPE
3. APPLICATION NAME
4. NEXT PROGRAM (NORMAL)
5. NEXT PROGRAM (ERROR)
```

The Program Generator presents the Program Definition screen (PG-2), then asks you to identify the program you want to change. After you provide the filename, the Program Generator looks for the parameter file (program.PP). If the file does not exist, you'll receive the error message

PROGRAM NOT FOUND . . . <CR> TO CONTINUE

When the Program Generator locates the parameter file, it displays the message

LOADING PARAMETERS FOR program . . . PLEASE WAIT

What happens next is that the Program Generator displays each screen format that you completed when you built the program parameter file. It begins with the Program Definition screen, displaying the existing entries for the remaining prompts following the program name.

As each screen is displayed, the ANY CHANGE question appears. You may elect to change the information displayed (by responding Y to the Any Change question), or pass on to the next screen after responding N (or pressing NEW LINE). Because the default response is N, you may move through a series of screen formats quickly if you have no changes to make.

If any of the changes imply that additional data-entry screens are necessary, the Program Generator will display the appropriate screen and allow you to enter information. For example, you want the new program to use reference files even though the original one didn't. You change the N following the ANY REFERENCE FILES question to a Y. Now the Program Generator will display the overlay screen in which you may name the reference files that the new program will use.

Adding and Deleting

When you define a series of similar program features, such as a list of field validation tests, record selection tests, print line conditions, or even a related set of detail lines, the PROXI program uses these program elements in the same sequence you entered them. When you build a new program you may want to change the series without redefining it completely. The PROXI function keys INSERT and DELETE will help you do this.

When you are working with serial entries such those as we described above, you may press INSERT at the Any Change question to insert a similar entry before the one displayed. The Program Generator will erase the entries on the screen. You then supply the information to be inserted and respond N to the Any Change question that follows. Your insertion is complete. The Program Generator then displays the next screen in the series (the same one displayed before the insertion). Remember to press the INSERT key *at* the Any Change question; you cannot use this key *within* the Any Change cycle.

The DELETE key (refer to the PROXI template) will remove the displayed serial item from the sequence. As with the INSERT key, you must press it at the Any Change question, not during the Any Change cycle. After deleting an item, you will see the next item in the series (if any) displayed.

Unless you've supplied the maximum number of entries permitted for the series (often 99), the Program Generator will prompt you for any additional entries you want to append to the current series. If you have nothing to append, just press FINISHED.

The SKIP IT Key

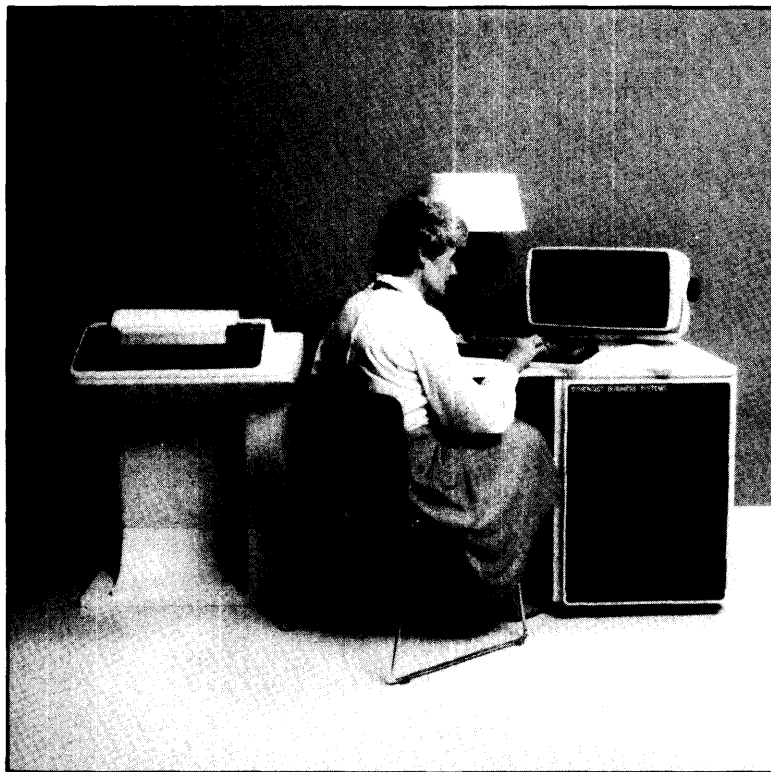
If you are changing a Report Writer or a Form Printing Program, you can use the SKIP IT key at the Any Change prompt. The SKIP IT key allows you to confirm all remaining screens (as if you responded N to all further Any Change questions) and proceed directly to the final screen, the COBOL Code Generation screen. This key saves you the time and trouble of confirming a long series of screens after you have made your final change.

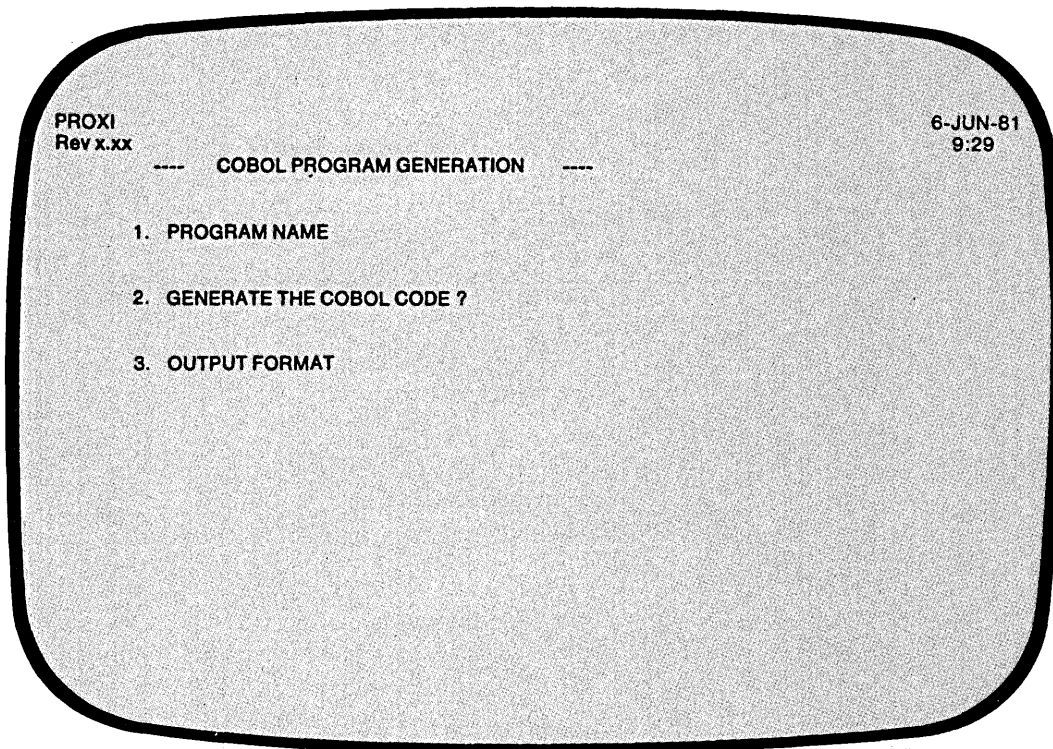
Reminders

You cannot use the Change function to convert one type of program to another, (for example, a File Maintenance Program into a File Inquiry Program). *Do not* try to change the program type displayed on the initial screen. Each type of program uses a different set of parameters. If you want to build a different type of program you must use the PROXI modules to create a new program.

Also, remember that when you change the program parameter file you may also need to build or modify other files. You may have to create or modify screen formats and their related copy files, or to supply additional sets of copy files to describe any new data files the program will use.

You may build a new program based on an existing program by calling the Change function and supplying the name of the existing program. During the Any Change cycle for the Program Definition screen, replace the old program name with the new name. The Program Generator will create a new program with the parameters you specify during the session. The existing program will remain as it is.





Quick Reference

Prompt	Default	Range
PROGRAM NAME	none	AOS filename
GENERATE COBOL CODE	Y	Y or N
OUTPUT FORMAT	1	1 or 2

You selected item #3 from the Program Generator menu, "Generate COBOL code." The Program Generator is ready to combine the various copy files you created with the main program framework to produce a complete source program ready for compilation.

IMPORTANT: You should perform this operation *after* completing all other PROXI functions related to this program.

The Prompts

1. PROGRAM NAME

Enter the name of the PROXI program you want to generate code for. The name you supply should be the name of a program parameter file (without the .PP extension) that you built earlier using the Program Generator.

2. GENERATE THE COBOL CODE ?

Enter Y if you want to generate the code for your PROXI program. *Note that Y is the default response.* If you've changed your mind and want to return to the Program Generator menu, enter N.

3. OUTPUT FORMAT

Enter 1 for card format (code with line numbers), or 2 for CRT format (code without line numbers).

What Next?

As the Program Generator builds each division and section of the COBOL source code, it displays the titles of these program units on the screen.

You return to the Program Generator menu (PG-1). If you generated code, you may proceed to compile it after you end the PROXI session. Be sure that you have completed all necessary operations before attempting to compile the code. If you neglect to create the required copy files, for example, the compiler will be unable to compile the source code.

End of Chapter

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

Chapter 3

The Screen Generator Module

This chapter describes the PROXI Screen Generator. The purpose of the Screen Generator is to help you build and incorporate screen formats into a File Maintenance or File Inquiry Program. Both programs use screen formats to communicate with the program operator. The Screen Generator performs the following functions:

- Defines a new or modifies an existing screen format.
- Prints a copy of a screen format and a description of its fields.
- Generates a COBOL screen definition copy file for use in the Screen Section of your program.
- Generates a COBOL screen procedure copy file for use in the Procedure Division of your program.
- Prints a copy of a screen definition or a screen procedure copy file.

The Screen Generator menu (SG-1) lists these functions and allows you to select the specific operation you want to perform.

The following pages provide background information that you will find helpful when using the Screen Generator. The screen descriptions in this chapter assume that you are familiar with the information that follows.

Using the Screen Generator

If you are creating a File Maintenance or File Inquiry Program, you will need to use custom-made screen formats to guide the operator's input and to display information. You must perform a series of steps to build and incorporate each screen format into your program:

- 1) Define the screen format (menu item #1 or #2). This operation produces a screen format parameter file. The file contains information about the screen image and its fields. The Screen Generator assigns the name *screen.SF* to this file (where *screen* is the name you've given this format). The second menu item allows you to alter the parameters in an existing .SF file.

- 2) Create a screen definition copy file for the program's Screen Section (menu item #4). The Screen Generator will build the copy file using information in the parameter (.SF) file. Therefore you must create the .SF file before attempting to build the copy file. If you alter the .SF file, be sure to recreate the screen definition copy file. The Screen Generator assigns the name *screen.SD* to this file (where *screen* is the name you've given this format).
- 3) Create a screen procedure copy file for the program's Procedure Division (menu item #5). The Screen Generator will build the copy file using information in the parameter (.SF) file. Therefore you must create the .SF file before attempting to build the copy file. If you alter the .SF file, be sure to recreate the Screen Procedure copy file. The Screen Generator assigns the name *screen.PL* to this file (where *screen* is the name you've given this format).

You must perform all these operations for each screen format *before* attempting to generate COBOL code through the Program Generator. (See Chapter 2.) The printing functions (menu items 3 and 6) are provided for your convenience; they do not contribute to the program.

Figure 3-1 diagrams the PROXI Screen Generator module, showing the sequence of data-entry screens for each menu path.

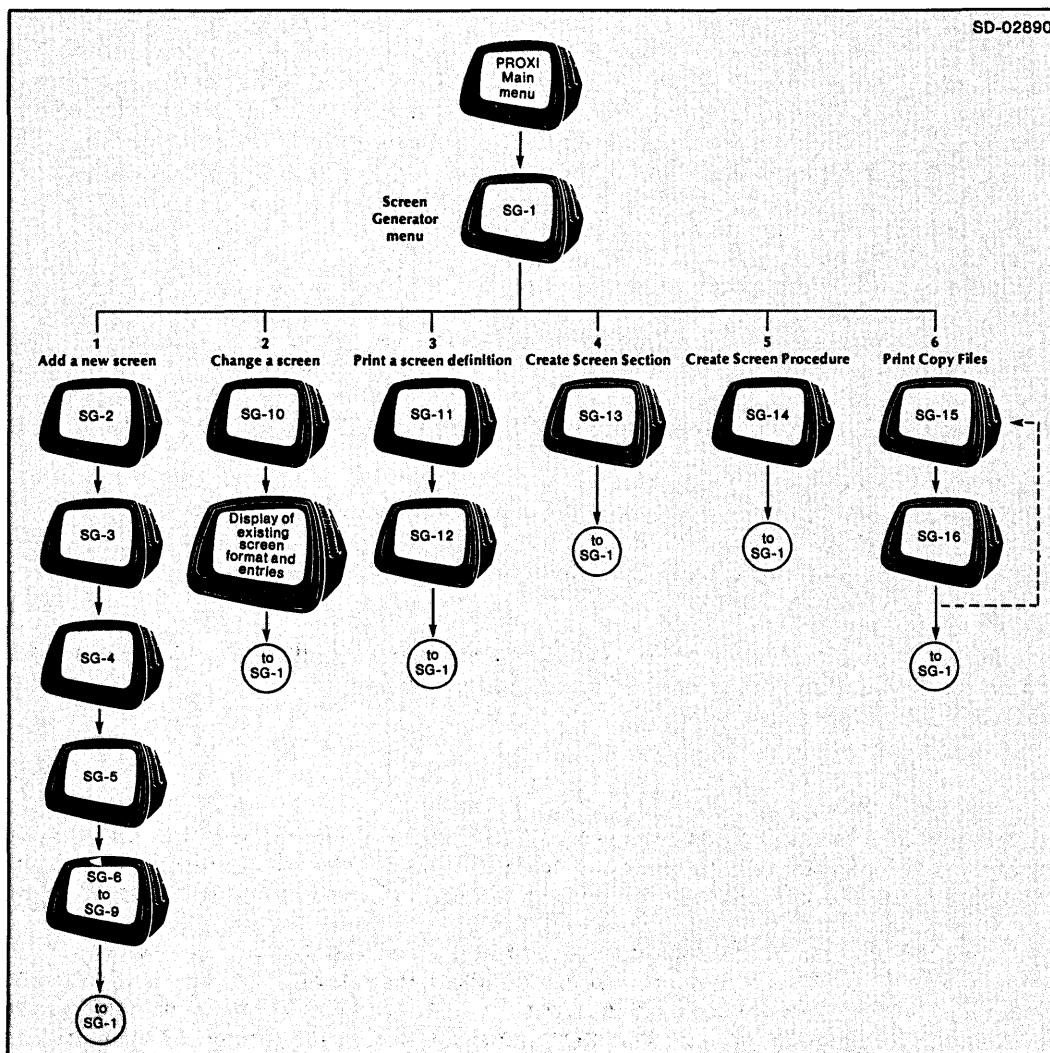


Figure 3-1. The Screen Generator Module Data-Entry Screens

The Screen Format

A screen format may contain literal (constant) fields and variable fields. Literal fields appear on the operator's screen exactly as you enter them into the format. They may include any printable character except the caret (^) and quotation mark (").

A string of one or more carets identifies a variable field. Figure 3-2 shows a sample screen format and points out the literal and variable fields.

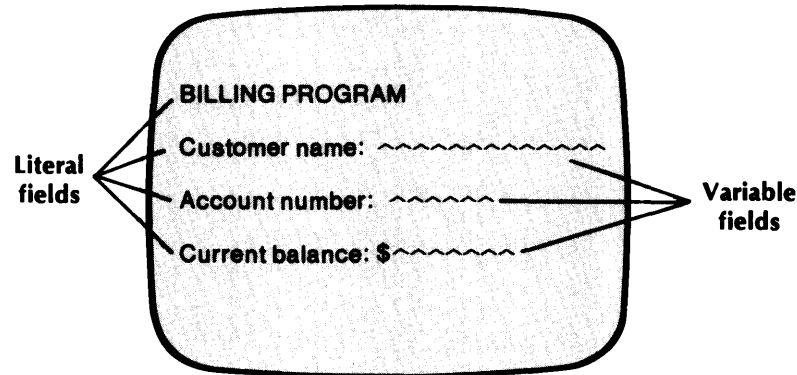


Figure 3-2. Literal and Variable Fields

You enter "Customer Name:" as a literal field. This string prompts the operator to make an entry. You use a string of carets to identify the field that will contain the operator's entry.

Variable Fields

There are three types of variable fields. The way you use the field determines its type.

- An *Entry* field displays or accepts an operator entry. The operator keys in a response to a prompt.
- A *Display* field displays a value resulting from a program operation on an entry field. For example, the operator may enter a code number which the program translates into a meaningful character string.
- A *Format* field appears as part of a constant field. For example, you may keep track of the number of transactions the operator enters. The constant field "Transaction No:" may be followed by a format field "n" showing the actual number.

Figure 3-3 illustrates a sample screen format which includes different types of variable fields.

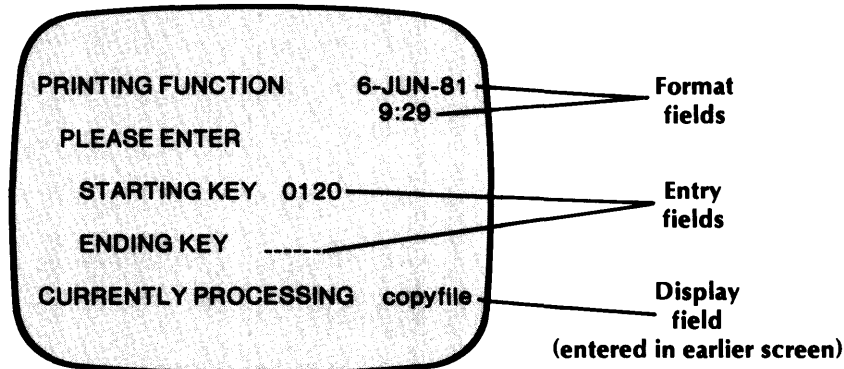


Figure 3-3. The Three Types of Variable Fields

Numbered Prompts

After the operator completes a screen format, the program automatically performs the Any Change cycle. To help the operator identify a particular entry field, you should use numbered prompts. In Figure 3-2 we showed a screen with two entry fields. Figure 3-4 illustrates how this screen format would appear if it included numbered prompts.

```
BILLING PROGRAM
1. Customer name: ~~~~~
2. Account number: ~~~~~
Current balance: $ ~~~~~
```

Figure 3-4. Screen Format Using Numbered Prompts

To enter a numbered prompt, begin the literal with a unique sequence number followed by a period. (Refer to Figure 3-4.) The number and its prompt must lie on the same line. The numbers must also be consecutive (i.e., *not* 5., 10., 15., etc.).

What if you want to assign the same prompt number to different fields? For example, you've entered the following prompt into the screen format:

1. Today's date: ^^ / ^^ / ^^

You expect the operator to enter the date in MM/DD/YY format. Each of these fields is distinct, however. The Screen Generator will associate the numbered prompt with *only* the first field, that is, the month. To allow the operator to modify the day and year during the Any Change cycle, you must associate these fields with the entry sequence number 1. You do this during the field definition phase.

When you define the month field, the Screen Generator automatically supplies the Entry Sequence Number 1 because you provided the numbered prompt. You must explicitly assign Entry Sequence Number 1 to the day and year fields when you define them. Then, since all three fields have the same entry sequence number, the operator can modify the entire date during the Any Change cycle.

Record Key Fields

The first screen for a File Maintenance or Inquiry Program *must* request the operator to enter one or more fields corresponding to the RECORD KEY. This requirement allows the program to locate the proper record for the display screen.

When building the screen format, identify the RECORD KEY fields by placing an asterisk (*) immediately before the numbered prompt. Figure 3-5 illustrates a numbered prompt for a RECORD KEY field.

BILLING PROGRAM

1. Customer name: ~~~~~

*2. Account number: ~~~~~

Current balance: \$ ~~~~~

Figure 3-5. Screen Format Showing RECORD KEY Field

Manipulating the Screen - Blanking and Overlays

For each screen format you build, the Screen Generator asks if you want the program to erase the previous screen before it displays the new screen. Normally you would want to do this. However, in some cases you may want one screen format to overlay another. You could set up the original screen as illustrated in Figure 3-6.

1. Customer name: ~~~~~

2. Address: ~~~~~

3. New account ? ^

Figure 3-6. Original Screen Format

If the operator answers No to the “New Account” question, the program displays a completely new screen format. But, if the answer is Yes, the overlay screen appears (without blanking the existing display). See Figure 3-7.

1. Customer name: Ian Young

2. Address: 34 Cameron Court
Woods Hole MA

3. New account ? Y

ASSIGNED #: 32S-88R

Type NEW LINE to continue:.....

} Overlay screen displaying program-assigned account number

Figure 3-7. Original Screen with Overlay Screen

To create the overlay effect you would request that the program *not* blank the screen before initiating the overlay screen format. Of course, it is up to you to design the screens so that they overlay correctly.

If you want to overwrite a character in the previous screen (assuming you don't erase the screen) you *must* place a character at the same position in the overlay screen - even if it is a space. When you design a screen format, underline characters appear on the current line wherever no other character exists. By making use of the overlay feature and carefully laying out your screens, you can perform some fancy screen manipulations.

Defining Fields

After you complete the design of a screen format, the Screen Generator begins a series of data-entry screens in which you define each variable field within the format. At the top of the data-entry screen you'll see information displayed about the field. Figure 3-8 illustrates an example.

The screenshot shows a screen with a light gray background and a black border. In the top left corner, the text reads "PROXI", "Rev x.xx", and "SCREEN FIELD DEFINITION". In the top right corner, the date and time "6-JUN-81" and "9:29" are displayed. Below this, a line of text reads "LINE 12 FIELD 1 START COL 43 LENGTH 16". At the bottom, the instruction "1. Enter customer's last name:" is followed by a wavy line representing a text input field.

Figure 3-8. Information About the Current Field

As you can see, the screen displays the line number, field number within the line, starting column, and length of the current field. Below this information appears the screen format line containing the current field. Be sure to check this information so that you know which field the Screen Generator is referring to - especially if the line contains more than one field.

The first information you provide is usually the field type (display, entry, or format) and the entry sequence number. If you used a numbered prompt for this field, the Screen Generator assigns the field Entry type, and uses the prompt number as the entry sequence number.

Depending on the type of field, the Screen Generator prompts you for specific information about the field and the way you want to process it.

For entry fields you specify

- The name of the field
- The type of data the field will contain
- The display format (PICTURE) for this field
- The change method that the operator can use
- A prompt line (optional) which will appear at the bottom of the screen
- Whether or not an entry is required in this field
- Whether or not the operator must fill the field
- Whether or not the field is secure (entry is not echoed)
- A default value for this field (optional)
- Whether or not you want to include validation tests to check the operator's entry
- Whether or not you want to include your own code to provide additional processing for this field value

For display and format fields you specify

- The name of the field
- The type of data it will contain
- The display format
- Whether or not you want the field to blink
- Whether or not you want the program to sound the bell when displaying the field

After you completely define one field, the Screen Generator repeats the series of definition screens for the next field, and continues until you've defined every field in the screen format.

Validity Tests

If you want validation tests performed on an entry field, you can specify from one to 99 logical tests that the program will use to check the operator's entry. If the entry value passes the validity checks (i.e., the result of the tests is true), the program will accept the operator's entry. If the entry value fails validity checking, the program erases the entry and prompts the operator to reenter a value.

Each validity test consists of four parts arranged in the following format:

$$\left. \begin{array}{l} \text{IF} \\ \text{AND} \\ \text{OR} \end{array} \right\} \text{field-name} \left. \begin{array}{l} \text{EQ} \\ \text{NE} \\ \text{LT} \\ \text{LE} \\ \text{GT} \\ \text{GE} \end{array} \right\} \left. \begin{array}{l} \text{field-name} \\ \text{literal} \end{array} \right\}$$

The first validation test always begins with IF. You'll receive prompts for the remaining three parts of the first test.

The program will apply the following hierarchy of evaluation when the validation checking consists of multiple tests:

1. Tests joined by AND
2. Tests joined by OR
3. Nested tests joined by IF

Using Your Own Code

If you intend to use your own code in a PROXI File Maintenance or File Inquiry Program, you must associate this code with a particular screen format. Each screen format allows you to specify Own Code to be performed before or after the program processes an operator entry. You simply provide the name of the code and the PROXI program will supply the appropriate COPY statement to include your code.

Generating Copy Files

After you define a screen format, the Screen Generator builds the parameter file screen.SF. Your File Maintenance or File Inquiry Program, however, requires source code copy files that define and manage the screen format. The Screen Generator provides two functions that create a Screen Section copy file and a Screen Procedure copy file.

If you make any changes in the screen format or its field definitions you should regenerate the corresponding copy files. If you make minor changes to a screen format you may need to rebuild only one of these copy files, but to be safe you should rebuild both. If you want specific information about the parameters used by each copy file, see Appendix E.

The Printing Functions

The Screen Generator provides a set of printing functions for your convenience; they do not contribute to the development of your program. When you request a printing function, the Screen Generator asks

OUTPUT TO THE PRINTER?

If you answer Y, the Screen Generator directs output to the line printer. If, however, the printer is not immediately available, you'll see the message:

PRINTER IN USE

Then the OUTPUT TO THE PRINTER question reappears. If you want to retry the printer, enter Y. If you don't want to use the printer, enter N. In this case the Screen Generator directs output to a disk file called PROXI\$nn (for a copy file) or PROXI\$SFnn (for a screen format file) where nn is your terminal number.

Changing a Screen Format

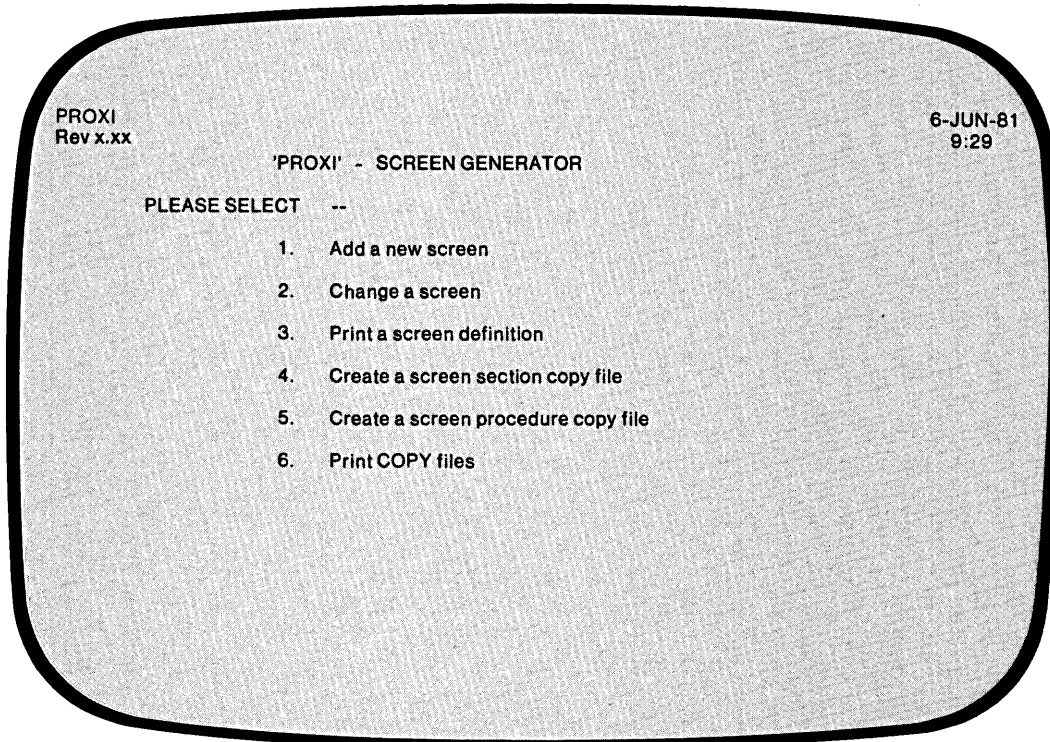
You may find it necessary to alter a screen format or the definition of one or more fields within a format. The Screen Generator allows you to make changes to any aspect of a screen format. For complete details about what you must do, refer to the description of the second item in the Screen Generator menu (SG-1), "Change a Screen."

The PROXI Function Keys

In the screen descriptions that follow we occasionally refer to the FINISHED, INSERT, and DELETE keys. You will not use the INSERT or DELETE keys unless you are modifying an existing screen format. We've mentioned them in this section because you may also refer to this section when changing a screen. If you are not changing a screen format, but creating a new one, you may use only the FINISHED key. Instructions for using this key are included in the appropriate screen descriptions.

SG-1
The Functions Menu

Screen Generator



This menu presents the functions of the Screen Generator module.

The Functions

1. ADD A NEW SCREEN

Select this function to create a new screen format. You paint the screen image and then provide information about the screen fields.

2. CHANGE A SCREEN

Select this function to modify an existing screen format. You can change your previous entries through the Any Change cycle. The Screen Generator also allows you to insert or delete parameters.

3. PRINT A SCREEN DEFINITION

Select this function to obtain a line printer listing of a screen format's parameters.

4. CREATE A SCREEN SECTION COPY FILE

Select this function to build the source code that defines a screen format in your program.

5. CREATE A SCREEN PROCEDURE COPY FILE

Select this function to build the source code that governs a screen format in your program.

6. PRINT COPY FILES

Select this function to obtain line printer listings of the source code copy files created for the Screen Section or Procedure Division of your program.

What Next?

Turn to the appropriate tab in this section for complete details about the Screen Generator function you selected.





SG-2

Adding a New Screen

Screen Generator

PROXI
Rev x.xx
SCREEN MAINTENANCE - ADD

6-JUN-81
9:29

SCREEN NAME

Quick Reference

Prompt	Default	Range
SCREEN NAME	None	An AOS filename

This screen appears after you select item 1 from the Screen Generator menu, "Add a New Screen."

The Prompt

SCREEN NAME

Enter the name of the screen format you are about to create. The Screen Generator places the parameter information into a file called *screen.SF*. where *screen* is the name you supplied. Do not include the .SF extension; the Screen Generator appends it.

If the file already exists, you receive the message:

SCREEN EXISTS <CR> *TO CONTINUE*

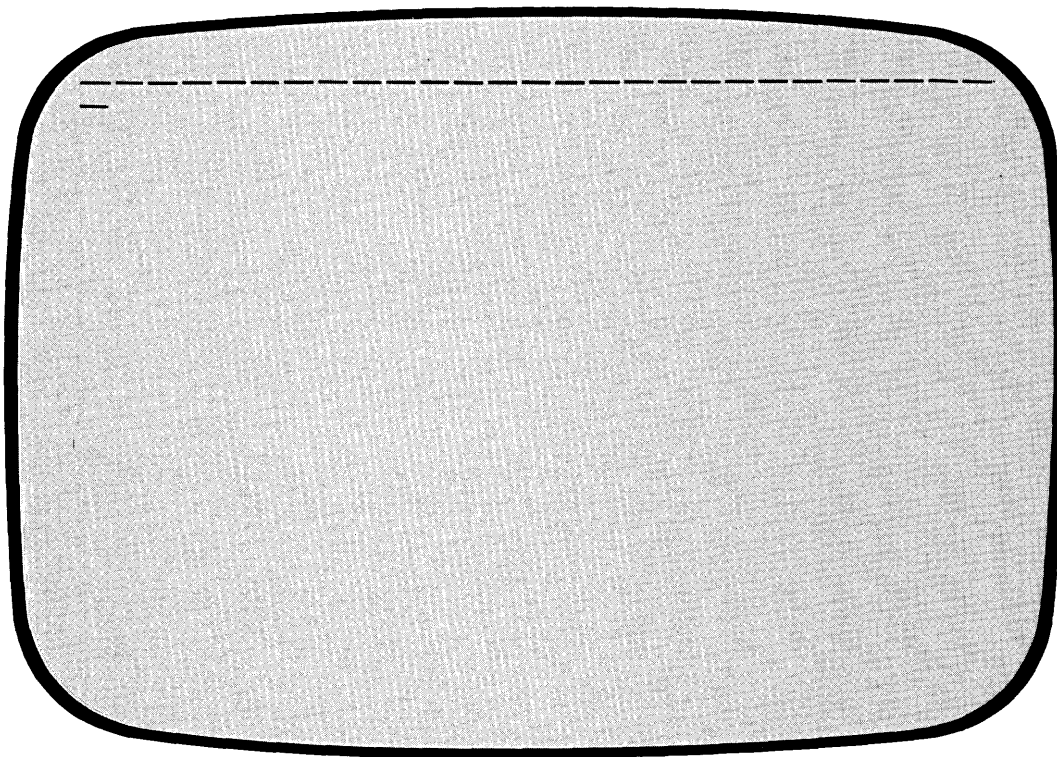
Press NEW LINE, then enter another name for the screen format file.

What Next?

After you name the screen, the Screen Generator allows you to create the screen format (SG-3).

SG-3
Adding a New Screen

Screen Generator



Quick Reference

- 1) Use carets to specify a variable field.
- 2) Follow numbered prompts with a period.
- 3) Use an asterisk before a numbered prompt to identify a key field.

The Screen Generator is ready for you to set up your screen format. The characters you enter from this point until you press FINISHED become part of the screen format. You have 23 lines to work with, and each has 80 characters. (Line 24 is reserved for prompt messages.)

To clear the screen at any point, press ESC. All entries you've made will disappear, leaving you with a blank screen once again. (If you are modifying a screen through the Any Change cycle and you press ESC, the screen reverts to the way it was before you modified it.)

Set up the screen exactly as you want it to appear to the operator. Enter literals (such as entry prompts) by typing the string in the appropriate location. *The string may not include a caret (^) or quotation mark (").* To specify variable fields (for entry, display, or format items), enter one or more contiguous carets to delimit the field. Each caret represents a character position for the variable field.

To enter a numbered prompt, precede the literal string with the appropriate entry sequence number and a period. For example,

3. Customer number:

You must identify RECORD KEY fields with an asterisk. Prefix the literal prompt as in this example:

*3. Customer number:

The next two pages list screen editing keys and cursor control keys that you may find helpful when designing a screen format.

You may include the operator's menu selection (for a File Maintenance Program) in a screen format. To do so you must define a format field of length 14. Link this field with the data item SELECTION-TYPE.

After you put the final touches on the screen format, press FINISHED. The Screen Generator executes any editing characters, displays the resulting screen, and initiates the Any Change cycle, allowing you to make additional changes.

What Next?

After you confirm the screen format, the Screen Generator presents the next data-entry screen (SG-4).

These two pages present tables listing screen editing and cursor control keys.

Table 3-1. Screen Editing Keys

Key	Action
→	Moves the cursor right one column.
←	Moves the cursor left one column.
↑	Moves the cursor to the beginning of the current input field.
ESC	Clears the screen to its former state. Newly created screens will be blanked. Modified screens will appear as they did after the last confirmation.
	Overwrites a character and shifts subsequent characters one column to the left.
\	Inserts a space and shifts subsequent characters one column to the right.
>	Inserts a blank line at the cursor and moves subsequent lines down one line. The last line disappears.
<	Deletes the line containing the cursor and moves subsequent lines up one line. The last line becomes blank.
NEW LINE	Moves the cursor to the beginning of the next line, or completes the screen format if on the last line.
CR	Blanks the characters to the right of the cursor and moves the cursor to the beginning of the next line. If the cursor is on the last line, it completes the screen format.
TAB	(Same as NEW LINE.)
ERASE PAGE	(Same as NEW LINE.)

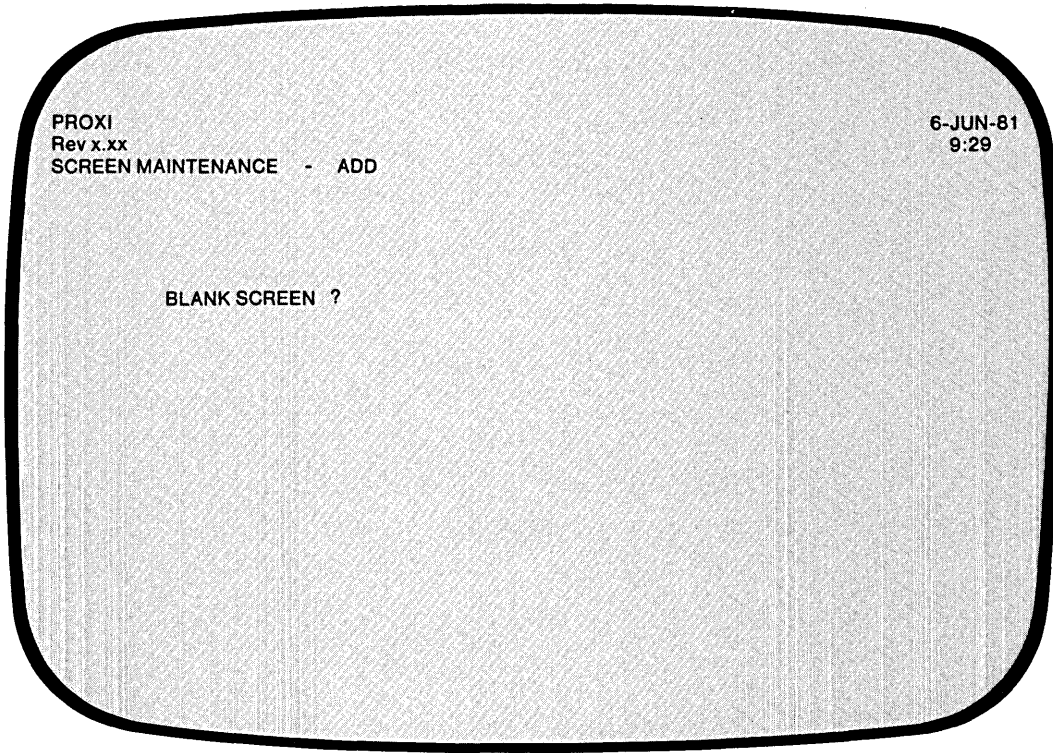
Table 3-2. Cursor Control Key Combinations

Key	Action
CTRL A	Moves the cursor to the end of the current line.
CTRL B	Moves the cursor to the end of the previous word in the current line.
CTRL E	Allows you to insert one character at the cursor position, moving all subsequent characters one place to the right. Note that you do not have to “close” the insertion with another CTRL E. Press CTRL E again <i>only</i> if you want to insert another character.
CTRL F	Moves the cursor forward to the beginning of the next word in the current line.
CTRL H	Moves the cursor to the beginning of the current line.
CTRL I	Moves the cursor to the beginning of the next line.
CTRL L	Moves the cursor to the beginning of the next line.
CTRL U	Erases the current line and positions the cursor at the beginning of the line.
CTRL X	Moves the cursor right one character position.
CTRL Y	Moves the cursor left one character position.

SG-4

Adding a New Screen

Screen Generator



Quick Reference

Prompt	Default	Range
BLANK SCREEN	Y	Y or N

This screen appears after you complete the screen format design.

Enter Y (the default answer) if you want the File Maintenance or File Inquiry Program to clear the operator's screen before displaying this screen format.

Enter N if you do not want to clear the screen. In this case the screen you've just described will overwrite the existing screen. That is, any character in the second screen will replace the corresponding character in the first. Note that you must explicitly enter blanks in the overwriting screen format if you want it to erase characters.

This feature allows you to create a "split screen" effect. You can retain information displayed at the top of a screen while a series of screen formats alter the bottom half.

What Next?

After responding to this question, you define each variable field in the screen format and provide any additional field processing information. The next data-entry screen you receive depends on the type of field you define.

Normally, the next screen you see is the one that follows in this manual (SG-5). This screen requests the field type and entry sequence number.

If, however, you entered a field with a numbered prompt, the Screen Generator already has the above information, so it skips to the screen that gathers specific information about an entry field (SG-6).

SG-5 Adding a New Screen

Screen Generator

```

PROXI
Rev x.xx
SCREEN FIELD DEFINITION
6-JUN-81
9:29

LINE nn FIELD nn START COL nn LENGTH nn
      xxxxx xxxxxxxxxxxx xxxxxx

SCREEN ID (Format, Display or Entry) ?      ENTRY SEQ # ?
  
```

Quick Reference

Prompt	Default	Range
SCREEN ID	E	F, D, or E
ENTRY SEQ #	None	1 - 99

This screen appears for each unnumbered variable field in the screen format. The screen displays the line number, the field number, the column where the field begins, and the field length. The line containing the specified field appears also.

The Prompts

SCREEN ID ?

Define this variable field as either Format, Display, or Entry.

Format describes a field that displays a value taken from a data file or a previous operator entry.

Display describes a field that displays the result of a processing step performed by the program. A display field is associated with a particular entry field.

Entry describes a field that accepts an operator's entry.

Enter F (format), D (display), or E (entry). Be sure to enter an uppercase letter.

ENTRY SEQ # ?

Specify the entry sequence number for the entry field. If you provided a numbered prompt, the number appears here. The sequence number determines the processing order for the screen's fields. It also allows the operator to identify a particular entry field during the Any Change cycle.

NOTE: If a numbered prompt applies to more than one entry field the Screen Generator recognizes the prompt for only the first of the fields. You must explicitly assign the same sequence number to subsequent fields.

For display fields, use the same entry sequence number as that of the associated entry field.

What Next?

The Screen Generator now requests additional information about the field. The prompts you receive depend on the type of field you assigned. If you specified an entry field, continue with the next screen description (SG-6).

If you defined this field as display or format, move on to screen description SG-7.

Remember, if you provided an entry sequence number for this field, the Screen Generator skips this screen (because it has the information already) and begins this field's definition with the next screen description (SG-6).

SG-6 Adding a New Screen

Screen Generator

PROXI 6-JUN-81
 Rev x.xx 9:29
SCREEN FIELD DEFINITION

LINE nn	FIELD nn	START COL nn	LENGTH nn
	xxx	xxxxxxxxx	xxxxx

SCREEN ID (Format, Display or Entry) ? E ENTRY SEQ # ? n

1. NAME OF FIELD	2. TYPE OF DATA
3. DISPLAY FORMAT	4. CHANGE METHOD
5. PROMPT LINE	
6. REQUIRED ?	7. FULL ?
8. SECURE ?	
9. DEFAULT VALUE	
10. VALIDATION ?	11. OWN CODE ?

Quick Reference

Prompt	Default	Range
NAME OF FIELD	None	A field name
TYPE OF DATA	X	X or 9
DISPLAY FORMAT	PIC X or PIC 9	A picture
CHANGE METHOD	R	R, E, or N
PROMPT LINE	Blank line	A field name or constant
REQUIRED	N	Y or N
FULL	N	Y or N
SECURE	N	Y or N
DEFAULT VALUE	None	A field name or constant
VALIDATION	N	Y or N
OWN CODE	N	Y or N

This overlay screen appears for an entry field.

The Prompts

1. NAME OF FIELD

Enter the COBOL data name that identifies the variable to contain the entered data.

2. TYPE OF DATA

Identify this field as either alphanumeric (X) or numeric (9). The default answer is X. The data type you specify here is for display purposes only. It does not have to match the field description in the FD or Working Storage.

3. DISPLAY FORMAT

Enter a PICTURE clause describing the format for this field. The program will use this picture to check the operator's input. The default format is PIC X or PIC 9 depending on the type of data you specified above.

4. CHANGE METHOD

You have three options: R (replacement), E (edit), and N (no change).

The *Replacement* method requires the operator to enter a value explicitly or to take the default (if one exists). This is the default change method.

The *Edit* method displays the current value for the field and allows the operator to alter any or all characters.

The *No change* method acts as does the replacement method except when the operator is performing the "Change/Inquire records" operation; in this case the operator will be prevented from making any change to the field.

5. PROMPT LINE

To assist the operator, you may display a character string literal or a data name value as an additional prompt message. Enter a literal (*enclosed by quotation marks*) or a data name. The prompt will appear on the bottom line of the screen (line 24) when the program is ready to accept operator input for this field. The program clears the prompt line after you complete the entry.

6. REQUIRED?

Enter Y if the operator *must* enter a value in this field. (You cannot provide a default value, therefore.) If you enter N (the default), the program will accept a null entry from the operator.

IMPORTANT: *DO NOT MAKE THE FIRST ENTRY FIELD IN A SCREEN A REQUIRED FIELD.* If you do, the program forces the operator to make an entry, thereby preventing use of the FINISHED key to conclude the operation.

7. FULL?

Enter Y if the operator *must* fill the entire field with characters (or take the default value provided). If you enter N (the default), the program will accept an incomplete field (trailing blanks).

8. *SECURE ?*

Enter Y if you want the program to echo operator input with asterisks. If you enter N (the default), operator input will appear on the screen as entered.

9. *DEFAULT VALUE*

Supply the value to be used by the program when the operator responds with a NEW LINE only. You cannot provide a default value for required fields. This is an optional entry. Be sure that this field's picture can handle the value. (The Screen Generator does not check this entry against the field's picture.)

10. *VALIDATION ?*

Enter Y if you want the program to perform validation tests on the entered value. If the operator's input does not pass the test, the program will clear the entry and wait for another response. You define the validation tests in another data-entry screen, which we describe later in this chapter. Refer to screen description SG-8 for details.

The default response is N. (You should make this response if you will not perform validity checking on the input.)

11. *OWN CODE ?*

Enter Y if you want the program to include source code which the program will execute before or after it processes this field. You identify this code in another data-entry screen, which we describe later in this chapter. Refer to screen description SG-9 for details.

The default response is N (which you should make if you do not intend to perform special processing before or after this field's entry).

What Next?

After you define this field, the Screen Generator requests information about the next field in the screen format. It will display the screen with which you define the field type (SG-5), unless you've implicitly identified the field as entry (by using a numbered prompt). In this case you'll see screen SG-6.

If this is the last variable field in the screen format, you'll return to the Screen Generator menu (SG-1).



SG-7
Adding a New Screen

Screen Generator

```

PROXI                                     6-JUN-81
Rev x.xx                                  9:29
SCREEN FIELD DEFINITION

LINE  nn  FIELD  nn  START COL  nn      LENGTH  nn
      xxx   xxxxxxx  xxxxxxx

SCREEN ID (Format, Display or Entry) ?  x      ENTRY SEQ # ?

1. NAME OF FIELD
2. TYPE OF DATA
3. DISPLAY FORMAT
4. BLINK ?
5. BELL ?
    
```

Quick Reference

Prompt	Default	Range
NAME OF FIELD	None	A field name
TYPE OF DATA	X	X or 9
DISPLAY FORMAT	PIC X or PIC 9	A picture
BLINK	N	Y or N
BELL	N	Y or N

This overlay screen appears for a display or format field.

The Prompts

1. NAME OF FIELD

Enter the COBOL data name that identifies the variable to contain the entered data.

2. TYPE OF DATA

Identify this field as either alphanumeric (X) or numeric (9). The default entry is X. The data type you specify here is for display purposes only; it does not have to match the field description in the FD or Working Storage.

3. DISPLAY FORMAT

Enter a PICTURE clause describing the format for this field. The program will use this picture to check the operator's input. The default format is PIC X or PIC 9 depending on the type of data you specified above.

4. BLINK ?

Enter Y if you want the program to blink this field on display. Enter N (the default) if you do not.

5. BELL ?

Enter Y if you want the program to sound the terminal bell when it displays this field. Enter N (the default) if you do not.

What Next?

After you define this field the Screen Generator requests information about the next field in the screen format. It will display the screen with which you define the field type (SG-5) unless you've implicitly identified the field as entry (by using a numbered prompt). In this case you'll see screen SG-6.

If this is the last variable field in the screen format, you'll return to the Screen Generator menu (SG-1).

SG-8

Adding a New Screen

Screen Generator

PROXI
Rev x.xx
SCREEN FIELD DEFINITION

6-JUN-81
9:29

LINE nn FIELD nn START COL nn LENGTH nn

xxxxxxxxx xxx xxxx

SCREEN ID (Format, Display or Entry) ? ENTRY SEQ # ?

FIELD VALIDATION PARAMETERS

VALIDATION TEST nn

1. IF/AND/OR
2. FIELD
3. CONDITION
4. OPERAND

Quick Reference

Prompt	Default	Range
IF/AND/OR	None	IF, AND, OR or the FINISHED, INSERT, or the DELETE key
FIELD	Current field	A data-file field
CONDITION	EQ	EQ, NE, LT, LE, GT, or GE
OPERAND	None	A field name or constant

This overlay screen appears when you have specified the use of validity tests on a data-entry field. The Screen Generator is now ready to accept from one to 99 logical tests, which it will perform on the value the operator enters for the indicated field. If the operator entry passes the logical test(s) you specify, the PROXI program will accept the entry. Otherwise the program will clear the field and await another operator response. (No error message appears.)

The Prompts

1. IF/AND/OR

Specify the appropriate logical operator for the indicated test. (The Screen Generator supplies "IF" for the first test.) After you have entered all the logical tests, press FINISHED.

2. FIELD

Enter the first field for this logical test. The default field name is that of the current field, (i.e., the field for which you are specifying validation tests). You may specify a data-file field, a Working Storage item, a constant, or a literal.

3. CONDITION

Enter one of the following conditional operators:

EQ (equal to)
NE (not equal to)
LT (less than)
LE (less than or equal to)
GT (greater than)
GE (greater than or equal to)

The default response is EQ.

4. OPERAND

Enter a field name or literal value to be compared with the first field. The field name may refer to a field in a data file, or to a Working Storage data name. (Be sure to enclose non-numeric literals in quotation marks.)

After providing each logical test, you receive the IF/AND/OR prompt once again. The sequence number of the next test appears. Enter the appropriate test or press FINISHED to complete the test entry cycle.

What Next?

If you indicated that you will supply your own code for this program, the Screen Generator asks you to identify the code (SG-9).

If this screen completes your definition of the field, the Screen Generator requests information about the next field in the screen format. It will display the screen with which you define the field type (SG-5), unless you've implicitly identified the field as entry (using a numbered prompt). In this case you'll see screen SG-6.

If you have just described the last screen field, you'll return to the Screen Generator menu (SG-1).

Adding a New Screen

```

PROXI
Rev x.xx
SCREEN FIELD DEFINITION
6-JUN-81
9:29

LINE nn FIELD nn START COL nn LENGTH nn
      xxx      xxxxxxxx      xxxxxxxx

SCREEN ID ( Format, Display or Entry) ? E      ENTRY SEQ # ? n
**  OWN CODE PARAMETERS      **

1. OWN CODE BEFORE SECT/PARA NAME
2. OWN CODE AFTER SECT/PARA NAME

```

Quick Reference

Prompt	Default	Range
OWN CODE BEFORE	None	A section or paragraph name
OWN CODE AFTER	None	A section or paragraph name

The Screen Generator presents this overlay screen when you've indicated that you want to include your own code to process this field entry.

The Screen Procedure copy file which executes this code will also check the Own Code status flag, OC-STATUS. If your code sets OC-STATUS to a nonzero value, the program will skip processing for the field (if set by Own Code before the field) or will repeat processing for the field (if set by Own Code after the field). If OC-STATUS is equal to zero, the program processes normally.

The file program.PL must contain all the Own Code that the program uses.

The Prompts

1. OWN CODE BEFORE SECT/PARA NAME

Enter the section or paragraph name of the COBOL source code that the PROXI File Maintenance Program will perform *before* it executes the standard code for this entry field.

2. OWN CODE AFTER SECT/PARA NAME

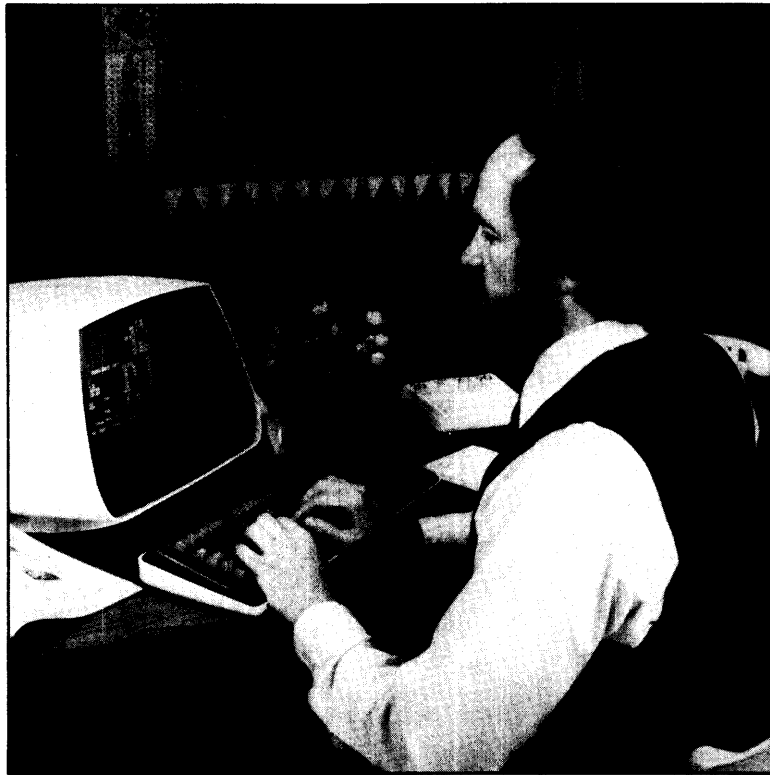
Enter the section or paragraph name of the COBOL source code that the PROXI File Maintenance Program will perform *after* it executes the standard code for this entry field.

What Next?

After you complete this screen, the Screen Generator requests information about the next field in the screen format. It displays the field definition screen (SG-5) unless you implicitly defined the field type as entry (by using a numbered prompt); in that case you move to screen SG-6.

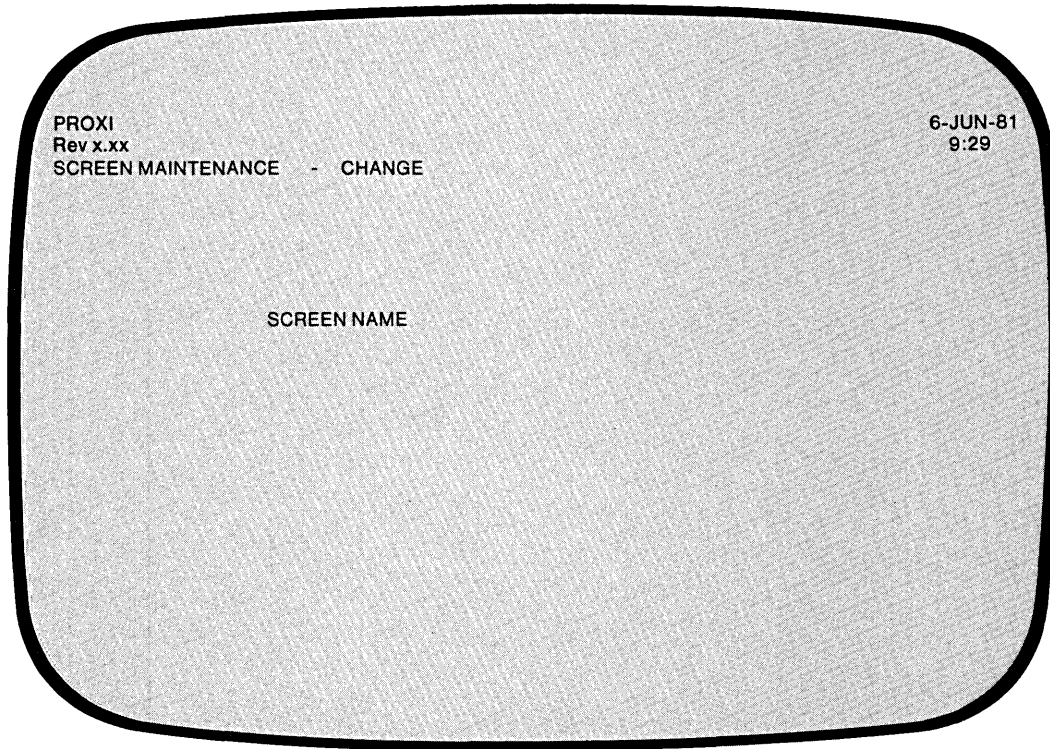
If the field just defined is the last variable field for this screen format, you return to the Screen Generator menu (SG-1).





SG-10 Changing a Screen

Screen Generator



Quick Reference

Prompt	Default	Range
SCREEN NAME	None	An AOS filename

You selected item #2 from the Screen Generator menu, "Change a Screen". The Screen Generator is ready for you to identify the existing screen format file that you want to modify.

The Prompt

SCREEN NAME

Enter the name of the screen format file. Do not include the .SF extension. (The Screen Generator assumes you will name a screen format parameter file.)

You may choose to create a new screen format file based on an existing file without changing the original. First, use the CLI COPY command to make a duplicate of the file. Then use the name of the new file when responding to this prompt.

When modifying an existing screen format file, one by one you'll see the data-entry screens that you completed when you created (or last modified) the screen format. That is, the Screen Generator displays the screen image, and the BLANK SCREEN question, and then it goes through the screen field definition data-entry screens including any validation test and Own Code parameter screens.

This section describes only what you may do when *changing* a screen format. Refer to the data-entry screen descriptions presented in the previous section, "Add a Screen", if you need specific information about a particular entry.

Modifying the Screen Format Image

The Screen Generator displays the existing screen format image and then prompts you with the Any Change question. If you do not want to alter the screen image, enter N (the default).

Enter Y if you want to redesign the screen image. You may use any editing or special keys as when creating the screen format. When you have completed the changes to the screen image, press FINISHED.

The BLANK SCREEN Question

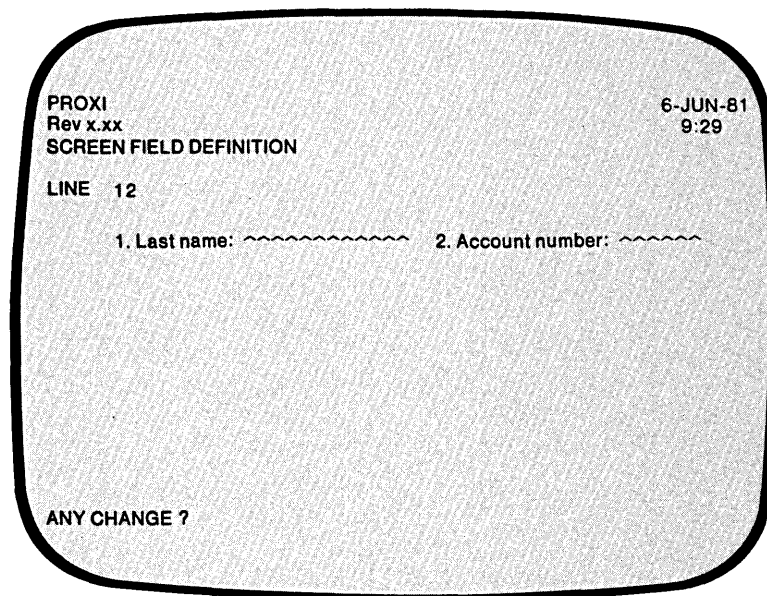
After modifying the screen image or bypassing that step, you will see the BLANK SCREEN question (SG-4), your previous response, and the Any Change question. Enter Y if you want to change your response; enter N (the default) if you do not.

Changing the Screen Field Descriptions

If you made modifications to the screen image, you'll be prompted to provide screen field definitions for the new and previously undefined fields. Perform the steps explained in the "Add a Screen" section as if you were defining the screen image for the first time.

For unchanged screen image fields, you'll see the field definitions you gave earlier along with the Any Change question. Respond N (the default) if the displayed parameters continue to be valid. Enter Y if you want to modify any of the entries.

The verification/modification process for each field happens in a number of steps. First the Screen Generator displays the line containing the field. See Figure 3-9.



The screenshot shows a terminal window with a light gray background and a black border. In the top left corner, it displays 'PROXI', 'Rev x.xx', and 'SCREEN FIELD DEFINITION'. In the top right corner, it shows the date '6-JUN-81' and time '9:29'. Below this, it says 'LINE 12'. The main content consists of two prompts: '1. Last name: ~~~~~' and '2. Account number: ~~~~~'. At the bottom left, it asks 'ANY CHANGE ?'.

Figure 3-9. Initial Step in Modifying a Field's Definition

If you want to change *any* aspect of this field, you must respond Y. Otherwise, the Screen Generator proceeds to display the next field.

If you answer Y, the first part of your definition appears, as shown in Figure 3-10.

```

PROXI                                     6-JUN-81
Rev x.xx                                 9:29
SCREEN FIELD DEFINITION

LINE 12  FIELD 1  START COL 22  LENGTH 12

    1. Last name: ~~~~~~  2. Account number: ~~~~~~

SCREEN ID (Format, Display or Entry) ?  E      ENTRY SEQ # ?  1

ANY CHANGE ?
  
```

Figure 3-10. Field Type and Entry Sequence Number

At this point you may verify the field type (entry, format, or display) or change it. For entry and format fields you may also modify the entry sequence number, if necessary. When you verify these values or complete your modifications, the Program Generator displays the remaining information about this field. Refer to Figure 3-11 which shows the format for an entry field.

```

PROXI                                     6-JUN-81
Rev x.xx                                 9:29
SCREEN FIELD DEFINITION

LINE 12  FIELD 1  START COL 22  LENGTH 12

    1. Last name: ~~~~~~  2. Account number: ~~~~~~

SCREEN ID (Format, Display or Entry) ?  E      ENTRY SEQ # ?  1

1. NAME OF FIELD  LAST-NAME                2. TYPE OF DATA  X
3. DISPLAY FORMAT
5. PROMPT LINE  " Enter the name as it appears on the order form "
6. REQUIRED ?  Y      7. FULL ?  N      8. SECURE ?  N
9. DEFAULT VALUE
10. VALIDATION ?  Y      11. OWN CODE ?  N

ANY CHANGE ?
  
```

Figure 3-11. Final Step in Changing a Field's Definition

If you want to modify any of the entries displayed you must respond Y to the Any Change question. As soon as you respond N, the Program Generator considers these entries confirmed and then moves on to the next field. Be careful when making your responses; if you accidentally press NEW LINE to the Any Change question, you cannot return to this screen to make further changes.

If you've changed the field type (from format to entry, for example) you may have to use the Any Change cycle on quite a few fields since there will be no existing values. It's a good idea to explicitly enter each value after specifying its item number. Avoid taking the default; one may not exist in this situation.

Changing the Validation Tests

The Screen Generator presents any validation tests you've defined. You may alter a test by responding Y to the Any Change prompt. Answer N (the default) if you want the test to remain as is. In addition to these two options, you may also insert one or more new validation tests, or to delete any existing test from the series.

To insert a validation test, press INSERT (function key #2) at the IF/AND/OR prompt. You may specify a validation test to be entered *before* the test currently displayed. You'll receive the familiar prompts for each part of the new logical test. The new test number appears above the prompts.

After you've modified or kept the final test, you'll receive prompts for additional tests, allowing you to append new tests to the current series. Press FINISHED if you do not want to add any tests.

To delete the currently displayed validation test, simply press the DELETE key (function key #3, *not* the DEL key!) at the IF/AND/OR prompt. The Screen Generator will remove this test from the series and display the next validation test, if one exists.

The Own Code Parameters

If you specified the use of your own code in the field definitions screen and did not change your answer, the Screen Generator displays your previous entries. You may change an entry by responding Y to the Any Change prompt. To complete your redefinition of this field, enter N. The Screen Generator will then go on to present the existing parameters for the next field in this screen format. After you have had the chance to redefine the entire screen format, you'll return to the Screen Generator menu (SG-1).

Screen Copy Files

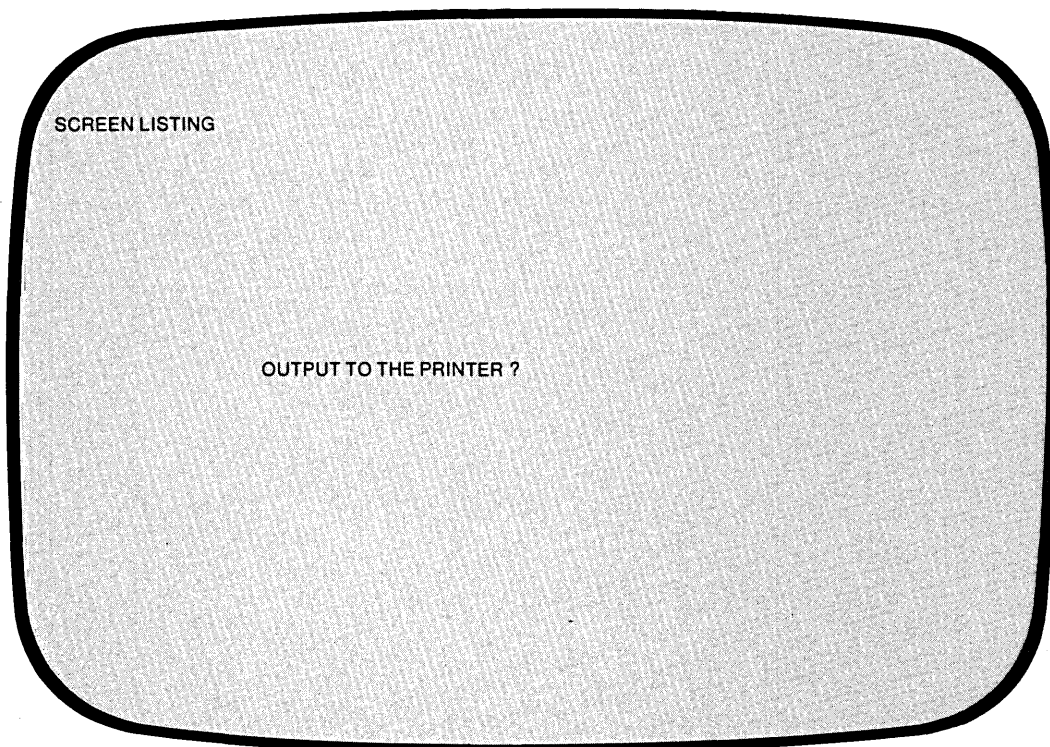
Remember that if you make any changes to the screen format parameter file (screen.SF), you may have to rebuild the corresponding Screen Section and Screen Procedure copy files. We suggest that you rebuild both these files to ensure that the copy files will reflect your changes.

Various parameters, however, apply to only one of the copy files. If your changes to the screen format are limited to certain entries, you may need to rebuild only one of these files. For details about which copy files use which parameters, see Appendix E.



SG-11
Printing a Screen Format

Screen Generator



Quick Reference

Prompt	Default	Range
OUTPUT TO THE PRINTER	Y	Y or N

You've selected item #3 from the Screen Generator menu. The Screen Generator is ready to print a copy of its specifications for a screen format you defined through the "Add a Screen" or "Change a Screen" procedures (menu items #1 or #2). The Screen Generator uses the contents of screen.SF to produce the listing.

The Prompt

OUTPUT TO THE PRINTER ?

If you answer Y, the screen format output goes to the line printer. If the printer is not available, you'll receive the message

PRINTER IN USE

and the "OUTPUT TO THE PRINTER?" question again.

If you respond N to the question, the screen format parameters go into the file PROXI\$SPnn, where nn is your terminal number. For example, if you are running the Screen Generator at @CON10, the output goes to the file PROXI\$SP10. If the file already exists, the new screen format overwrites any existing format.

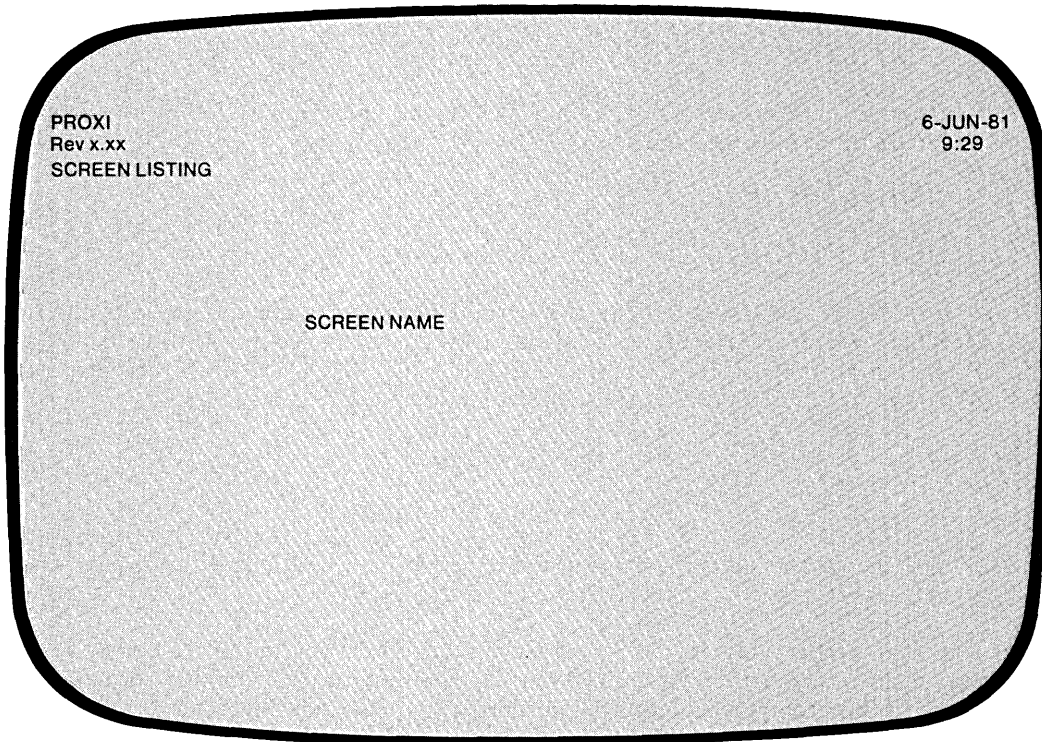
What Next?

The Screen Generator requests the name of the screen format. Continue with the next screen description (SG-12).

After you print a screen format, the Screen Generator returns you to this data-entry screen, allowing you to print another format. If you do not want to print another format, press FINISHED at this prompt: you'll return immediately to the Screen Generator menu (SG-1).

SG-12
Printing a Screen Format

Screen Generator



Quick Reference

Prompt	Default	Range
SCREEN NAME	None	An AOS filename

The Screen Generator now requests you to identify the screen format.

The Prompt

SCREEN NAME

Enter the name you assigned to the screen format: that is, the screen name you used in the "Add a Screen" or "Change a Screen" procedure. Do not include the .SF extension; it is understood.

You'll see the following message as the printing operation begins:

PRINTING . . . PLEASE WAIT

If the printer is not available, however, you'll see

PRINTER IN USE

OUTPUT TO THE PRINTER ?

Answer Y if you want to retry the printer. If you answer N, the Screen Generator will direct its output to a disk file called PROXI\$SPnn (where nn is your terminal number).

Figure 3-12 illustrates a sample screen parameter listing.

What Next?

After printing the specified screen format, the Screen Generator allows you to print another format. It immediately returns to the Screen Generator menu (SG-1).

SCREEN NAME: LISTSCR

SCREEN FORMAT

```

      1          2          3          4          5          6          7          8
1234567890123456789012345678901234567890123456789012345678901234567890
*****
1*
2*
3*
4*
5*
6*      *1. Listing Number:  ^^^^^^      2. Date:  ^^^^^^^^^
7*
8*
9*      3. Town:  ^^^^^^^^^^^^^^^
10*
11*
12*      4. Description:  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
13*
14*
15*      5. Original Price:  $^^^^^^^^
16*
17*
18*      6. Current Price:  $^^^^^^^^
19*
20*
21*
22*
23*
*****
1234567890123456789012345678901234567890123456789012345678901234567890
      1          2          3          4          5          6          7          8

```

Figure 3-12. Listing of Sample Screen Format Parameters (continues)

RUN: 24-NOV-80 8:39

PAGE 2

SCREEN NAME: LISTSCR

SCREEN FIELD DEFINITIONS

LINE	ENTRY	FIELD		DATA	DISPLAY		C	R	F	S						
COL #	TYPE	NAME		LEN	FORMAT		M	Q	L	Q						
6	30	1	K	LIST-NO	9	5	ZZZZ9				R	N	N	N		
				PROMPT:	"Check the Red Book before assigning a new number."											
6	56	2		LIST-DATE	X	8	X(8)				R	Y	Y	N		
				PROMPT:	"Use dd/mm/yy format."											
9	20	3		TOWN	X	12	X(12)				R	Y	N	N		
12	27	4		DESCRIPTION	X	30	X(30)				R	Y	N	N		
				PROMPT:	"Be sure to include the type of heating fuel."											
15	31	5		ORIGINAL-PRICE	9	7	ZZ9,999				R	N	N	N		
18	31	6		CURRENT-PRICE	9	7	ZZ9,999				R	N	N	N		
				PROMPT:	"Default value is the original price."											
				DEFAULT:	ORIGINAL-PRICE											

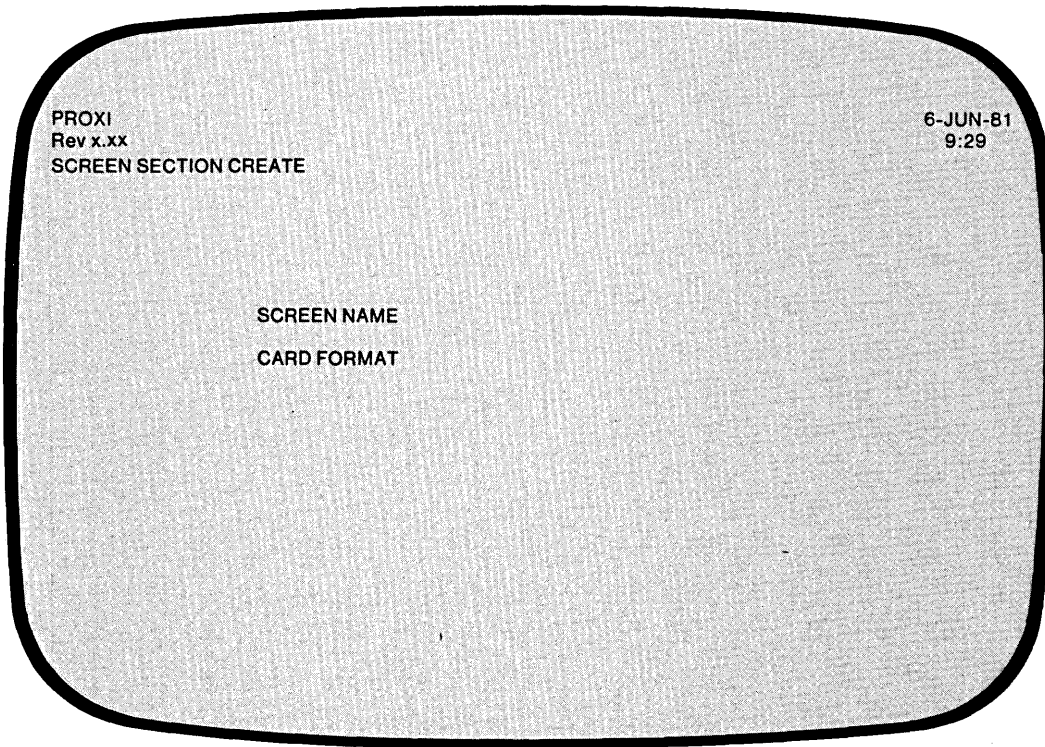
Figure 3-12. Listing of Sample Screen Format Parameters (concluded)





SG-13
Creating a Screen Section Copy File

Screen Generator



Quick Reference

Prompt	Default	Range
SCREEN NAME	None	An AOS filename
CARD FORMAT	Y	Y or N

You've selected item #4 from the Screen Generator menu, "Create a screen section copy file." The Screen Generator is ready to generate a source code copy file that will provide a definition of a screen format to the Screen Section of your program. The Screen Generator uses the screen definition parameter file to create this copy file.

If you have not yet created the parameter file through the "Add a Screen" or "Change a Screen" functions, press FINISHED. This will return you to the Screen Generator menu from which you can begin the screen definition procedures.)

The Prompts

SCREEN NAME

Enter the AOS filename (without the .SF extension) of the screen format parameter file that you want to generate a Screen Section copy file for.

CARD FORMAT

Enter Y if you want to produce a copy file with line numbers. If you want unnumbered code (CRT format), enter N. The default response is Y.

After you respond to the CARD FORMAT question, the Screen Generator begins to build a Screen Section copy file for the specified screen. You'll see the message

CREATING screen.SD

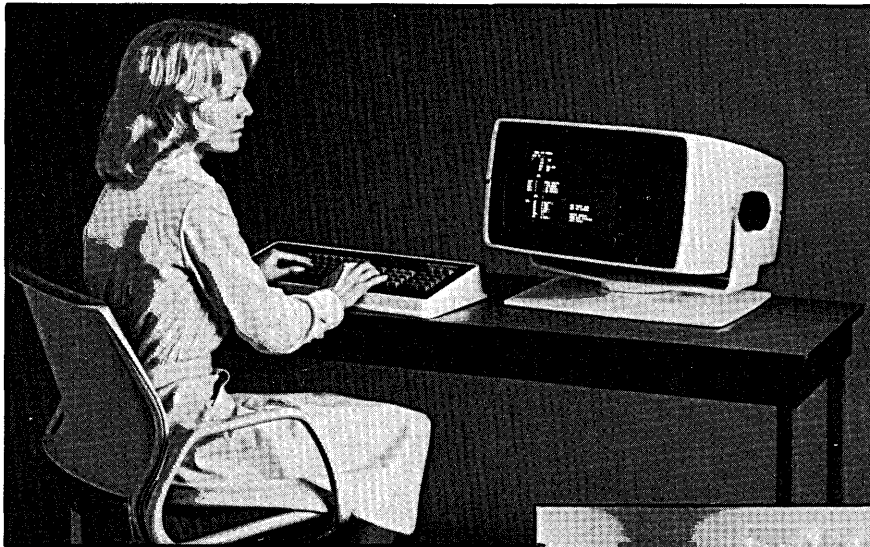
As the Screen Generator builds the file, statement line numbers will appear on the screen. (Only card format copy files will actually contain these statement numbers, however.)

If the file screen.SD exists, the new file overwrites it.

IMPORTANT: If you alter the parameter file after building the screen section copy file, you will probably have to rebuild the copy file. Refer to the "Change a Screen" section for details.

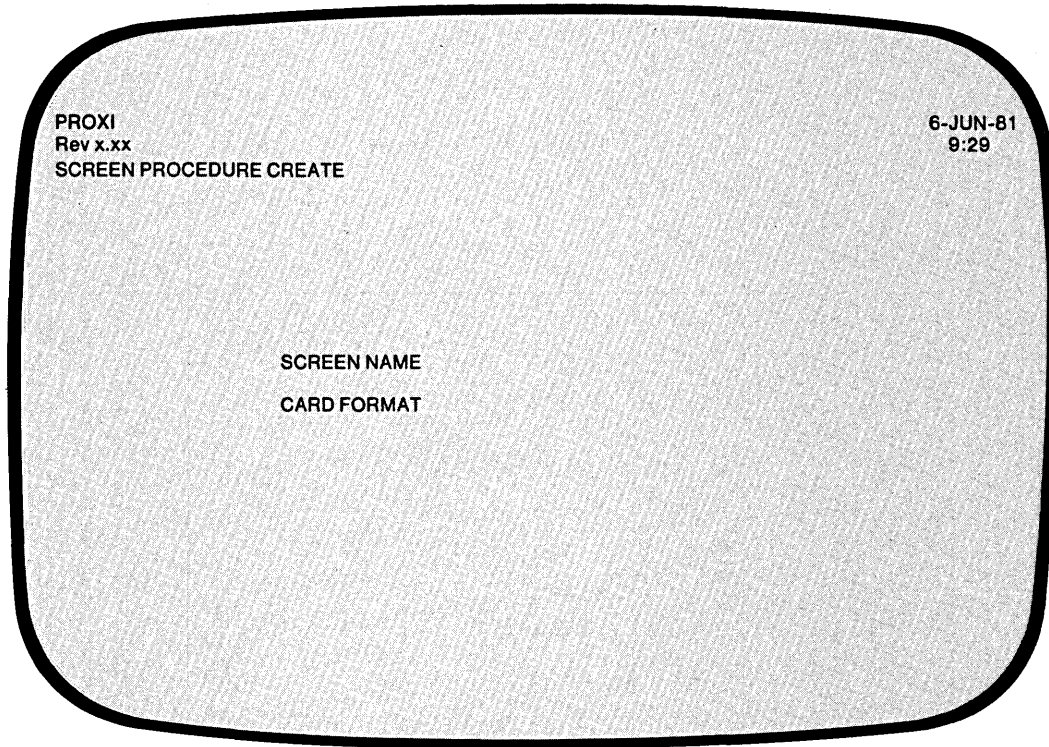
What Next?

After creating the screen section copy file, you'll return to the Screen Generator menu (SG-1).





Creating a Screen Procedure Copy File



Quick Reference

Prompt	Default	Range
SCREEN NAME	None	An AOS filename
CARD FORMAT	Y	Y or N

You've selected item #5 from the Screen Generator menu, "Create a screen procedure copy file." The Screen Generator is ready to generate a source code copy file that will provide a definition of a screen format to the Procedure Division of your program. It uses the screen definition parameter file to create this copy file.

If you have not yet created the parameter file through the "Add a Screen" or "Change a Screen" functions, press FINISHED. This will return you to the Screen Generator menu from which you can begin the screen definition procedures.)

The Prompts

SCREEN NAME

Enter the AOS filename (without the .SF extension) of the screen format parameter file for which you want to generate a Screen Procedure copy file.

CARD FORMAT

Enter Y if you want to produce a copy file with line numbers. If you want unnumbered code (CRT format), enter N. The default response is Y.

After you respond to the CARD FORMAT question, the Screen Generator begins to build a Screen Procedure copy file for the specified screen. You'll receive the message

CREATING screen.PL

As the Screen Generator builds the copy file, it displays the statement line numbers on the screen. (Only card format copy files will actually contain the statement number.)

If the file screen.PL exists, the new file overwrites it.

IMPORTANT: If you alter the parameter file after building the Screen Procedure copy file, you will probably have to rebuild the copy file. Refer to the "Change a Screen" section for details.

What Next?

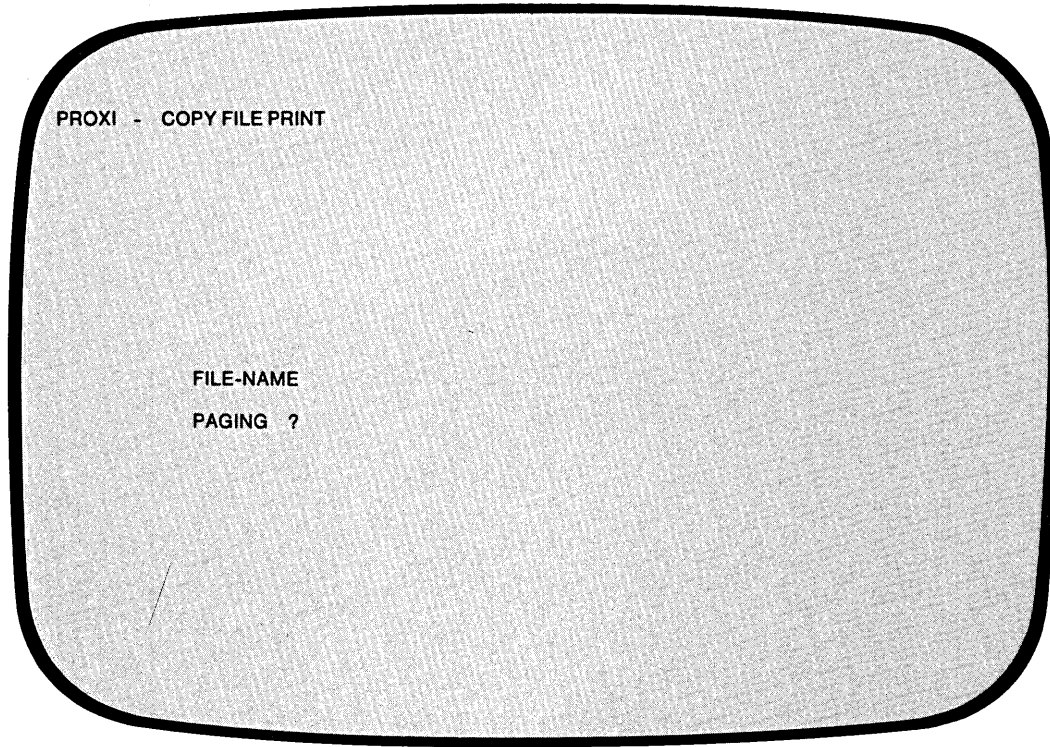
After creating the Screen Procedure copy file, you'll return to the Screen Generator menu (SG-1).





SG-15
Printing Screen Copy Files

Screen Generator



Quick Reference

Prompt	Default	Range
FILE NAME	None	An AOS filename
PAGING	Y	Y or N

You've selected item #6 from the Screen Generator menu. The Screen Generator is ready to print any copy file you request.

The Prompts

FILE-NAME

Enter the name of the copy file you want to print. Include the appropriate extension. Use the format:

screen { .SD }
 { .PL }

PAGING ?

Enter Y if you want a page break (form feed) whenever a slash (/) character appears in the indicator field. This is the default answer. If you enter N, the slash character will have no effect.

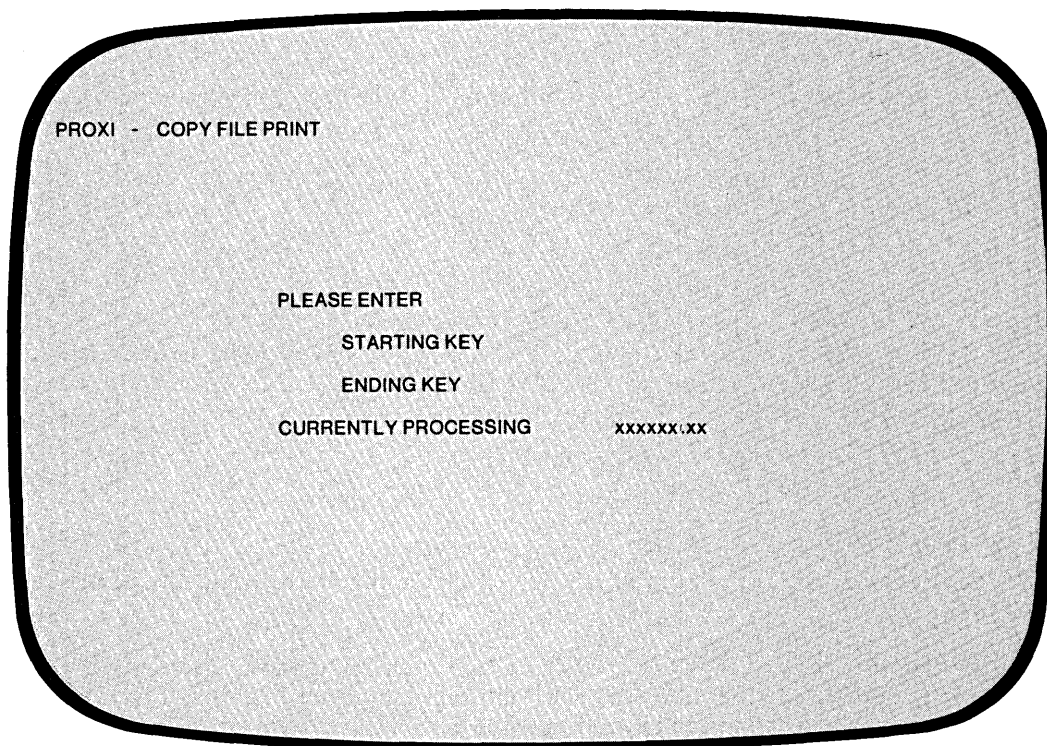
What Next?

After you respond to these questions, the Screen Generator moves on to the next screen (SG-16).

After printing the specified copy file, the Screen Generator returns to this screen and allows you to print another copy file. If you do not want to print another file, press FINISHED at the first prompt. You will immediately return to the Screen Generator menu (SG-1).

SG-16 Printing Screen Copy Files

Screen Generator



Quick Reference

Prompt	Default	Range
STARTING KEY	None	Determined by the data file
ENDING KEY	None	Determined by the data file

If you requested a card format copy file (including line numbers), you may obtain a partial listing.

The Prompts

STARTING KEY

Enter the line number where you want the listing to begin. COBOL line numbers begin with 000100 and increase by 100.

If you want a complete listing, type NEW LINE only. The Screen Generator displays ALL , and then prints the entire copy file.

ENDING KEY

If you specified a starting key, you may also enter the line number where you want the listing to end. If you enter a line number that is greater than the last line in the copy file, the printing program ignores your error and prints to the end of the file.

If you did not provide a starting key, the Screen Generator skips this prompt; it will print the entire copy file.

After you've provided all necessary information, you'll see this message:

*PRINTING . . . PLEASE WAIT
CURRENTLY PROCESSING copyfile*

If, however, the printer is unavailable, the following message appears:

*PRINTER IN USE
OUTPUT TO THE PRINTER ?*

Answer Y if you want to retry the printer. If you answer N, the Screen Generator will direct its output to a disk file called PROXI\$nn (where nn is your terminal number).

After the Screen Generator has printed the specified copy file, it presents you with the previous screen, allowing you to enter the name of another copy file for printing. If you do not want to print another copy file, press the FINISHED key. You then return to the Screen Generator menu.

What Next?

After printing the specified copy file, the Screen Generator repeats this series of screens (beginning with SG-15), allowing you to print another copy file.

End of Chapter

Chapter 4

The File Definitions Module

This chapter describes the PROXI File Definitions module. This module helps you build a set of copy files that contain specific information about the data files used by your program. The program uses COPY statements to incorporate this code within its framework. The PROXI File Definitions module allows you to

- Build a SELECT statement copy file for the File-Control Paragraph.
- Build a File Description (FD) entry copy file for the File-Control Paragraph.
- Create a copy file containing Data Description Entries for the Working-Storage Section.
- Create a Declaratives Section copy file.
- Print any of the above copy files.

The File Definitions Copy Files

To create any of these copy files, you complete a series of data-entry screens appropriate to the function you've selected. After the File Definer gathers all necessary information, it builds a copy file that corresponds to the specified data file. The naming of the copy file follows this format:

`datafile.extension`

where:

`datafile` is the name of the data file to which this copy file refers.

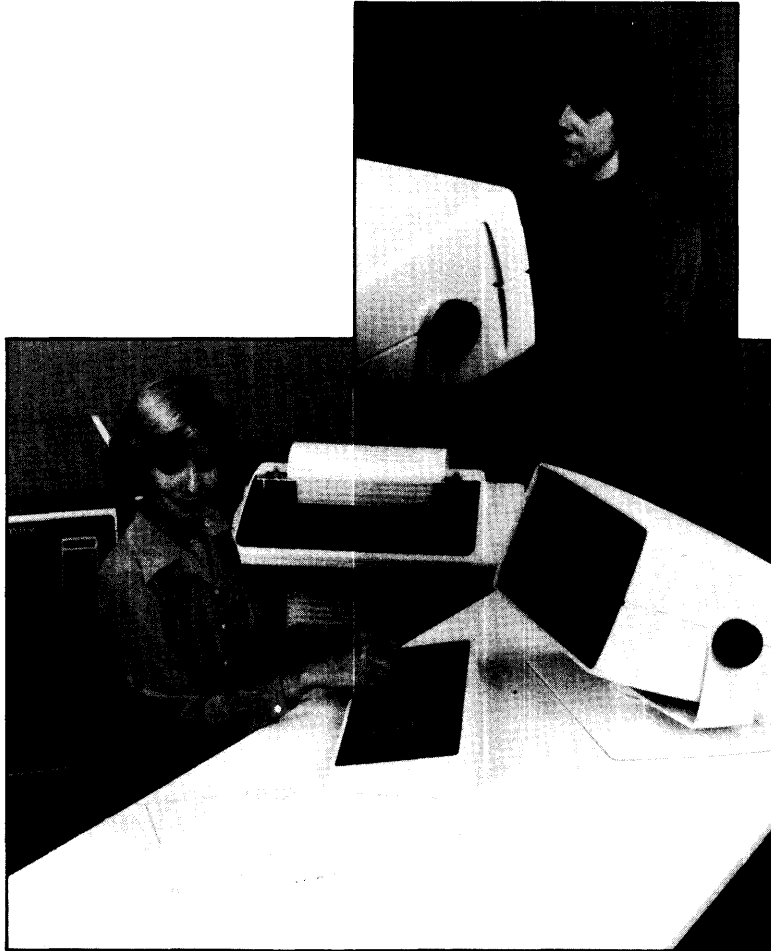
`.extension` is one of the following:

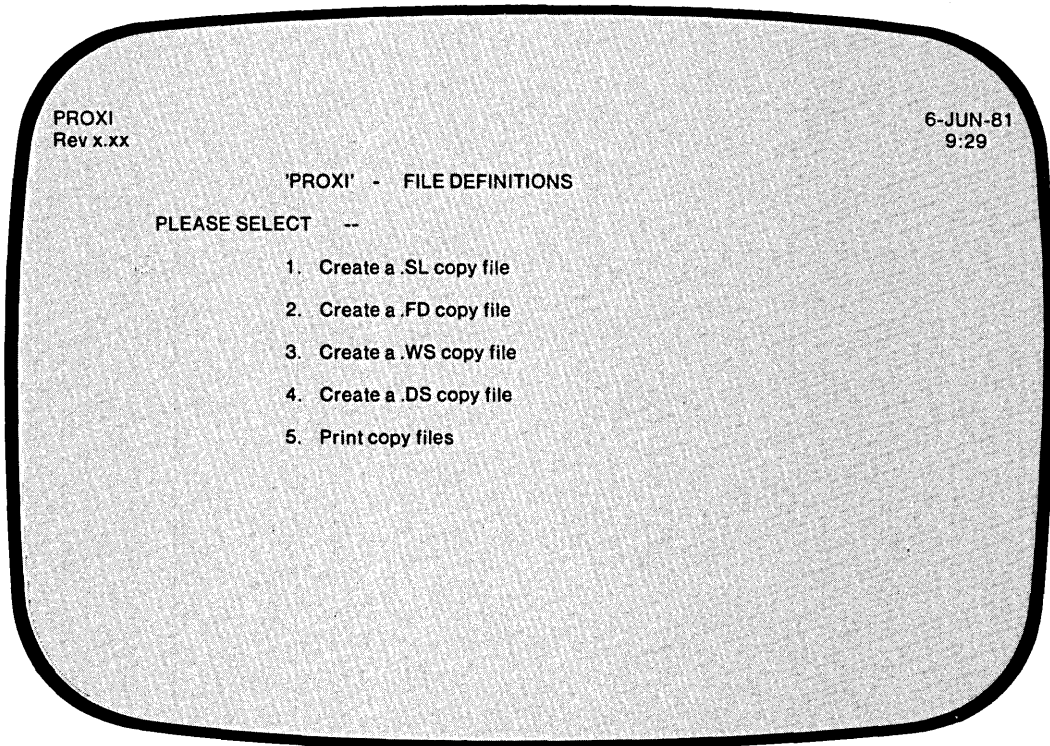
- `.SL` (a SELECT statement)
- `.FD` (a File Description (FD) entry)
- `.WS` (a Working Storage data description entry)
- `.DS` (a Declaratives Section)

You need to build each type of copy file for each data file your program will use. Other PROXI programs can use the same copy files if they work with the same data file.

You may use a text editor to change the contents of an existing copy file; you cannot use the File Definer to edit a copy file. If you specify the name of an existing file when calling one of the PROXI functions (except the Declaratives and Print functions), the new file will overwrite the existing file.

After selecting the File Definitions module from the main PROXI menu, you'll receive the File Definitions menu. Turn to the next section for a description of this menu and the options it gives you.





The function you choose will call up a specific series of data-entry screens. Turn to the appropriate tab in this section for information about your chosen file definition action.

Here is a quick summary of each function:

1. CREATE A .SL COPY FILE

Select this item to build a SELECT statement for a particular data file. You supply information for the ASSIGN TO, ORGANIZATION, ACCESS MODE, and other clauses.

2. CREATE A .FD COPY FILE

Select this item to build a File Description entry for a particular data file. You specify the RECORDING MODE and LABEL RECORDS clauses, as well as provide the data names, level numbers, and clauses for the FD entry.

3. CREATE A .WS COPY FILE

Select this item to build a set of Working Storage entries relating to a particular data file. As with the FD entry, you supply data names, their level numbers and clauses.

4. CREATE A .DS COPY FILE

Select this item to build a Declaratives Section for a particular data file. You do not have to provide any information other than the name of the data file.

5. PRINT COPY FILES

Select this function if you want to print a copy of any of the above files.

Figure 4-1 diagrams the PROXI File Definitions module, showing the various data-entry screens for each menu selection.

What Next?

Turn to the appropriate section in this manual, depending on the function you've selected.

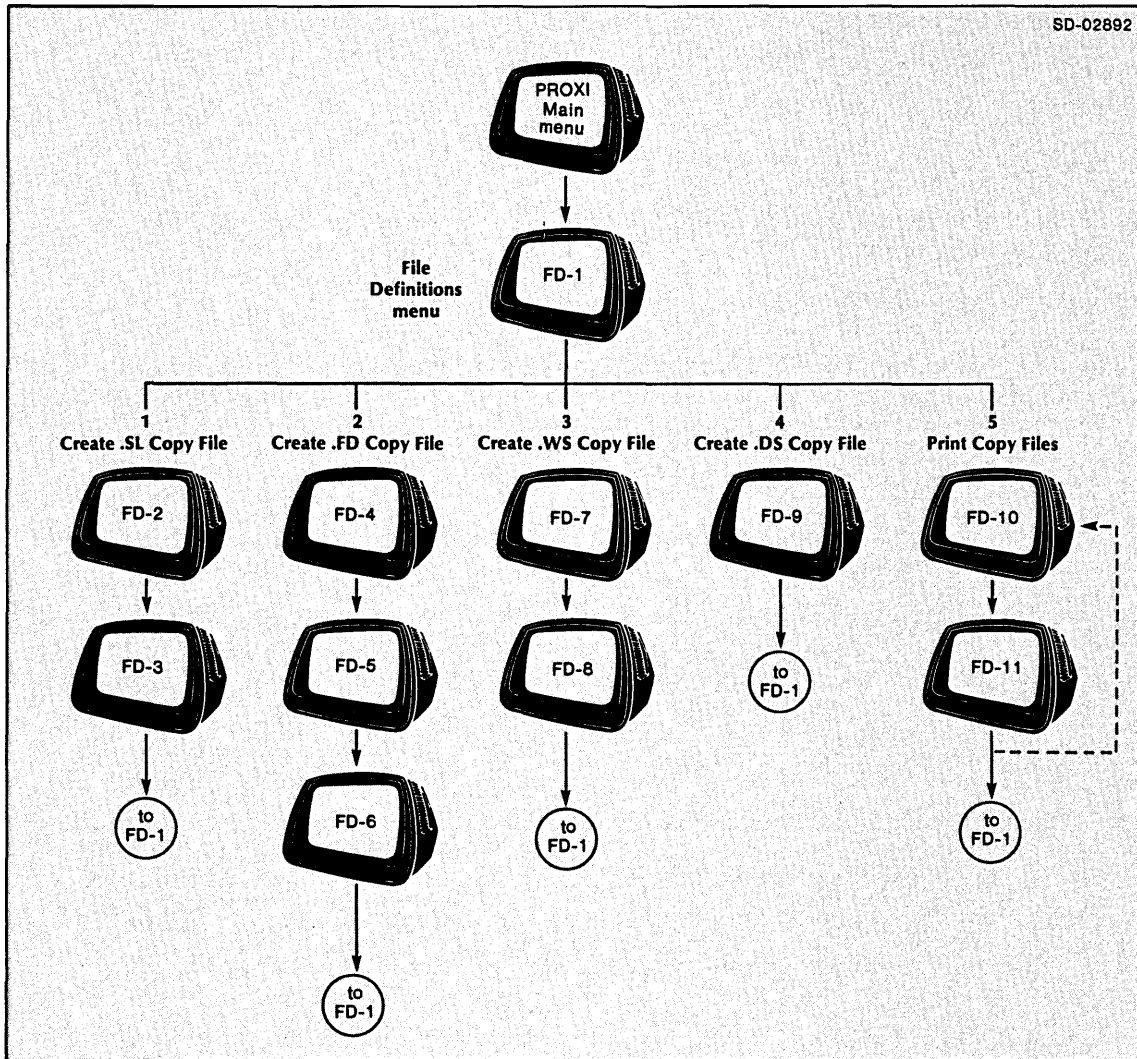
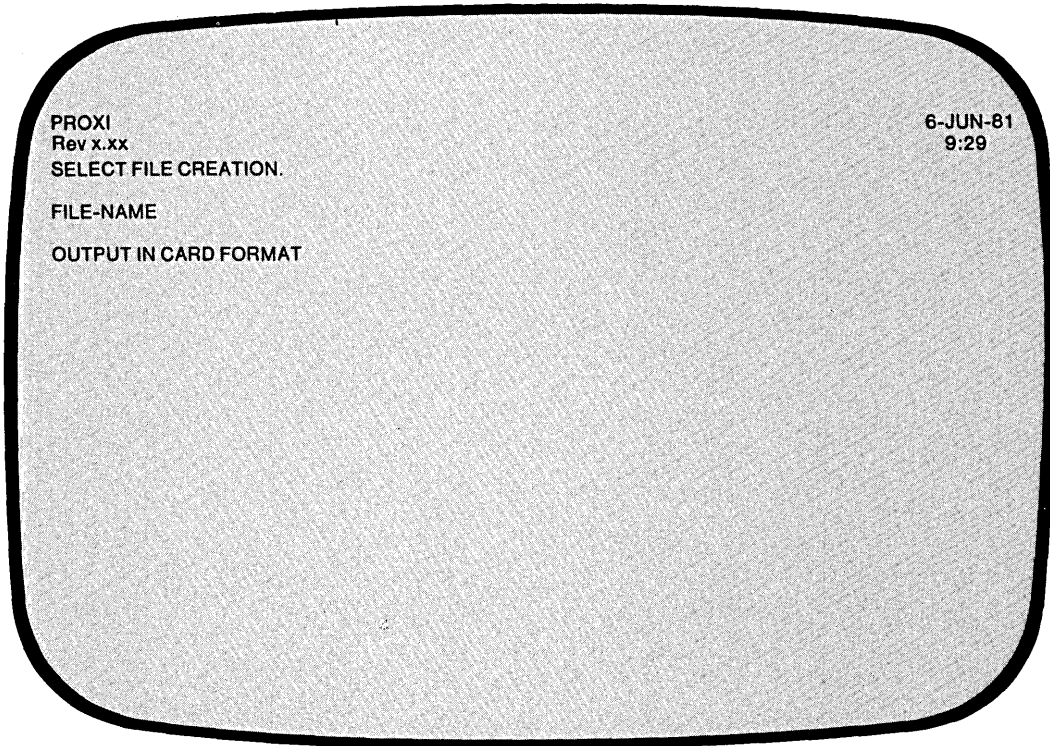


Figure 4-1. The File Definition Module Data-Entry Screens



FD-2
Building a SELECT Statement

File Definitions



Quick Reference

Prompt	Default	Range
FILE NAME	None	An AOS filename
OUTPUT IN CARD FORMAT	Y	Y or N

The File Definer is ready to build a SELECT statement copy file for a data file.

The Prompts

FILE-NAME

Enter the AOS filename that identifies the datafile you are about to describe. If datafile.SL exists, you'll receive an error message. *You cannot use this function to edit a file.* Enter another filename, or press FINISHED to cancel the function and return to the menu.

OUTPUT IN CARD FORMAT

Enter Y (the default) if you want the copy file to include line numbers. If you want unnumbered code (CRT format) enter N.

What Next?

The next data-entry screen requests you to fill in clause information for the SELECT statement (FD-3).

Building a SELECT Statement

```

PROXI
Rev x.xx
SELECT FILE CREATION.
                                     6-JUN-81
                                     9:29

      File Control Section.

SELECT xxxxxxxx
      ASSIGN TO                      " xxxxxxxx "
      ORGANIZATION IS
      ACCESS MODE IS
      RECORD KEY IS xxxxxxxx -KEY
      FILE STATUS IS  xxxxxxxx-STATUS
      INDEX SIZE IS  0000
      DATA SIZE IS  0000
    
```

Quick Reference

Prompt	Default	Range
ASSIGN TO	DISK	DISK, PRINTER, KEYBOARD or DISPLAY
ORGANIZATION IS	S*	I, R, or S
ACCESS MODE IS	S or D**	D, R, or S
INDEX SIZE IS	0	Integer
DATA SIZE IS	0	Integer

* Organization defaults to Sequential for non-disk devices. There is no default value for a disk file.

** Access mode defaults to Sequential for sequentially-organized files. The default access for files with indexed or relative organization is Dynamic.

This screen presents a skeleton SELECT statement using the name of the data file you just supplied.

The Prompts

ASSIGN TO

The default response, DISK, will appear after the prompt. Enter NEW LINE (if the data file resides on disk), or enter the appropriate device from among these:

PRINTER
KEYBOARD
DISPLAY

If you select anything other than DISK, the File Definer supplies the default value SEQUENTIAL for the ORGANIZATION and ACCESS MODE entries, and omits the INDEX SIZE prompt.

ORGANIZATION IS

If you specified DISK above, you must supply one of the following codes to describe the file's logical structure:

I for Indexed organization
R for Relative organization
S for Sequential organization

If you enter S (for sequential organization), the File Definer defaults the next entry (access mode) to sequential and omits the INDEX SIZE clause below.

ACCESS MODE IS

Supply one of the following codes to specify how COBOL gains access to the file:

D for Dynamic access
R for Random access
S for Sequential access

The default for a disk file is Dynamic. (Sequential access appears automatically for other device types.)

RECORD KEY and FILE STATUS

The name datafile-KEY automatically becomes the RECORD KEY (for indexed files only), and datafile-STATUS becomes the file status data item. These clauses are ignored for sequential files.

INDEX SIZE IS

For an indexed or relative file, enter an integer value. If you enter 0, the program will randomly allocate disk space for the index portion of the data file. Otherwise, supply the number of blocks to be reserved for contiguous allocation.

DATA SIZE IS

Enter an integer value as described above.

The INDEX SIZE and DATA SIZE clauses are ignored by AOS. We have provided them in case you want to convert an AOS PROXI program for use on a CS PROXI system.

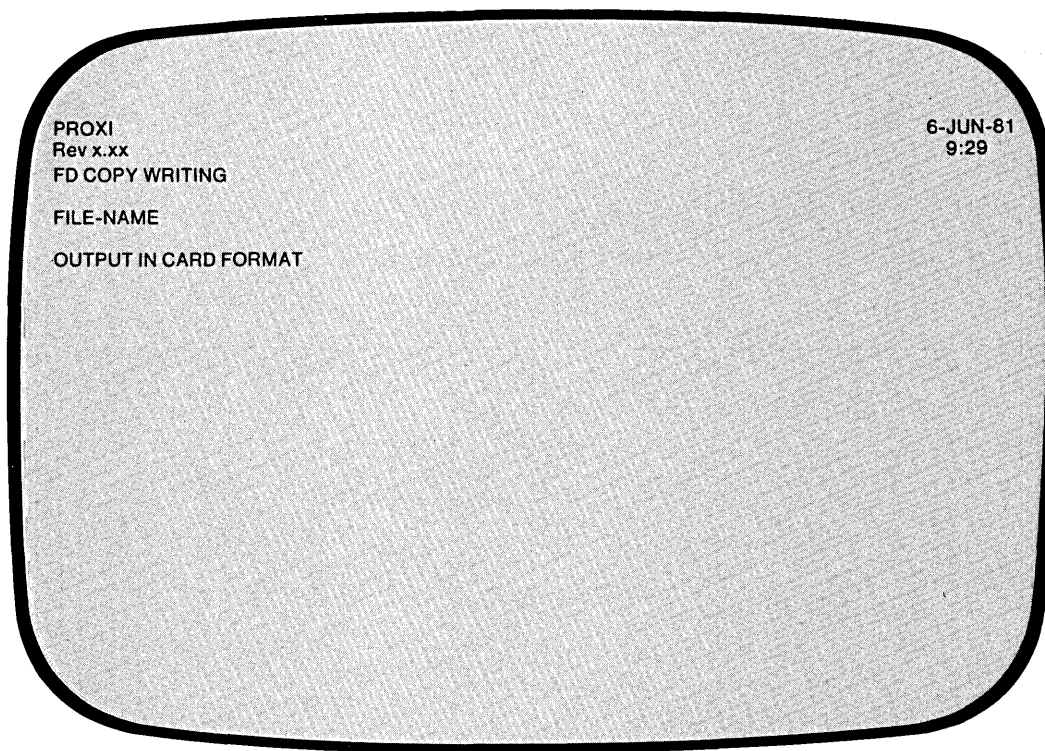
What Next?

This completes the SELECT statement. You return immediately to the File Definitions menu (FD-1).





Building a File Descriptor Entry



Quick Reference

Prompt	Default	Range
FILE NAME	None	An AOS filename
OUTPUT IN CARD FORMAT	Y	Y or N

The File Definer is ready to build a file description entry (FD) copy file for a data file.

The Prompts

FILE-NAME

Enter the AOS filename that identifies the data file you are about to describe. If datafile.FD exists, you'll receive an error message. If you want to modify a file you must use a text editor. Enter another filename or press FINISHED to cancel this function and return to the menu.

OUTPUT IN CARD FORMAT?

Enter Y (the default) if you want the copy file to include line numbers. If you want unnumbered code (CRT format), enter N.

What Next?

The next screen allows you to select the appropriate RECORDING MODE and LABEL RECORDS clauses for this FD entry (FD-5).

Building a File Descriptor Entry

PROXI
Rev x.xx

6-JUN-81
9:29

FILE SECTION.
FD XXXXXXXX
SELECT FD-TYPE _

- 1 RECORD MODE IS FIXED
LABEL RECORDS ARE STANDARD
- 2 RECORDING MODE IS FIXED
LABEL RECORDS ARE OMITTED.
- 3 RECORDING MODE IS VARIABLE
LABEL RECORDS ARE STANDARD
- 4 RECORDING MODE IS VARIABLE
LABEL RECORDS ARE OMITTED.

This data-entry screen allows you to specify the RECORDING MODE and LABEL RECORDS clauses for this data file. (The name of the data file appears at the top left of the screen in the FD entry.)

The File Definer provides the following combinations:

Code	Clauses
1	RECORDING MODE IS FIXED LABEL RECORDS ARE STANDARD
2	RECORDING MODE IS FIXED LABEL RECORDS ARE OMITTED
3	RECORDING MODE IS VARIABLE LABEL RECORDS ARE STANDARD
4	RECORDING MODE IS VARIABLE LABEL RECORDS ARE OMITTED

Select the code number that corresponds to the appropriate pair of clauses. (The code 1 appears as the default response.)

IMPORTANT: The PROXI system provides the RECORDING MODE clause for compatibility with Data General's Commercial Systems (CS) Interactive COBOL. For AOS applications, you must remove this statement from the FD copy file before compiling the program.

What Next?

After you make your choice, the next data-entry screen appears (FD-6). It displays the information gathered so far and prompts you to enter data names with their level numbers and clauses.

FD-6
Building a File Descriptor Entry

File Definitions

PROXI
Rev x.xx
6-JUN-81
9:29
FILE SECTION.
FD xxxxxxxx
RECORDING MODE IS xxxxxx
LABEL RECORDS ARE xxxxxx.
LEVEL

Quick Reference

Prompt	Default	Range
LEVEL	None	01 - 49, 77, H, *, /, or the FINISHED key
DATANAME	None	A data-file data name
CLAUSE	None	(See Table 4-1)

This data-entry screen allows you to build a file description for the specified data file. The screen displays the name of the data file as well as the RECORDING MODE and LABEL RECORDS clauses you just selected.

The Prompts

LEVEL

Enter the appropriate two-digit level number (01-49). You will immediately receive the prompt for a data name (see below).

You may also enter a comment line. To do this, enter an asterisk followed by NEWLINE. You'll receive the prompt for a comment line. For example,

LEVEL *) _____

If you want the COBOL compiler to insert a form feed before your comment (so that it appears at the top of a page), enter a slash character instead of an asterisk.

LEVEL /) _____

Your comment may be up to 62 characters long. Terminate it with NEW LINE.

If you need help, enter H) at the LEVEL prompt. The File Definer will display information at the bottom of the screen to assist you.

DATA-NAME

Enter the appropriate COBOL data name. Note that the File Definer *does not* check your entry for correct syntax.

CLAUSE

After you enter a data name, the File Definer prompts you to specify a clause. To do this, enter the appropriate clause code from Table 4-1. Some of these codes will supply a complete clause; for others you must provide additional information or codes. For example, if you enter P to specify a PICTURE clause, the screen displays

CLAUSE 03 ENTRY-ITEM PIC _____

Enter the appropriate picture string.

Note that the File Definer *does not* check your entries for accuracy or consistency. It will attach any clause you specify regardless of data type or other clauses.

An error message will signal an invalid clause code. Type NEW LINE to continue. If you need help, type H) following the clause prompt. A brief summary of the clause codes will appear at the bottom of the screen.

To complete a data name description, use the FINISHED key at the CLAUSE prompt. This will place a period at the end of the entry. You will then receive the LEVEL prompt again.

The first 01 entry you make for this data file must be the name of the data-file record. The PROXI system uses the name datafile-RECORD to refer to a record. Your entry should look like this:


01 datafile-RECORD.

If you do not make this entry, you may get compilation errors signalling an undefined data name.

What Next?

When you have completed the entire FD entry, use the FINISHED key at the LEVEL prompt. The message "LEVEL END" will appear, signalling the end of the FD entry. You'll then return to the File Definitions menu (FD-1).

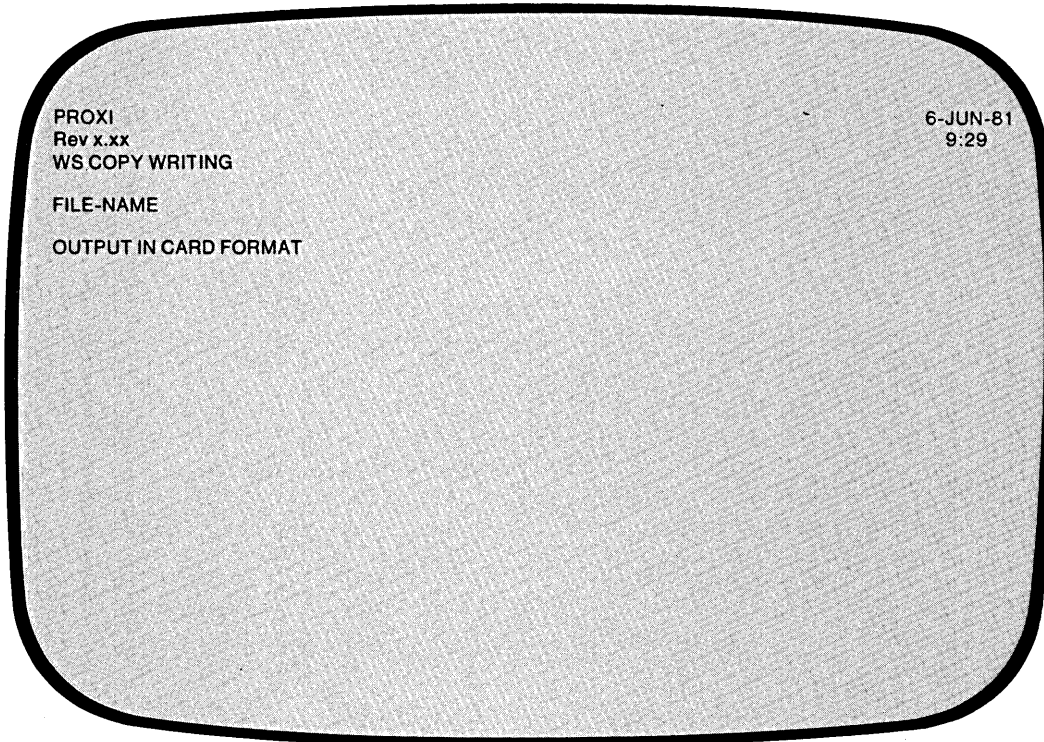
Table 4-1. Clause Codes

Key	Clause	Remarks
B	BLANK WHEN ZERO	(Requires no additional entry.)
H	none	Displays help information at the lower part of the screen.
I	INDEXED BY	Supply a COBOL data name and press NEW LINE.
)) _____	Supply additional COBOL data name for an INDEXED BY clause. Then press NEW LINE.
J	JUSTIFIED RIGHT	(Requires no additional entry.)
O	OCCURS nnnn TIMES	Supply a positive integer for nnnn and then press NEW LINE.
P	PICTURE	Supply a COBOL picture string and then press NEW LINE.
R	REDEFINES	Supply a COBOL data name and then press NEW LINE.
S	SIGN IS ____	Enter T for TRAILING or L for LEADING. Then, if you want to add SEPARATE, enter S; otherwise press NEW LINE.
U	USAGE IS	Enter D for DISPLAY, or C for COMPUTATIONAL.
Z	SYNC	Enter L for LEFT, or R for RIGHT.
	.	Press the FINISHED key to signal that the clauses for a data item are complete. A period will terminate the statement.

When you enter the clause codes I (INDEXED BY), comma (a continued INDEXED BY), or R (REDEFINES), the value most recently entered for one of these clauses will appear as the default. You may use this feature to rapidly specify several index items with similar names.



Building a Working Storage Entry



Quick Reference

Prompt	Default	Range
FILE NAME	None	An AOS filename
OUTPUT IN CARD FORMAT	Y	Y or N

The File Definer is ready to build a copy file containing data description entries for items pertaining to a data file. A PROXI program that uses this data file will insert the contents of this copy file into its Working Storage Section. Use this file to declare any data items that your program will use, but that are not declared anywhere else. If the FD statement does not include declarations for datafile-KEY and datafile-STATUS (see FD-3), you must include them here.

The PROXI program expects you to build a Working Storage file for each data file the program uses. There are no restrictions on what entries you must place in which file. Be sure not to supply duplicate or conflicting entries, however. All the copy files containing Working Storage information will be incorporated into the program structure through COPY statements. The File Definer allows you to build the Working Storage section bit by bit through these files.

The Prompts

FILE-NAME

Identify the AOS file associated with the data items you are about to define.

OUTPUT IN CARD FORMAT

Enter Y (the default) if you want the copy file to include line numbers. If you want un-numbered code (CRT format), enter N.

What Next?

The next screen prompts you to enter the individual data items with their level number and clauses (FD-8).

Building a Working Storage Entry

PROXI
Rev x.xx
WORKING-STORAGE SECTION.

6-JUN-81
9:29

LEVEL

Quick Reference

Prompt	Default	Range
LEVEL	None	01 - 49, H, *, /, or the FINISHED key
DATA-NAME	None	A data-file data name
CLAUSE	None	(See Table 4-2)

This data-entry screen allows you to define data items relating to the specified data file. The initial prompt is for a level number.

The Prompts

LEVEL

Enter the appropriate two-digit level number (01-49, or 77). You will immediately receive the prompt for a data name (see below).

You may also enter a comment line. To do so, enter an asterisk followed by a NEW LINE. You'll receive a prompt for a comment line. For example,

LEVEL *) _____

If you want the COBOL compiler to insert a form feed before your comment (so that it appears at the top of a page), enter a slash character instead of an asterisk.

LEVEL /) _____

Your comment may be up to 62 characters long. Terminate it with NEW LINE.

If you need help, enter H) at the LEVEL prompt. The File Definer will display information at the bottom of the screen to assist you.

DATA-NAME

Enter the appropriate COBOL data name. Note that the File Definer *does not* check your entry for correct syntax.

CLAUSE

After you enter the data name, the File Definer prompts you to specify one or more clauses. To do this, enter the appropriate clause code from Table 4-2. Some of these codes will supply a complete clause; for others, you must provide additional information or codes. For example, if you enter P to specify a PICTURE clause, you'll see

```
CLAUSE 03 ENTRY-ITEM PIC _____
```

Enter the appropriate picture string.

Note that the File Definer *does not* check your entries for accuracy and consistency. It will attach any clause you specify regardless of any other clauses entered. An error message will signal an invalid clause code entry. Type NEW LINE to continue.

If you need help, type H) following the CLAUSE prompt. A brief summary of the clause codes will appear at the bottom of your screen.

Required Entries

You must include a FILE STATUS item for this data file. When you build the SELECT statement copy file, the File Definer assigns a file status data item to datafile-STATUS. You must, therefore, define this item in each .WS copy file. Use this code:

```
01 datafile-STATUS PIC XX.
```

where:

datafile is the AOS filename identifying the data file.

Also, if the data file has relative or indexed organization, you must include a RECORD KEY item in this copy file. The PROXI program uses the name datafile-KEY for the RECORD KEY item. Enter it as follows:

```
01 datafile-KEY PIC 9(4) USAGE IS COMPUTATIONAL.
```

where:


datafile is the AOS filename of the data file.

To complete a data name description, press the FINISHED key at the CLAUSE prompt. A period will appear at the end of the entry. You will then receive the LEVEL prompt again.

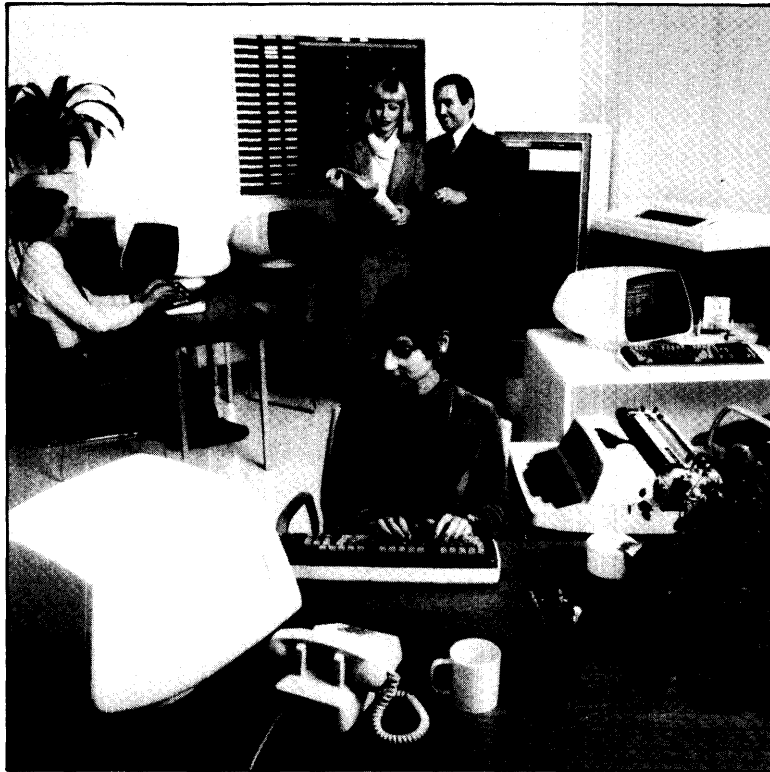
What Next?

When you have completed your entries, press the FINISHED key at the LEVEL prompt. You will see the message "LEVEL END" before you return to the File Definitions menu (FD-1)

Table 4-2. Clause Codes

Key	Clause	Remarks
B	BLANK WHEN ZERO	(Requires no additional entry.)
H	none	Displays help information at the lower part of the screen.
I	INDEXED BY	Supply a COBOL data name and then press NEW LINE.
,	' _____	Supply additional COBOL data name for an INDEXED BY clause. Then press NEW LINE.
J	JUSTIFIED RIGHT	(Requires no additional entry.)
O	OCCURS nnnn TIMES	Supply a positive integer for nnnn and then press NEW LINE.
P	PICTURE	Supply a COBOL picture string and then press NEW LINE.
R	REDEFINES	Supply a COBOL data name and then press NEW LINE.
S	SIGN IS ____	Enter T for TRAILING or L for LEADING. Then, if you want to add SEPARATE, enter S; otherwise press NEW LINE.
-	- " _____	This clause permits you to continue a literal string. You must enter the closing quotation mark.
U	USAGE IS	Enter D for DISPLAY, or C for COMPUTATIONAL.
V	VALUE IS	Enter a literal value or a COBOL data item. Note that you must enclose a literal value in quotation marks.
Z	SYNC	Enter L for LEFT, or R for RIGHT followed by NEW LINE.
	.	Press the FINISHED key to signal that the clauses for a data item are complete. A period will appear after the last clause.

When you enter the clause codes I (INDEXED BY), comma (a continued INDEXED BY), or R (REDEFINES), the value most recently entered for one of these clauses will appear as the default. You may use this feature to rapidly specify several index items with similar names.



FD-9
Building a Declaratives Section

File Definitions

PROXI
Rev x.xx
DECLARATIVE CODE CREATE

6-JUN-81
9:29

FILE NAME
OUTPUT IN CARD FORMAT

Quick Reference

Prompt	Default	Range
FILE NAME	None	An AOS filename
OUTPUT IN CARD FORMAT	Y	Y or N

The File Definer is ready to build a Declaratives Section copy file for a data file. The program will use a copy statement to include this code in its Procedure Division.

The Prompts

FILE-NAME

Enter the AOS filename that identifies the data file for which you are building a Declaratives Section.

OUTPUT IN CARD FORMAT

Enter Y (the default) if you want the copy file to include line numbers. If you want unnumbered code (CRT format), enter N.

The File Definer deletes any existing copy file with the name *datafile.DS*. It will use this name for the new Declaratives file. This file uses a standard format; you do not have to enter any additional information.

What Next?

After displaying a soothing *PLEASE WAIT* message, the File Definer builds the copy file and then returns you to the File Definitions menu (FD-1)

Figure 4-2 displays the format of the Declaratives Section.

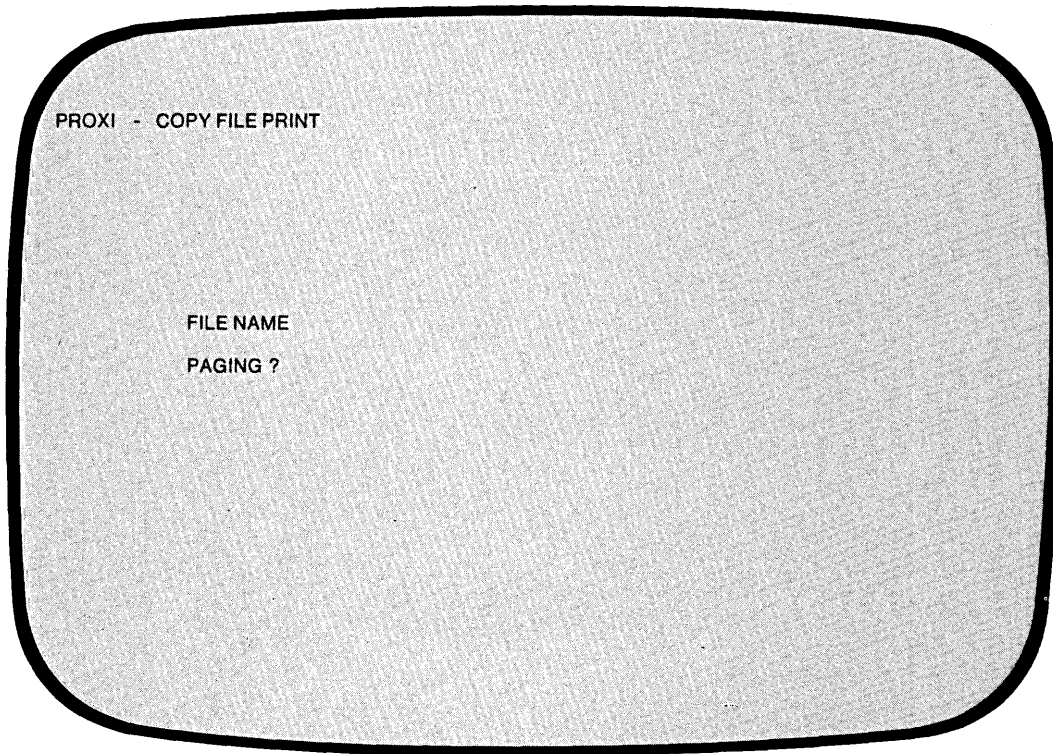
```
000100 datafile-ERROR SECTION.  
000200             USE AFTER ERROR PROCEDURE ON datafile.  
000300 PROCESS-datafile-ERROR.  
000400     IF datafile-STATUS = HARDWARE-ERROR OR  
000500     datafile-STATUS = OPEN-ERROR OR  
000600     datafile-STATUS = DISK-FULL OR  
000700     datafile-STATUS = OVER-LOCK-LIMIT  
000800     MOVE datafile-STATUS TO FILE-ERROR-STATUS  
000900     MOVE "datafile" TO FILE-ERROR-NAME  
001000     DISPLAY FILE-ACCESS-ERROR-SCREEN  
001100     STOP RUN.
```

Figure 4-2. Declaratives Code (Card Format)



FD-10
Printing Datafile Copy Files

File Definitions



Quick Reference

Prompt	Default	Range
FILE NAME	None	An AOS filename
PAGING	Y	Y or N

You've selected the Print function from the File Definitions menu. The File Definer is ready to print any copy file you request.

The Prompts

FILE NAME

Enter the name of the copy file you want to print. Include the appropriate extension. Use the format:

datafile (.SL)
(.FD)
(.WS)
(.DS)

If the file does not exist you will receive an error message. Press NEW LINE to enter another filename, or press FINISHED to cancel the Print function and return to the File Definitions menu.

PAGING ?

Enter Y (the default) if you want a page break (form feed) whenever a slash (/) character appears in the indicator field. If you enter N, a slash will have no effect on the listing.

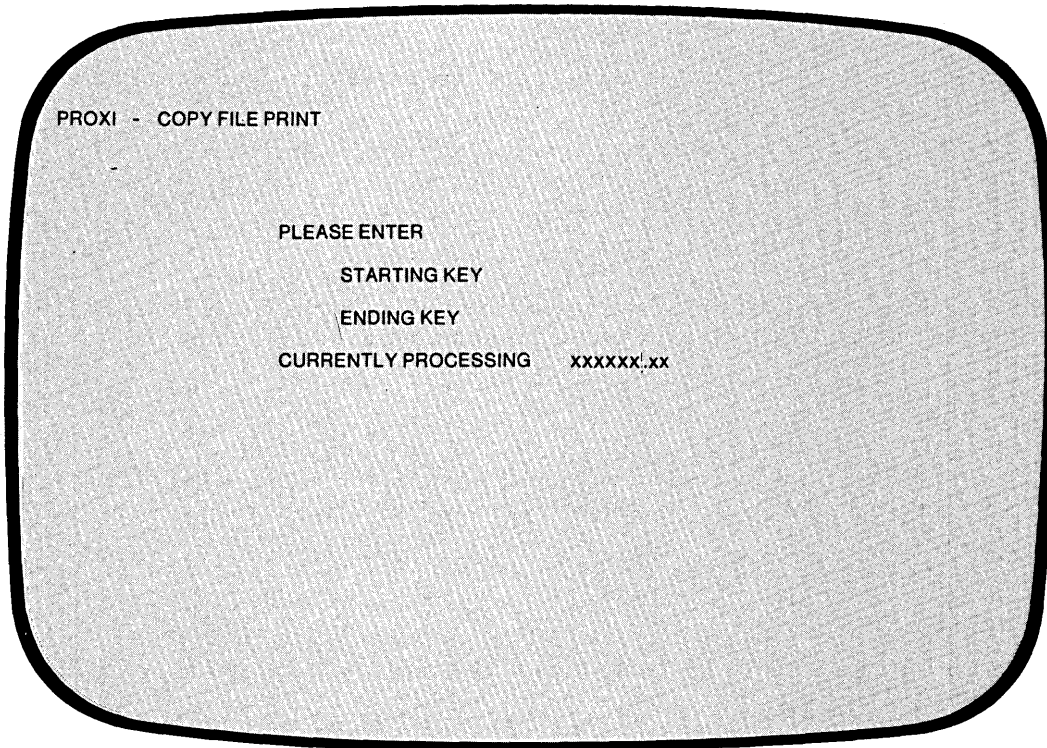
What Next?

The next screen allows you to specify a particular range of the copy file to be used for printing (FD-11).

After printing a copy file, the File Definer returns you to this screen, allowing you to repeat the operation. If you do not want to print another copy file, press FINISHED at the first prompt. You will return to the File Definitions menu (FD-1).

FD-11
Printing Datafile Copy Files

File Definitions



Quick Reference

Prompt	Default	Range
STARTING KEY	Entire file	000100 - last line
ENDING KEY	Last line	000100 - last line

If the copy file is in card format (with line numbers) you may obtain a partial listing.

The Prompts

STARTING KEY

Enter the line number where you want the listing to start. COBOL line numbers begin with 000100 and increase by 100.

If you want a complete listing, type NEW LINE only. The File Definer displays the word ALL , and then prints the entire copy file.

ENDING KEY

Enter the line number where you want the listing to end. If you enter a line number greater than the last line of the file, the printing program ignores your error and prints to the end of the file. If you did not supply a starting key, the File Definer skips this prompt; it will print the entire file.

After the Any Change cycle, you will see the message:

*PRINTING . . . PLEASE WAIT
CURRENTLY PROCESSING datafile*

If the printer is unavailable, you'll receive the following message:

*PRINTER IN USE
OUTPUT TO THE PRINTER ?*

Answer Y if you want to retry the printer. If you answer N, the File Definitions module directs output to a disk file named PROXI\$nn (where nn is your terminal number).

What Next?

After the File Definer has printed the specified copy file, it returns to the previous screen (FD-10), allowing you to enter the name of another copy file for printing.

End of Chapter

Chapter 5

Running and Documenting a PROXI Program

Compiling the Code

Before compiling the source code for your PROXI program, be sure that you've

- created a SELECT statement file, a File Descriptor file, a Working Storage entries file, and a Declaratives Section file for *each* data file your program will use.
- created a Screen Procedure and Screen Section copy file for *each* screen format your program will use (if any).
- created the program framework through the Program Generator.
- provided the Own Code that the program will use (if any).

Other Hints

Follow the instructions in the appropriate COBOL Reference Manual to compile the source code. If you produced card format code, remember to use the /C command switch.

You may receive compilation errors. These errors usually indicate that you've made inconsistent entries (i.e., referred to a data item with two different data names), omitted required entries (such as status or key data items), or that you've specified illegal options.

Note that each FD statement copy file will contain a RECORDING MODE clause. AOS accepts this clause for sequential files only. If you define an Indexed or Random file, use an editor to remove this clause from the .FD file before you attempt to compile the program.

Binding the Program

Remember to include ICALL in your CBIND line if your PROXI program uses INFOS® II indexed files. ICALL is the COBOL interface to the INFOS® II system.

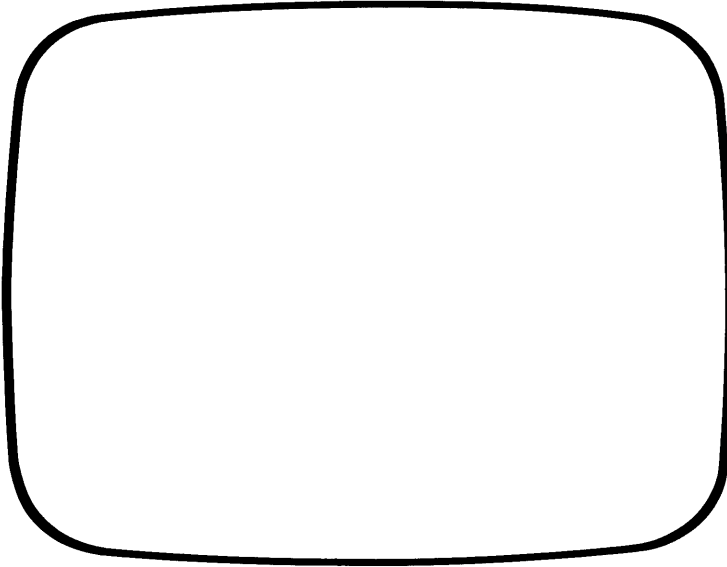
Documenting the Program

Appendix D gives the program operator a general introduction to the PROXI system. Because each PROXI program is custom made, we can only explain so much; the rest is up to you.

To make it easy for you to document your program, we've provided a "Do-It-Yourself" form. You can use this form to describe screen formats for File Maintenance and File Inquiry programs. Simply reproduce the screen image, supply a brief explanation of each prompt including the range of acceptable answers, the default value, and any special information you want the operator to be aware of.

End of Chapter

PROXI Screen Format



Program Name: _____

Screen Number: _____ of _____

Prompt:
Range: _____ Default: _____
Explanation:
Prompt:
Range: _____ Default: _____
Explanation:
Prompt:
Range: _____ Default: _____
Explanation:
Prompt:
Range: _____ Default: _____
Explanation:
Prompt:
Range: _____ Default: _____
Explanation:
Prompt:
Range: _____ Default: _____
Explanation:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

Appendix **A**

PROXI Terminology

This appendix presents a summary of terms used in this manual. For details about a particular term, refer to the appropriate sections of the manual.

Accumulator

One of 99 PROXI registers available for performing calculations. The accumulators that you have access to are named A01 through A99. There are also automatic totaling accumulators which you do not have direct access to.

Any Change Cycle

The Any Change cycle allows you to verify or change your entries to a data-entry screen. After you have made the last response, the question "Any Change ?" appears at the bottom of the screen. You may verify your entries (by responding N or pressing NEW LINE only), or you may respond Y. You identify the item you want to change by entering its prompt number following the "What Number ?" question. This cycle repeats until you respond N.

Automatic Totaling

For Report Writer and Form Printing Programs you may request automatic totaling of any field. When a control break occurs at the appropriate level, totals are generated for the specified fields and a total line is printed.

Card Format

The PROXI system prefers to create code in card format; that is, with line numbers. The alternative is CRT format (without line numbers).

Change Methods

The way in which an operator may alter an entry field. The Replacement method requires the operator to enter a new value. The Edit method allows the operator to modify an existing value character by character. The No-Change method prevents the operator from modifying the displayed value.

Column 999

A nonprinting field used in Report Writer and Form Printing Programs to perform calculations.

Conditional Field

Part of a print line. The PROXI system allows you to specify a field to be conditionally included on the next print line.

Conditional Printing

A feature of Report Writer and Form Printing Programs that allows the programmer to specify from one to 99 logical tests to be performed to determine whether or not to print a line. Conditional printing may govern detail, total, top of form, and page break lines.

Control Break Field

For Report Writer and Form Printing Programs you can specify one or more control break fields. When the value of this field changes as the next record is read, a control break occurs generating one or more levels of totaling.

Copy File

A file containing COBOL source code. This code becomes part of a program through the use of a COPY statement. The code in the file replaces the COPY statement in the program. This feature allows many programs to use the same set of code without duplication.

CRT Format

Source code generated without line numbers. The alternative form is card format, which includes line numbers.

Data Dictionary

A list of data names and their pictures. The Program Generator builds the Data Dictionary using information entered in the .SL and .FD files.

Data-item

A variable that is defined either as a record field or as a Working Storage item.

Data file

A file used by a PROXI program for input and/or output. Each PROXI program uses a principal data file. The program may optionally use up to nine additional data files called reference files.

Default Value

The default value is an assumed response or entry when only a NEW LINE is entered.

DELETE Key

A PROXI function key that allows a serially-defined item to be removed from the series. This key operates only during a Change operation.

Detail Field

One of up to 99 fields within a detail line. This field may represent a literal, a nonprinting field for computation, or a data-item value.

Detail Print Line

One of up to nine lines that appear in Report Writer and Form Printing programs for each processed record of the principal data file.

Display Field

A type of field specified in a screen format. This field displays a value after the program processes an entry made by the operator.

.DS File

A copy file containing Declaratives Section source code for a particular data file.

Entry Field

A field in a screen format which accepts an entry from the program operator.

Entry Sequence Number

A numeric prefix to a prompt, indicating the place it holds in the sequence of entries. A numbered prompt automatically assigns a sequence number to an entry field. When creating a screen format you may explicitly assign a sequence number to a field.

Escape (ESC) Key

This key allows you to clear your entries to the current screen and return to the initial state (blank if you are working with a new screen, or the previous entries if you are changing a screen).

.FD File

A source code copy file that contains file descriptor entries for a particular data file.

Field Validation

One or more logical tests to be performed on an operator entry to determine whether or not to accept the entry value.

FINISHED Key

The PROXI function key that signals completion of a screen format or an operation. The FINISHED key, when used at a data-entry screen, cancels a serial-definition process or returns you to the next higher level.

File Maintenance Program

A PROXI program type that allows the operator to select from four operations to a principal data file: add a record, change/display a record, delete a record, or print the file's contents.

File Inquiry Program

A PROXI program that allows the operator to examine one or more records in a principal data file.

Form Printing Program

A PROXI program that generates forms using a principal data file and up to nine additional reference files.

Format Field

A field that displays a data-item value as the screen is displayed. This field does not depend on any operator entry (as does a display field).

Header Line

One of up to nine lines printed at the top of each page of a report or form. These lines contain only literals.

INSERT Key

A PROXI function key used to insert a serial item into a sequence. This key operates only when changing a program's parameter file.

Key Field

A field, within a record, that allows the record to be identified or located.

Key Range Selection

A feature that allows the operator to specify a particular range of records on which an operation is to be performed within a data file.

Legend Line

One of up to nine lines of text that appear on the first page of a report (and optionally on all subsequent pages).

Maintenance Menu

The list of four operations that a PROXI File Maintenance Program can perform on the principal data file.

NEXT RECORD Key

A PROXI function key used only by the program operator to indicate the next sequential record on the principal data file. This key eliminates the need for the operator to enter a key field to identify a data-file record.

Numbered Prompt

A prompt that is preceded by a number that indicates the place it holds in the sequence of entry fields. The number may be used to identify the entry item during the Any Change cycle.

Own Code

Programmer-supplied source code that becomes part of the program framework through a COPY statement.

Page Break Line

A line that is printed on a report when printing reaches the bottom of a page. Usually this involves subtotals.

Parameter File

A file containing information about a program. This file is used to generate source code.

.PL File

A source code copy file containing code to be included into the procedure division of a PROXI program.

.PP File

A parameter file containing information about a PROXI program. This file is used to generate the COBOL code for the program.

Principal Data file

The main data file used by the PROXI program. This is the file on which a File Maintenance or File Inquiry Program operates. It is the main source of information for a Report Writer or Form Printing Program.

Prompt

A request for information. A prompt precedes an entry field and usually specifies the type of information required.

Record

A unit of information within a data file.

Record Field

A unit of information within a record.

Record Key

A field within a record that may be used to locate the record in the data file.

Record Selection

One or more logical tests to be performed to determine whether or not to accept a particular record for processing.

Reference File

A data file that provides additional information for processing the principal data file. Reference files eliminate the need to duplicate information on a single file.

Report Writer Program

A PROXI program that generates reports using a principal data file and up to nine reference files.

Screen Field

A field within a screen format. There are four main types of screen fields: literal, entry, format, and display.

Screen Format

An image of a data-entry screen. File Maintenance and File Inquiry Programs use one to four screen formats to interact with the program operator. The screen format may contain a number of different types of fields which may display variables and constants or request data from the operator.

Screen Image

The picture of a screen format.

Serially Defined Item

One item in a series of similar items. The PROXI system allows you to define a series of logical tests (for record selection, field validation, and conditional printing), a series of lines (detail, total, page break, and top of form lines), reference files, etc. When requesting serial information, the appropriate PROXI module loops through one or more screens until you signal that the series is complete (or provide the maximum number of entries).

SKIP IT Key

A PROXI function key that allows you to automatically verify all the remaining entries when changing a Report Writer or Form Printing Program.

Sort Key

A record field on which you perform a sort operation. The PROXI system allows ten sort keys when sorting the principal data file.

.SD File

A copy file containing Screen Section code for a particular screen format.

.SF File

A screen format parameter file. This file contains information about a screen format; source code copy files may be generated from this information.

.SL File

A source code copy file containing a SELECT statement for a data file.

Top-of-Form Print Line

One of up to nine lines that will appear at the top of the first (or only) page of a form.

Total Print Line

One of up to nine lines printed on reports and forms when a control break occurs. The total line may include calculated values, automatic totals, and/or literals.

.WS File

A source code copy file containing Working Storage entries for a data file.

End of Appendix

Appendix **B**

PROXI Error Messages

This appendix lists the error messages and error codes you may encounter while using the PROXI system.

Error Messages

BAD CLAUSE ... <CR> TO CONTINUE

The code you entered is not valid for the type of clause you are building. Refer to the appropriate screen description in this manual.

FILE DOES NOT EXIST <CR> TO CONTINUE

The file you named does not exist. Check your spelling. You may have to add the extension to identify the correct file (for Printing functions, for example).

INVALID CLAUSE TYPE ... <CR> TO CONTINUE

You entered an invalid code when trying to specify a clause for a data name. Type H following the clause prompt for a summary of the clauses you may use. Refer to the appropriate screen description for a full explanation of the clause codes.

NOT LEVEL NUMBER OR INDICATOR <CR> TO CONTINUE

You entered illegal characters in response to the LEVEL prompt. You must enter a two-digit number; use 0 to prefix numbers one through nine. The valid entries are 01 through 49, and 77 (for Working Storage only).

NOT VALID DEVICE TYPE "CR" TO CONTINUE

You may choose from the following devices: DISK, DISPLAY, KEYBOARD, or PRINTER. Enter the device type exactly as shown here; you may not abbreviate.

NOT VALID LEVEL NUMBER <CR> TO CONTINUE

You entered an illegal level number. The valid entries are 01 through 49, and 77 (for Working Storage only).

NOT VALID ORGANIZATION TYPE "CR" TO CONTINUE

You may choose from the following types of file organization: I (Indexed), R (Relative), or S (Sequential). Enter only the initial, uppercase letter.

PLEASE NOTE ERROR STATUS <CR> TO CONTINUE

The PROXI system encountered an error. Take note of the displayed error code and filename. Refer to the section of this appendix that lists the numeric error codes.

PROGRAM ALREADY EXISTS <CR> TO CONTINUE

You named an existing program parameter file (program.PP). Either specify another name (to build a new program) or return to the Program Generator menu and select the Change function (to modify the existing program file).

SCREEN DOES NOT EXIST <CR> TO CONTINUE

You named a nonexistent screen format file when calling the "Change a Screen" function. The Screen Generator searches for screen.SF.

SCREEN EXISTS <CR> TO CONTINUE

When attempting to create a new screen format file, you named a file that already exists.

SELECT FILE ALREADY EXISTS TYPE "CR" TO CONTINUE

You named an existing SELECT (.SL) file. You can use the Print function to obtain a listing of the existing file. If you want to modify the file you must use a text editor.

Error Codes

The following code values may appear with this message:

PLEASE NOTE ERROR STATUS <CR> TO CONTINUE

Code	Meaning
00	Successful completion.
10	End of file detected.
22	Invalid key condition - duplicate key not permitted.
23	Invalid key condition - the record does not exist.
24	Invalid key condition - the relative key value is too large.
30	I/O error (such as data check, parity error, or transmission error).
34	The disk is full; there is no room to add the record.
91	An error occurred when the program attempted to open a file. The file may not exist.
92	The file is either not opened, or not opened in the correct mode. The file may be locked by a previous CLOSE with the LOCK option.
94	The record is either locked, opened exclusively, or in use (if you issued an exclusive OPEN).

Code	Meaning
96	The record is marked as logically deleted.
97	The program attempted to REWRITE or DELETE without executing the previous READ for an indexed file with sequential access.
99	An INFOS error occurred. If you specified an INFOS status item in the SELECT clause, the status item contains the INFOS error code.

End of Appendix

Appendix C

Sample Programs Built Using the PROXI System

We provide this appendix to better illustrate the steps you take to build a PROXI program. Here's a sample situation:

The realtors at Birks and Abergeldie Real Estate want a program that will update their database of house and property listings, and another program that will produce a report showing the available houses and properties in each town. Using Data General's PROXI software, their programmer built File Maintenance and Report Writer programs.

The following pages record each data-entry screen and each entry made during the PROXI sessions.

The File Maintenance Program

The realtors use an indexed file to store information about the house listings handled by their office. Figure C-1 illustrates the record format. The listing number serves as the key to identify a particular record.

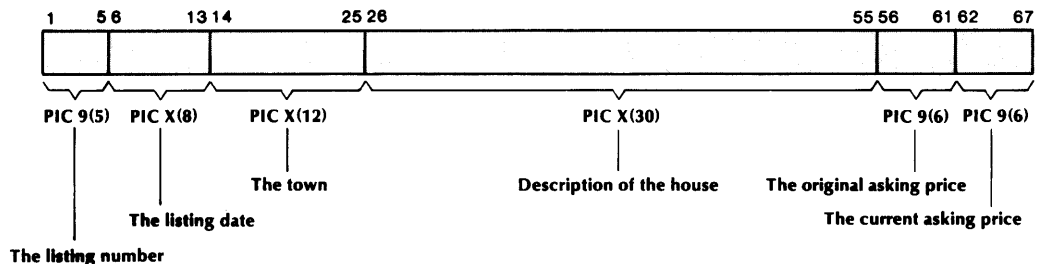


Figure C-1. Record Layout

SD-02894

Screen FD-8

01 LISTDATA-STATUS

PIC XX.

Screen FD-1

Select function #4 from the File Definitions menu:
"Create a .DS copy file".

Screen FD-9

FILE NAME
OUTPUT IN CARD FORMAT?

LISTDATA
Y

Screen FD-1

Press FINISHED to return to the Main Menu.

Main Menu

Enter 2 to call the Screen Generator module.

Screen SG-1

Select function #1 from the Screen Generator menu:
"Add a new screen".

Screen SG-2

SCREEN NAME

LISTSCR

Screen SG-3

Enter the screen as shown in Figure C-2.

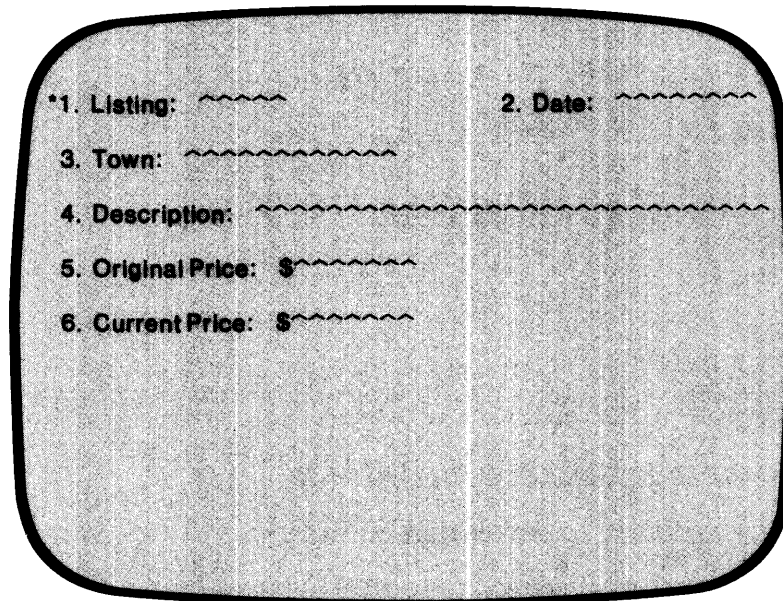


Figure C-2. Screen Layout

Screen SG-4

BLANK SCREEN? Y

Screen SG-6

(This screen repeats for each field in the format.)

LINE 6 FIELD 1 START COL 30 LENGTH 5

NAME OF FIELD	LISTDATA-KEY
TYPE OF DATA	9
DISPLAY FORMAT	ZZZZ9
CHANGE METHOD	R (Replacement)
PROMPT	

"Check the Red Book before assigning a new listing."

REQUIRED	N
FULL	N
SECURE	N
DEFAULT VALUE	}
VALIDATION	N
OWN CODE	N

LINE 6 FIELD 2 START COL 56 LENGTH 8

NAME OF FIELD	LIST-DATE
TYPE OF DATA	X
DISPLAY FORMAT	X(8)
CHANGE METHOD	R (Replacement)
PROMPT	"Use dd/mm/yy format."
REQUIRED	Y
FULL	Y
SECURE	N
DEFAULT VALUE	}
VALIDATION	N
OWN CODE	N

LINE 9 FIELD 1 START COL 20 LENGTH 12

NAME OF FIELD	TOWN
TYPE OF DATA	X
DISPLAY FORMAT	X(12)
CHANGE METHOD	R (Replacement)
PROMPT	}
REQUIRED	Y
FULL	N
SECURE	N
DEFAULT VALUE	}
VALIDATION	N
OWN CODE	N

LINE 12 FIELD 1 START COL 27 LENGTH 30

NAME OF FIELD	DESCRIPTION
TYPE OF DATA	X
DISPLAY FORMAT	X(30)
CHANGE METHOD	R (Replacement)

PROMPT

“Be sure to include the type of heating fuel.”

<i>REQUIRED</i>	Y
<i>FULL</i>	N
<i>SECURE</i>	N
<i>DEFAULT VALUE</i>	}
<i>VALIDATION</i>	N
<i>OWN CODE</i>	N

LINE 15 FIELD 1 START COL 31 LENGTH 7

<i>NAME OF FIELD</i>	ORIGINAL-PRICE
<i>TYPE OF DATA</i>	9
<i>DISPLAY FORMAT</i>	999,999
<i>CHANGE METHOD</i>	R (Replacement)
<i>PROMPT</i>	}
<i>REQUIRED</i>	Y
<i>FULL</i>	N
<i>SECURE</i>	N
<i>DEFAULT VALUE</i>	}
<i>VALIDATION</i>	N
<i>OWN CODE</i>	N

LINE 18 FIELD 1 START COL 31 LENGTH 7

<i>NAME OF FIELD</i>	CURRENT-PRICE
<i>TYPE OF DATA</i>	9
<i>DISPLAY FORMAT</i>	999,999
<i>CHANGE METHOD</i>	R (Replacement)
<i>PROMPT</i>	

“Default value is the original price.”

<i>REQUIRED</i>	N
<i>FULL</i>	N
<i>SECURE</i>	N
<i>DEFAULT VALUE</i>	ORIGINAL-PRICE
<i>VALIDATION</i>	N
<i>OWN CODE</i>	N

Screen SG-1

Select function #4 from the Screen Generator menu: “Create a screen section copy file.”

Screen SG-13

<i>SCREEN NAME</i>	LISTSCR
<i>CARD FORMAT</i>	Y

Screen SG-1

Select function #5 from the Screen Generator menu: “Create a screen procedure copy file.”

Screen SG-14

<i>SCREEN NAME</i>	LISTSCR
<i>CARD FORMAT</i>	Y

Screen SG-1

Press FINISHED to return to the Main Menu.

Main Menu

Enter 1 to call the Program Generator module.

Screen PG-1

Select function #1 from the Program Generator menu: "Create a new program".

Screen PG-2

<i>PROGRAM NAME</i>	LISTINGS
<i>PROGRAM TYPE</i>	MAINT
<i>APPLICATION NAME</i>	Current Listings Update
<i>NEXT PROGRAM (NORMAL)</i>	LOGON
<i>NEXT PROGRAM (ERROR)</i>	LOGON

Screen PG-3

<i>FILE NAME</i>	LISTDATA
<i>MENU ITEM NAME</i>	Listings
<i>SCREEN NAME 1</i>	LISTSCR
<i>OWN CODE .WS</i>	N
<i>OWN CODE .PL</i>	N
<i>PRINT PROGRAM</i>	SHOWLIST (the Report Writer Program)
<i>REFERENCE FILES</i>	N

Screen PG-5

<i>GENERATE COBOL CODE</i>	Y
<i>OUTPUT FORMAT</i>	1 (card format)

Screen PG-1

Press FINISHED to return to the Main Menu.

Main Menu

Press FINISHED to exit the PROXI session.

You may now compile the code. Figure C-5 (later in this appendix) is the compilation listing for this program.

The Report Writer Program

When asked in screen PG-3 for a Print program to process the file, we supplied the name SHOWLIST. This is the name of the Report Writer program designed to print the listings for the realtors. This program reports the available houses and properties in each town handled by this office.

Figure C-3 illustrates the report's format.

```

      BIRKS & ABERGELDIE REAL ESTATE COMPANY
This report shows the active listings for this office.
For information about old listings, see the Blue Book.

```

Town	Listing	Description	Price
XXXXXXXXXX	XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	\$9,999,999
XXXXXXXXXX	XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	\$9,999,999 REDUCED
XXXXXXXXXX	XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	\$9,999,999
Total value for this town:			\$9,999,999
XXXXXXXXXX	XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	\$9,999,999
XXXXXXXXXX	XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	\$9,999,999

Figure C-3. Report Layout

This program uses the same data file as does the File Maintenance Program; there is no need to redefine it. Because the Report Writer Program does not use any additional data files or any screen formats, the programmer calls the Program Generator.

The PROXI session begins with the main menu.

Main Menu

Enter 1 to call the Program Generator module.

Screen PG-1

Select function #1 from the Program Generator menu:
 "Create a new program".

Screen PG-2

PROGRAM NAME	SHOWLIST
PROGRAM TYPE	REPORT
APPLICATION NAME	Current Listings
NEXT PROGRAM (NORMAL)	LOGON
NEXT PROGRAM (ERROR)	LOGON

Screen PG-9

NAME OF FILE TO PRINT	LISTDATA
KEY RANGE SELECTION	N
SORT FILE	Y
ANY REFERENCE FILES	N

Screen PG-11

SORT KEY 1
SORT KEY 2
SORT KEY 3

TOWN
CURRENT-PRICE
LISTDATA-KEY

Screen PG-12

PRINT OR SPOOL
SPOOL FILE
PAGE WIDTH
NO. PRINT LINES
RUN TIME PRINTED
REPORT TITLE 1-30
REPORT TITLE 31-60
LEGEND LINES
HOW MANY PAGE 2
HEADING LINES

O (Operator's option)
SPOOLIST
80
60
N
BIRKS & ABERGELDIE REAL ESTATE
COMPANY
Y
0
Y

Screen PG-13

Legend Line 1:

This report shows the active listings for this office.

Legend Line 2:

For information about old listings, see the Blue Book.

Legend Line 3:

(Press FINISHED.)

Screen PG-14

Header Line 1:

□□□□□Town□□□□□□□□Listing□□□□□□□□□□Description□□□□□□□□□□□□□□□Price

Header Line 2:

□□□□□□.....□□□□□□□□□.....□□□□□□□□□□□□□□□□.....

Header Line 3:

(Press FINISHED.)

Screen PG-15

RECORD SELECTION? N

Screen PG-17

DETAIL LINE 1

BLANK LINES BEFORE 0
BLANK LINES AFTER * (conditional field)
PRINT ON CONDITIONS Y

Screen PG-18

CONDITION 1:

IF/AND/OR	(IF)
FIELD 1	CURRENT-PRICE
CONDITION	LT
FIELD 2	ORIGINAL-PRICE

CONDITION 2:

(Press FINISHED.)

Screen PG-19

DETAIL LINE 1 FIELD NUMBER 1

COLUMN NUMBER	70
FIELD OR CONSTANT	"REDUCED"
PRINT FORMAT	X(7)

DETAIL LINE 1 FIELD NUMBER 2

(Press FINISHED.)

Screen PG-17

DETAIL LINE 2

NUMBER BLANK LINES BEFORE	0
NUMBER BLANK LINES AFTER	0
PRINT ON CONDITIONS	N

Screen PG-19

DETAIL LINE 2 FIELD NUMBER 1

COLUMN NUMBER	5
FIELD OR CONSTANT	TOWN
PRINT FORMAT	X(12)

DETAIL LINE 2 FIELD NUMBER 2

COLUMN NUMBER	20
FIELD OR CONSTANT	LISTDATA-KEY
PRINT FORMAT	ZZZZ9
COMPUTATION	↓

DETAIL LINE 2 FIELD NUMBER 3

COLUMN NUMBER	28
FIELD OR CONSTANT	DESCRIPTION
PRINT FORMAT	X(30)

DETAIL LINE 2 FIELD NUMBER 4

COLUMN NUMBER	61
FIELD OR CONSTANT	CURRENT-PRICE
PRINT FORMAT	\$Z,ZZ9,999
COMPUTATION	T

DETAIL LINE 2 FIELD NUMBER 5

(Press FINISHED.)

Screen PG-17

DETAIL LINE 3

(Press FINISHED.)

Screen PG-21

CONTROL BREAK FIELD TOWN
BLANK LINES AFTER 2

Screen PG-22

CONTROL BREAK TOWN TOTAL LINE 1

NUMBER BLANK LINES BEFORE 1

Screen PG-23

TOTAL LINE 1 FIELD 1

COLUMN NUMBER 32
FIELD OR CONSTANT "Total value for this town:"
PRINT FORMAT X(26)

TOTAL LINE 1 FIELD 2

(Press FINISHED.)

Screen PG-22

CONTROL BREAK TOWN TOTAL LINE 2

(Press FINISHED.)

Screen PG-21

(Press FINISHED.)

Screen PG-24

GENERATE THE COBOL CODE Y
OUTPUT FORMAT 1

Screen PG-1

Press FINISHED to return to the Main Menu.

Main Menu

Press FINISHED to exit the PROXI session.

After the PROXI session, the generated code is ready to be compiled. A sample of the intended output appears in Figure C-4. Figure C-6 is the compilation listing for the Report Writer program.

BIRKS & ABERGELDIE REAL ESTATE COMPANY

This report shows the active listings for this office.
For information about old listings, see the Blue Book.

Town	Listing	Description	Price
HAMILTON	84653	2 Bdrm townhouse condo, parking	\$65,000
HAMILTON	38175	Dutch Colonial, 3 Bdrm, 2 Bath	\$83,500
HAMILTON	56721	7 room Ranch, 2 car garage	\$87,500
HAMILTON	34982	Handyman special, 2 acres	\$88,000
HAMILTON	58320	Oliver Colonial, needs work	\$88,000
HAMILTON	87621	Farmhouse, 7 room, new siding	\$115,000 REDUCED
HAMILTON	54994	Mod. Colonial, fine construct.	\$164,900
Total value for this town:			\$691,900
LINCOLN	23887	2 acre lot, quiet area	\$29,000
LINCOLN	34527	Starter home, wood stove	\$52,000
LINCOLN	37421	Antique duplex, restored, gas	\$57,000
LINCOLN	63499	Cape, kingsz master bdrm, acre	\$59,900
LINCOLN	66583	New condo, conven. loc., trans	\$75,500
LINCOLN	32385	Split Entr, 4 Bdrm, fam rm, bar	\$89,900 REDUCED
LINCOLN	67544	Townhse condo, 2 Bdrm, 2 bath	\$99,500
LINCOLN	34551	Contemp carge house, 6 acres	\$220,000
Total value for this town:			\$682,800
NEWBURY	28873	7 acres, perked and ready	\$49,000
NEWBURY	74563	3 Family, o-b-b-4, sep. util.	\$53,500
NEWBURY	34591	Split Level, unique floor plan	\$83,500
NEWBURY	84656	4 Bdrm home, stable, paddock	\$117,000
NEWBURY	12053	Garr. Col., half acre, fireplace	\$118,900
Total value for this town:			\$421,900
NEWBURYPORT	60654	6 room cottage, near beach	\$69,900
NEWBURYPORT	42653	C.E.Col., nice area, needs work	\$74,000 REDUCED
NEWBURYPORT	32878	19th C. home - historic distr	\$83,500
Total value for this town:			\$227,400
PEABODY	34871	Duplex, 2 Bdrm each side	\$53,000
PEABODY	73242	2 Bdrm Ranch, fenced yard	\$53,900
PEABODY	73645	Oversize Cape, private, gas	\$60,500
PEABODY	32745	Stately 8 rm Col., country kit	\$66,000
PEABODY	43973	Cape, 4 Bdrm, oil heat, porch	\$74,900
PEABODY	50488	9 room Ranch, in-law apt	\$86,900
PEABODY	88472	Estate, 9 Bdrms, 3.7 acres	\$195,000
Total value for this town:			\$590,200
ROCKPORT	47301	Country cottage on lake, porch	\$42,900
ROCKPORT	34820	Custom Ranch, move-in cond.	\$72,900 REDUCED
ROCKPORT	33275	Mod. Colonial, 2 Bdrm, fam rm	\$83,500
ROCKPORT	45762	3 Bdrm home, shop w/ sep entr	\$89,500
ROCKPORT	64853	3 Bdrm townhouse condo, pool	\$95,000
Total value for this town:			\$324,800
THOMSON	46183	New Cape, fenced yrd, garage	\$77,800
THOMSON	44385	3 Bdrm Ranch, wooded lot	\$87,500 REDUCED
THOMSON	29483	Condo, 2Bdrm, restored Colon'l	\$89,900
THOMSON	68534	Contemp, center fireplace, gas	\$102,500
THOMSON	34621	Antique Col., restro, 5 acres	\$174,000
Total value for this town:			\$531,700

Figure C-4. Sample Report Produced by the SHOWLIST Program

```
0001
0002      000100
0003      000200 IDENTIFICATION DIVISION.
0004      000300 PROGRAM-ID.          LISTINGS.
0005      000400 AUTHOR.              PROXI  REV 1.10.
0006      000500
0007      000600 ENVIRONMENT DIVISION.
0008      000700 CONFIGURATION SECTION.
0009      000800 SOURCE-COMPUTER. CS-20.
0010      000900*                    CS-30.
0011      001000*                    CS-40.
0012      001100*                    CS-60.
0013      001200 OBJECT-COMPUTER. CS-20.
0014      001300*                    CS-30.
0015      001400*                    CS-40.
0016      001500*                    CS-60.
0017      001600 INPUT-OUTPUT SECTION.
0018      001700 FILE-CONTROL.
0019      001800      COPY "LISTDATA.SL".
0020C     000000
0021C     000100      SELECT LISTDATA
0022C     000200          ASSIGN TO DISK      "LISTDATA"
0023C     000300          ORGANIZATION IS INDEXED
0024C     000400          ACCESS MODE IS DYNAMIC
0025C     000500          RECORD KEY IS LISTDATA-KEY
0026C     000600          FILE STATUS IS LISTDATA-STATUS.
0027C     000700
0028
0029      001900
```

Figure C-5. Compilation Listing of the File Maintenance Program (continues)

```

0030      002000/
0031      002100 DATA DIVISION.
0032      002200 FILE SECTION.
0033      002300     COPY "LISTDATA.FD".
0034C     000100 FD LISTDATA
0035C     000200     RECORDING MODE IS VARIABLE
0036C     000300     LABEL RECORDS ARE OMITTED.
0037C     000400 01 LISTING-RECORD.
0038C     000500     03 LISTDATA-KEY                PIC 9(5).
0039C     000600     03 LIST-DATE                  PIC X(8).
0040C     000700     03 TOWN                        PIC X(12).
0041C     000800     03 DESCRIPTION                PIC X(30).
0042C     000900     03 ORIGINAL-PRICE             PIC 9(6).
0043C     001000     03 CURRENT-PRICE              PIC 9(6).
0044
0045      002400

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0046      002500/
0047      002600 WORKING-STORAGE SECTION.
0048      002700
0049      002800 01 APPLICATION-NAME PIC X(30)
0050      002900 VALUE "Current Listings Update".
0051      003000
0052      003100 01 OC-STATUS PIC 9.
0053C     000100 COPY "LISTDATA.WS".
0054C     000200 01 LISTDATA-STATUS PIC XX.
0055
0056      003200
0057      003300
0058      003400
0059      003500 COPY "PROXISCF01.WS".
0060C     000100
0061C     000200 01 ANY-CHANGE-ANSWER.
0062C     000300 05 ANY-CHANGE-CHAR-1 PIC X.
0063C     000400 05 FILLER PIC X.
0064C     000500 01 ANY-CHANGE-DEFAULT PIC X.
0065C     000600 01 CHANGE-NO PIC 99.
0066C     000700 01 WHAT-NUMBER-ANSWER.
0067C     000800 03 WHAT-NUMBER-CHAR PIC X OCCURS 2 TIMES.
0068C     000900
0069C     001000 01 ESCAPE-CODE PIC 99.
0070C     001100
0071C     001200 01 ESCAPE-KEY PIC 99 VALUE 01.
0072C     001300 01 F1-KEY PIC 99 VALUE 02.
0073C     001400 01 F2-KEY PIC 99 VALUE 03.
0074C     001500 01 F3-KEY PIC 99 VALUE 04.
0075C     001600 01 F7-KEY PIC 99 VALUE 08.
0076C     001700 01 F8-KEY PIC 99 VALUE 09.
0077C     001800 01 END-KEY PIC 99 VALUE 09.
0078C     001900
0079C     002000
0080C     002100 01 FILE-ERROR-STATUS PIC XX.
0081C     002200 01 FILE-ERROR-NAME PIC X(10) VALUE SPACES.
0082C     002300
0083C     002400 01 FILE-STATUS-CODE-TABLE.
0084C     002500 03 RECORD-ON-FILE PIC XX VALUE "00".
0085C     002600 03 I-O-OK PIC XX VALUE "00".
0086C     002700 03 AT-END PIC XX VALUE "10".
0087C     002800 03 INVALID-KEY PIC XX VALUE "21".
0088C     002900 03 DUPLICATE-KEY PIC XX VALUE "22".
0089C     003000 03 RECORD-NOT-FOUND PIC XX VALUE "23".
0090C     003100 03 BOUNDARY-ERROR PIC XX VALUE "24".
0091C     003200 03 HARDWARE-ERROR PIC XX VALUE "30".
0092C     003300 03 DISK-FULL PIC XX VALUE "34".
0093C     003400 03 OPEN-ERROR PIC XX VALUE "91".
0094C     003500 03 CLOSE-ERROR PIC XX VALUE "91".
0095C     003600 03 NO-FILE PIC XX VALUE "91".
0096C     003700 03 MODE-ERROR PIC XX VALUE "92".
0097C     003800 03 RECORD-LOCKED PIC XX VALUE "94".
0098C     003900 03 USE-ERROR PIC XX VALUE "94".
0099C     004000 03 NO-DIRECTORY PIC XX VALUE "96".
0100C     004100 03 OVER-LOCK-LIMIT PIC XX VALUE "97".
0101C     004200 03 NO-CONTIGUOUS-SPACE PIC XX VALUE "98".
0102C     004300 03 FULL-PRINTER-TABLE PIC XX VALUE "99".
0103C     004400
0104C     004410 01 HASH-W PIC XX VALUE "#W".
0105C     004420
0106C     004500 01 MESSAGE-FIELD PIC X(45).
0107C     004600 01 MESSAGE-WAIT-ENTRY PIC X.
0108C     004700

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0109C 004800 01 QUESTION-ANSWER PIC X.
0110C 004900
0111C 005000 01 ULINES PIC X(79) VALUE ALL "_".
0112C 005100 01 PROMPT-LINE PIC X(79) VALUE SPACES.
0113C 005200 01 MENU-SELECTION PIC 9.
0114C 005300 01 SELECTION-TYPE PIC X(14).
0115C 005400
0116C 005410
0117C 005420 01 SYSTEM-DATE.
0118C 005430 03 SYSTEM-YEAR PIC 99 VALUE ZERO.
0119C 005440 03 SYSTEM-MONTH PIC 99 VALUE ZERO.
0120C 005450 03 SYSTEM-DAY PIC 99 VALUE ZERO.
0121C 005460
0122C 005470 01 TODAYS-DATE PIC 999999 VALUE ZERO.
0123C 005472
0124C 005474 01 TODAYS-DATE-REDEFINE REDEFINES TODAYS-DATE.
0125C 005480 03 TODAYS-MONTH PIC 99.
0126C 005490 03 TODAYS-DAY PIC 99.
0127C 005500 03 TODAYS-YEAR PIC 99.
0128C 005600
0129C 005700 01 MONTH-NAMES.
0130C 005800 10 FILLER PIC X(36)
0131C 005900 VALUE "JANFEBMARAPR MAYJUNJUL AUGSEP OCTNOVDEC".
0132C 006000
0133C 006100 01 MONTH-TABLE REDEFINES MONTH-NAMES.
0134C 006200 10 MONTH-NAME PIC XXX OCCURS 12 TIMES.
0135C 006300
0136C 006400 01 SYSTEM-TIME.
0137C 006500 10 SYSTEM-HOUR PIC 99 VALUE ZERO.
0138C 006600 10 SYSTEM-MINUTE PIC 99 VALUE ZERO.
0139C 006700 10 SYSTEM-SECOND PIC 99 VALUE ZERO.
0140C 006800 10 SYSTEM-HUNDRETH PIC 99 VALUE ZERO.
0141

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0142 003600/
0143 003700 SCREEN SECTION.
0144 003800 01 MENU-SCREEN.
0145 003900 05 BLANK SCREEN.
0146 004000 05 LINE 01 COL 01 PIC X(30) FROM APPLICATION-NAME.
0147 004100 05 LINE 08 COL 10 "PLEASE SELECT".
0148 004200 05 LINE 08 COL 25 PIC 9 TO MENU-SELECTION.
0149 004300 05 LINE 10 COL 16 "1. ADD NEW Listings.".
0150 004400 05 LINE 12 COL 16 "2. CHANGE/INQUIRE".
0151 004500 05 LINE 12 COL 34 "Listings.".
0152 004600 05 LINE 14 COL 16 "3. DELETE Listings.".
0153 004700 05 LINE 16 COL 16 "4. PRINT Listings.".
0154 004800
0155 004900 01 END-SCREEN.
0156 005000 05 LINE 08 COL 25 "END".
0157 005100
0158 005200 COPY "LISTSCR.SD".
0159C 000100 01 LISTSCR-FORMAT-SCREEN.
0160C 000200 05 BLANK SCREEN.
0161C 000300 05 LINE 06 COL 09 "*1. Listing Number: ".
0162C 000400 05 LINE 06 COL 46 "2. Date: ".
0163C 000500 05 LINE 09 COL 10 "3. Town: ".
0164C 000600 05 LINE 12 COL 10 "4. Description: ".
0165C 000700 05 LINE 15 COL 10 "5. Original Price: $".
0166C 000800 05 LINE 18 COL 10 "6. Current Price: $".
0167C 000900
0168C 001000 01 LISTSCR-ENTRY-SCREEN.
0169C 001100 05 LISTSCR-01 LINE 06 COL 30 PIC 99999
0170C 001200 TO LISTDATA-KEY.
0171C 001300 05 LISTSCR-02 LINE 06 COL 56 PIC X(8)
0172C 001400 TO LIST-DATE
0173C 001500 REQUIRED FULL
0174C 001600 05 LISTSCR-03 LINE 09 COL 20 PIC X(12)
0175C 001700 TO TOWN
0176C 001800 REQUIRED
0177C 001900 05 LISTSCR-04 LINE 12 COL 27 PIC X(30)
0178C 002000 TO DESCRIPTION
0179C 002100 REQUIRED
0180C 002200 05 LISTSCR-05 LINE 15 COL 31 PIC ZZ9,999
0181C 002300 TO ORIGINAL-PRICE.
0182C 002400 05 LISTSCR-06 LINE 18 COL 31 PIC ZZ9,999
0183C 002500 TO CURRENT-PRICE.
0184C 002600
0185C 002700 01 LISTSCR-END-SCREEN.
0186C 002800 05 LINE 06 COL 30 "END".
0187C 002900 01 LISTSCR-DISPLAY-SCREEN.
0188C 003000 05 LISTSCR-D-01 LINE 06 COL 30 PIC 99999
0189C 003100 USING LISTDATA-KEY.
0190C 003200 05 LISTSCR-D-02 LINE 06 COL 56 PIC X(8)
0191C 003300 USING LIST-DATE.
0192C 003400 05 LISTSCR-D-03 LINE 09 COL 20 PIC X(12)
0193C 003500 USING TOWN.
0194C 003600 05 LISTSCR-D-04 LINE 12 COL 27 PIC X(30)
0195C 003700 USING DESCRIPTION.
0196C 003800 05 LISTSCR-D-05 LINE 15 COL 31 PIC ZZ9,999
0197C 003900 USING ORIGINAL-PRICE.
0198C 004000 05 LISTSCR-D-06 LINE 18 COL 31 PIC ZZ9,999
0199C 004100 USING CURRENT-PRICE.
0200
0201 005300
0202 005400 COPY "PROXISCF01.SD".
0203C 000100
0204C

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)


```

0205C 000200 01 ANY-CHANGE-SCREEN.
0206C
0207C 000300 05 LINE 24 BLANK LINE.
0208C
0209C 000400 05 LINE 24 COL 1 "ANY CHANGE ?".
0210C
0211C 000500 05 LINE 24 COL 15 PIC XX FROM SPACE TO ANY-CHANGE-ANSWER.
0212C
0213C 000600
0214C
0215C 000700 01 ANY-CHANGE-DEFAULT-SCREEN.
0216C
0217C 000800 05 LINE 24 COL 15 PIC X FROM ANY-CHANGE-ANSWER.
0218C
0219C 000900
0220C
0221C 001000 01 FILE-ACCESS-ERROR-SCREEN.
0222C
0223C 001100 05 BLANK SCREEN.
0224C
0225C 001200 05 BLINK LINE 12 COL 20 "E R R O R".
0226C
0227C 001300 05 " . . . FILE ACCESS ERROR: ".
0228C
0229C 001400 05 PIC X(2) FROM FILE-ERROR-STATUS.
0230C
0231C 001500 05 LINE 13 COL 20 "FROM".
0232C
0233C 001600 05 LINE 13 COL 26 PIC X(10) FROM FILE-ERROR-NAME.
0234C
0235C 001700 05 LINE 13 COL 37 "FILE".
0236C
0237C 001800
0238C
0239C 001900 01 MESSAGE-SCREEN.
0240C
0241C 001900 05 ERROR-MESSAGE-LINE.
0242C 002000 10 CLEAR-MESSAGE LINE 24 BLANK LINE.
0243C
0244C 002100 10 LINE 24 COL 5 PIC X(45) FROM MESSAGE-FIELD.
0245C 002110 10 BELL.
0246C
0247C 002200 05 LINE 24 COL 52 "TYPE "CR" TO CONTINUE".
0248C
0249C 002300 05 LINE 24 COL 78 PIC X(1) TO MESSAGE-WAIT-ENTRY.
0250C
0251C
0252C 002500
0253C
0254C 002600 01 PROMPT-SCREEN.
0255C
0256C 002700 10 CLEAR-PROMPT LINE 24 BLANK LINE.
0257C
0258C 002800 10 LINE 24 COL 1 PIC X(79) FROM PROMPT-LINE.
0259C
0260C 002900
0261C
0262C 003000 01 QUESTION-SCREEN.
0263C
0264C 003100 03 LINE 24 COLUMN 01 PIC X(45) FROM MESSAGE-FIELD.
0265C
0266C 003200 03 LINE 24 COLUMN 31 PIC X
0267C

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

0268C	003300	TO QUESTION-ANSWER.
0269C		
0270C	003400	
0271C		
0272C	003500	01 WAIT-PRINTING.
0273C		
0274C	003600	10 BLANK SCREEN.
0275C		
0276C	003700	10 LINE 1 COL 1 PIC X(30) FROM APPLICATION-NAME.
0277C		
0278C	003800	10 LINE 12 COL 20 "PRINTING PLEASE WAIT".
0279C		
0280C	003900	
0281C		
0282C	004000	01 WAIT-PROCESSING.
0283C		
0284C	004100	10 BLANK SCREEN.
0285C		
0286C	004200	10 LINE 1 COL 1 PIC X(30) FROM APPLICATION-NAME.
0287C		
0288C	004300	10 LINE 12 COL 20 "PROCESSING PLEASE WAIT".
0289C		
0290C	004400	
0291C		
0292C	004500	01 WAIT-LINE-24.
0293C		
0294C	004600	10 LINE 24 BLANK LINE.
0295C		
0296C	004700	10 LINE 24 COL 1 "PLEASE WAIT".
0297C		
0298C	004800	
0299C		
0300C	004900	01 WHAT-NUMBER-SCREEN.
0301C		
0302C	005000	05 LINE 24 COL 18 "WHAT NUMBER ?".
0303C		
0304C	005100	05 LINE 24 COL 33 PIC XX FROM SPACES TO WHAT-NUMBER-ANSWER.
0305C		
0306		

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0307 005500/
0308 005600 PROCEDURE DIVISION.
0309 005700 DECLARATIVES.
0310 005800     COPY "LISTDATA.DS".
0311C 000100 LISTDATA-ERROR SECTION.
0312C 000200     USE AFTER ERROR PROCEDURE ON LISTDATA.
0313C 000300 PROCESS-LISTDATA-ERROR.
0314C 000400     IF LISTDATA-STATUS = HARDWARE-ERROR OR
0315C 000500     LISTDATA-STATUS = OPEN-ERROR OR
0316C 000600     LISTDATA-STATUS = DISK-FULL OR
0317C 000700     LISTDATA-STATUS = OVER-LOCK-LIMIT
0318C 000800     MOVE LISTDATA-STATUS TO FILE-ERROR-STATUS
0319C 000900     MOVE "LISTDATA" TO FILE-ERROR-NAME
0320C 001000     DISPLAY FILE-ACCESS-ERROR-SCREEN
0321C 001100     STOP RUN.
0322
0323 005900
0324 006000 EXIT-DECLARATIVES. EXIT.
0325 006100 END DECLARATIVES.

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0326 006200/
0327 006300* ***** LEVEL 1 *****
0328 006400
0329 006500 MAIN-LOGIC SECTION.
0330 006600 BEGIN.
0331 006700 PERFORM OPEN-LISTDATA.
0332 006800 IF LISTDATA-STATUS IS NOT = I-O-OK
0333 006900 GO TO ERROR-PROGRAM.
0334 007000 MOVE LOW-VALUES TO LISTDATA-KEY.
0335 007100
0336 007200
0337 007300 PERFORM GET-DATES.
0338 007400
0339 007500 MOVE "N" TO ANY-CHANGE-ANSWER.
0340 007600
0341 007700 COPY "PROXISCF02.PL".
0342C 000100
0343C 000200 MAINTENANCE-MENU.
0344C 000300 DISPLAY MENU-SCREEN.
0345C 000400 ACCEPT MENU-SCREEN.
0346C 000500 ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0347C 000600 IF ESCAPE-CODE = ESCAPE-KEY
0348C 000700 GO TO MAINTENANCE-MENU.
0349C 000800 IF ESCAPE-CODE = END-KEY
0350C 000900 DISPLAY END-SCREEN
0351C 001000 PERFORM END-OF-PROGRAM.
0352C 001100 MOVE SPACES TO SELECTION-TYPE.
0353C 001200 IF MENU-SELECTION IS = 1
0354C 001300 MOVE "ADD" TO SELECTION-TYPE
0355C 001400 PERFORM ADD-RECORDS.
0356C 001500 IF MENU-SELECTION IS = 2
0357C 001600 MOVE "CHANGE/INQUIRE" TO SELECTION-TYPE
0358C 001700 PERFORM CHANGE-RECORDS.
0359C 001800 IF MENU-SELECTION IS = 3
0360C 001900 MOVE "DELETE" TO SELECTION-TYPE
0361C 002000 PERFORM DELETE-RECORDS.
0362C 002100 IF MENU-SELECTION IS = 4
0363C 002200 MOVE "PRINT-OUT" TO SELECTION-TYPE
0364C 002300 PERFORM PRINT-RECORDS.
0365C 002400 GO TO MAINTENANCE-MENU.
0366

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0367 007800/
0368 007900* ***** LEVEL 2 *****
0369 008000
0370 008100 ADD-RECORDS SECTION.
0371 008200 ADD-RECORDS-BEGIN.
0372 008300 DISPLAY LISTSCR-FORMAT-SCREEN.
0373 008400 PERFORM LISTSCR-ENTER-KEY.
0374 008500 IF ESCAPE-CODE = ESCAPE-KEY
0375 008600 OR ESCAPE-CODE = F1-KEY
0376 008700 GO TO ADD-RECORDS-BEGIN.
0377 008800 IF ESCAPE-CODE = END-KEY
0378 008900 GO TO ADD-RECORDS-EXIT.
0379 009000 PERFORM VERIFY-NEW-LISTDATA.
0380 009100 IF LISTDATA-STATUS IS = RECORD-ON-FILE
0381 009200 GO TO ADD-RECORDS-BEGIN.
0382 009300 PERFORM LISTSCR-ENTER.
0383 009400 IF ESCAPE-CODE = ESCAPE-KEY
0384 009500 PERFORM NOT-PROCESSED
0385 009600 GO TO ADD-RECORDS-BEGIN.
0386 009700 PERFORM LISTSCR-ANY-CHANGE.
0387 009800 IF ESCAPE-CODE = ESCAPE-KEY
0388 009900 PERFORM NOT-PROCESSED
0389 010000 GO TO ADD-RECORDS-BEGIN.
0390 010100 PERFORM ADD-LISTDATA.
0391 010200 GO TO ADD-RECORDS-BEGIN.
0392 010300 ADD-RECORDS-EXIT. EXIT.
0393 010400
0394 010500 CHANGE-RECORDS SECTION.
0395 010600 CHANGE-RECORDS-START-FILE.
0396 010700 START LISTDATA KEY GREATER THAN
0397 010800 LISTDATA-KEY
0398 010900 INVALID KEY GO TO CHANGE-RECORDS-BEGIN.
0399 011000 CHANGE-RECORDS-BEGIN.
0400 011100 UNLOCK LISTDATA.
0401 011200 DISPLAY LISTSCR-FORMAT-SCREEN.
0402 011300 PERFORM LISTSCR-ENTER-KEY.
0403 011400 IF ESCAPE-CODE = ESCAPE-KEY
0404 011500 GO TO CHANGE-RECORDS-BEGIN.
0405 011600 IF ESCAPE-CODE = END-KEY
0406 011700 GO TO CHANGE-RECORDS-EXIT.
0407 011800 IF ESCAPE-CODE = F1-KEY
0408 011900 PERFORM NEXT-LISTDATA-RECORD
0409 012000 ELSE
0410 012100 PERFORM READ-LISTDATA-RECORD.
0411 012200 IF LISTDATA-STATUS IS NOT = I-O-OK
0412 012300 - GO TO CHANGE-RECORDS-START-FILE.
0413 012400 DISPLAY LISTSCR-DISPLAY-SCREEN.
0414 012500 PERFORM LISTSCR-ANY-CHANGE.
0415 012600 IF ESCAPE-CODE = ESCAPE-KEY
0416 012700 PERFORM NOT-PROCESSED
0417 012800 GO TO CHANGE-RECORDS-BEGIN.
0418 012900 PERFORM CHANGE-LISTDATA-RECORD.
0419 013000 GO TO CHANGE-RECORDS-BEGIN.
0420 013100 CHANGE-RECORDS-EXIT. EXIT.
0421 013200
0422 013300 DELETE-RECORDS SECTION.
0423 013400 DELETE-RECORDS-START-FILE.
0424 013500 START LISTDATA KEY GREATER THAN
0425 013600 LISTDATA-KEY
0426 013700 INVALID KEY GO TO DELETE-RECORDS-BEGIN.
0427 013800 DELETE-RECORDS-BEGIN.
0428 013900 UNLOCK LISTDATA.
0429 014000 DISPLAY LISTSCR-FORMAT-SCREEN.

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0430 014100 PERFORM LISTSCR-ENTER-KEY.
0431 014200 IF ESCAPE-CODE = ESCAPE-KEY
0432 014300 GO TO DELETE-RECORDS-BEGIN.
0433 014400 IF ESCAPE-CODE = END-KEY
0434 014500 GO TO DELETE-RECORDS-EXIT.
0435 014600 IF ESCAPE-CODE = F1-KEY
0436 014700 PERFORM NEXT-LISTDATA-RECORD
0437 014800 ELSE
0438 014900 PERFORM READ-LISTDATA-RECORD.
0439 015000 IF LISTDATA-STATUS IS NOT = I-O-OK
0440 015100 GO TO DELETE-RECORDS-START-FILE.
0441 015200 DISPLAY LISTSCR-DISPLAY-SCREEN.
0442 015300 PERFORM VERIFY-LISTDATA.
0443 015400 IF QUESTION-ANSWER IS = "N"
0444 015500 GO TO DELETE-RECORDS-BEGIN.
0445 015600 PERFORM DELETE-LISTDATA-RECORD.
0446 015700 GO TO DELETE-RECORDS-BEGIN.
0447 015800 DELETE-RECORDS-EXIT. EXIT.
0448 015900
0449 016000 PRINT-RECORDS SECTION.
0450 016100 PRINT-RECORDS-BEGIN.
0451 016200 DISPLAY WAIT-LINE-24.
0452 016300 PERFORM CLOSE-FILES.
0453 016400 CALL PROGRAM "SHOWLIST".
0454 016500 PRINT-RECORDS-EXIT. EXIT.
0455 016600
0456 016700 END-OF-PROGRAM SECTION.
0457 016800 END-OF-PROGRAM-BEGIN.
0458 016900 DISPLAY WAIT-LINE-24.
0459 017000 PERFORM CLOSE-FILES.
0460 017100 CALL PROGRAM "LOGON".
0461 017200 STOP RUN.
0462 017300 END-OF-PROGRAM-EXIT. EXIT.
0463 017400
0464 017500 ERROR-PROGRAM SECTION.
0465 017600 ERROR-PROGRAM-BEGIN.
0466 017700 DISPLAY WAIT-LINE-24.
0467 017800 CALL PROGRAM "LOGON".
0468 017900 STOP RUN.
0469 018000 ERROR-PROGRAM-EXIT. EXIT.

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0470 018100/
0471 018200* ***** LEVEL 3 *****
0472 018300
0473 018400 COPY "LISTSCR.PL".
0474C 000100 LISTSCR-ENTER-KEY SECTION.
0475C 000200
0476C 000300 ENTER-KEY.
0477C 000400
0478C 000500 PERFORM LISTSCR-01-ENTRY.
0479C 000600
0480C 000700 IF ESCAPE-CODE = ESCAPE-KEY
0481C 000800 GO TO LISTSCR-ENTER-KEY-EXIT.
0482C 000900 IF ESCAPE-CODE = F1-KEY
0483C 001000 GO TO LISTSCR-ENTER-KEY-EXIT.
0484C 001100 IF ESCAPE-CODE = END-KEY
0485C 001200 DISPLAY LISTSCR-END-SCREEN,
0486C 001300 GO TO LISTSCR-ENTER-KEY-EXIT.
0487C 001400
0488C 001500 LISTSCR-ENTER-KEY-EXIT. EXIT.
0489C 001600
0490C 001700 LISTSCR-ENTER SECTION.
0491C 001800
0492C 001900 LISTSCR-ENTER-DATA.
0493C 002000
0494C 002100 PERFORM LISTSCR-02-ENTRY.
0495C 002200
0496C 002300 IF ESCAPE-CODE NOT = ESCAPE-KEY
0497C 002400 PERFORM LISTSCR-03-ENTRY.
0498C 002500
0499C 002600 IF ESCAPE-CODE NOT = ESCAPE-KEY
0500C 002700 PERFORM LISTSCR-04-ENTRY.
0501C 002800
0502C 002900 IF ESCAPE-CODE NOT = ESCAPE-KEY
0503C 003000 PERFORM LISTSCR-05-ENTRY.
0504C 003100
0505C 003200 IF ESCAPE-CODE NOT = ESCAPE-KEY
0506C 003300 PERFORM LISTSCR-06-ENTRY.
0507C 003400
0508C 003500 LISTSCR-ENTER-EXIT. EXIT.
0509C 003600
0510C 003700 LISTSCR-ANY-CHANGE SECTION.
0511C 003800
0512C 003900 LISTSCR-ANY-QUESTION.
0513C 004000 MOVE "N" TO ANY-CHANGE-DEFAULT.
0514C 004100 PERFORM ANY-CHANGE.
0515C 004200 IF ESCAPE-CODE = ESCAPE-KEY OR
0516C 004300 ANY-CHANGE-ANSWER = "N"
0517C 004400 GO TO LISTSCR-ANY-CHANGE-EXIT.
0518C 004500
0519C 004600 LISTSCR-CHANGE-SELECTION.
0520C 004700
0521C 004800 PERFORM WHAT-NUMBER.
0522C 004900 IF ESCAPE-CODE = ESCAPE-KEY
0523C 005000 GO TO LISTSCR-ANY-QUESTION.
0524C 005100 IF CHANGE-NO < 01 OR CHANGE-NO > 06
0525C 005200 GO TO LISTSCR-CHANGE-SELECTION.
0526C 005300
0527C 005400 GO TO
0528C 005500 CHANGE-LISTSCR-01
0529C 005600 , CHANGE-LISTSCR-02
0530C 005700 , CHANGE-LISTSCR-03
0531C 005800 , CHANGE-LISTSCR-04
0532C 005900 , CHANGE-LISTSCR-05

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0533C 006000           , CHANGE-LISTSCR-06
0534C 006100           DEPENDING ON CHANGE-NO.
0535C 006200
0536C 006300           GO TO LISTSCR-CHANGE-SELECTION.
0537C 006400
0538C 006500 CHANGE-LISTSCR-01.
0539C 006600           PERFORM LISTSCR-NO-CHANGE.
0540C 006700           GO TO LISTSCR-ANY-QUESTION.
0541C 006800
0542C 006900 CHANGE-LISTSCR-02.
0543C 007000           PERFORM LISTSCR-02-ENTRY.
0544C 007100           GO TO LISTSCR-ANY-CHANGE-END.
0545C 007200
0546C 007300 CHANGE-LISTSCR-03.
0547C 007400           PERFORM LISTSCR-03-ENTRY.
0548C 007500           GO TO LISTSCR-ANY-CHANGE-END.
0549C 007600
0550C 007700 CHANGE-LISTSCR-04.
0551C 007800           PERFORM LISTSCR-04-ENTRY.
0552C 007900           GO TO LISTSCR-ANY-CHANGE-END.
0553C 008000
0554C 008100 CHANGE-LISTSCR-05.
0555C 008200           PERFORM LISTSCR-05-ENTRY.
0556C 008300           GO TO LISTSCR-ANY-CHANGE-END.
0557C 008400
0558C 008500 CHANGE-LISTSCR-06.
0559C 008600           PERFORM LISTSCR-06-ENTRY.
0560C 008700           GO TO LISTSCR-ANY-CHANGE-END.
0561C 008800 LISTSCR-ANY-CHANGE-END.
0562C 008900           IF ESCAPE-CODE NOT = ESCAPE-KEY
0563C 009000           GO TO LISTSCR-ANY-QUESTION.
0564C 009100
0565C 009200 LISTSCR-ANY-CHANGE-EXIT.
0566C 009300
0567C 009400 LISTSCR-NO-CHANGE SECTION.
0568C 009500 NO-CHANGE.
0569C 009600           MOVE "CHANGE NOT ALLOWED TO THIS FIELD"
0570C 009700           TO MESSAGE-FIELD.
0571C 009800           PERFORM DISPLAY-MESSAGE.
0572C 009900
0573C 010000 LISTSCR-ENTRY-LOGIC SECTION.
0574C 010100
0575C 010200 LISTSCR-01-ENTRY.
0576C 010300           MOVE "Check the Red Book before assigning a new number."
0577C 010400           TO PROMPT-LINE, DISPLAY PROMPT-SCREEN.
0578C 010500           DISPLAY LISTSCR-01 , ACCEPT LISTSCR-01 .
0579C 010600           ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0580C 010700           DISPLAY LISTSCR-D-01 .
0581C 010800           DISPLAY CLEAR-PROMPT.
0582C 010900
0583C 011000 LISTSCR-02-ENTRY.
0584C 011100           MOVE "Use dd/mm/yy format."
0585C 011200           TO PROMPT-LINE, DISPLAY PROMPT-SCREEN.
0586C 011300           DISPLAY LISTSCR-02 , ACCEPT LISTSCR-02 .
0587C 011400           ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0588C 011500           DISPLAY LISTSCR-D-02 .
0589C 011600           DISPLAY CLEAR-PROMPT.
0590C 011700
0591C 011800 LISTSCR-03-ENTRY.
0592C 011900           DISPLAY LISTSCR-03 , ACCEPT LISTSCR-03 .
0593C 012000           ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0594C 012100           DISPLAY LISTSCR-D-03 .
0595C 012200

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)


```

0596C 012300 LISTSCR-04-ENTRY.
0597C 012400     MOVE "Be sure to include the type of heating fuel."
0598C 012500         TO PROMPT-LINE, DISPLAY PROMPT-SCREEN.
0599C 012600     DISPLAY LISTSCR-04 , ACCEPT LISTSCR-04 .
0600C 012700     ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0601C 012800         DISPLAY LISTSCR-D-04 .
0602C 012900         DISPLAY CLEAR-PROMPT.
0603C 013000
0604C 013100 LISTSCR-05-ENTRY.
0605C 013200     DISPLAY LISTSCR-05 , ACCEPT LISTSCR-05 .
0606C 013300     ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0607C 013400         DISPLAY LISTSCR-D-05 .
0608C 013500
0609C 013600 LISTSCR-06-ENTRY.
0610C 013700     MOVE "Default value is the original price."
0611C 013800         TO PROMPT-LINE, DISPLAY PROMPT-SCREEN.
0612C 013900     DISPLAY LISTSCR-06 , ACCEPT LISTSCR-06 .
0613C 014000     ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0614C 014100     IF ESCAPE-CODE NOT = ESCAPE-KEY
0615C 014200         IF CURRENT-PRICE = ZERO
0616C 014300             MOVE
0617C 014400                 ORIGINAL-PRICE
0618C 014500                     TO CURRENT-PRICE
0619C 014600     DISPLAY LISTSCR-D-06 .
0620C 014700     DISPLAY CLEAR-PROMPT.
0621C 014800
0622C 014900 LISTSCR-VALIDATE SECTION.
0623
0624 018500

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0625 018600/
0626 018700* ***** INPUT/OUTPUT ROUTINES *****
0627 018800
0628 018900 I-O-SECTION SECTION.
0629 019000 OPEN-LISTDATA.
0630 019100 OPEN I-O LISTDATA.
0631 019200 IF LISTDATA-STATUS = USE-ERROR
0632 019300 PERFORM FILE-NOT-AVAILABLE.
0633 019400 VERIFY-NEW-LISTDATA.
0634 019500 READ LISTDATA RECORD,
0635 019600 IF LISTDATA-STATUS = RECORD-ON-FILE
0636 019700 OR LISTDATA-STATUS = RECORD-LOCKED
0637 019800 PERFORM ALREADY-ON-FILE.
0638 019900 ADD-LISTDATA.
0639 020000 WRITE LISTING-RECORD
0640 020100 INVALID KEY PERFORM ALREADY-ON-FILE.
0641 020200 IF LISTDATA-STATUS = RECORD-LOCKED
0642 020300 PERFORM ALREADY-ON-FILE.
0643 020400 READ-LISTDATA-RECORD.
0644 020500 READ LISTDATA RECORD LOCK
0645 020600 INVALID KEY PERFORM NOT-ON-FILE.
0646 020700 IF LISTDATA-STATUS = RECORD-LOCKED
0647 020800 PERFORM LISTDATA-IN-USE.
0648 020900 NEXT-LISTDATA-RECORD.
0649 021000 READ LISTDATA NEXT RECORD LOCK
0650 021100 AT END PERFORM NOT-ON-FILE.
0651 021200 IF LISTDATA-STATUS = RECORD-LOCKED
0652 021300 PERFORM LISTDATA-IN-USE.
0653 021400 CHANGE-LISTDATA-RECORD.
0654 021500 REWRITE LISTING-RECORD.
0655 021600 UNLOCK LISTDATA.
0656 021700 DELETE-LISTDATA-RECORD.
0657 021800 DELETE LISTDATA RECORD.
0658 021900 IF LISTDATA-STATUS IS = I-O-OK
0659 022000 PERFORM LISTDATA-DELETED.
0660 022100 UNLOCK LISTDATA.
0661 022200
0662 022300
0663 022400 CLOSE-FILES.
0664 022500 CLOSE LISTDATA.
0665 022600

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0666 022700/
0667 022800* ***** MESSAGES *****
0668 022900
0669 023000 MESSAGES SECTION.
0670 023100 ALREADY-ON-FILE.
0671 023200 MOVE "ERROR: Record already on file." TO MESSAGE-FIELD.
0672 023300 PERFORM DISPLAY-MESSAGE.
0673 023400 CHANGE-NOT-ALLOWED.
0674 023500 MOVE "CHANGES NOT ALLOWED TO THESE FIELDS"
0675 023600 TO MESSAGE-FIELD.
0676 023700 PERFORM DISPLAY-MESSAGE.
0677 023800 LISTDATA-DELETED.
0678 023900 MOVE "LISTDATA RECORD DELETED" TO MESSAGE-FIELD.
0679 024000 PERFORM DISPLAY-MESSAGE.
0680 024100 LISTDATA-IN-USE.
0681 024200 MOVE "LISTDATA RECORD IN USE" TO MESSAGE-FIELD.
0682 024300 PERFORM DISPLAY-MESSAGE.
0683 024400 FILE-NOT-AVAILABLE.
0684 024500 MOVE "LISTDATA FILE IS IN USE" TO MESSAGE-FIELD.
0685 024600 PERFORM DISPLAY-MESSAGE.
0686 024700 NOT-ON-FILE.
0687 024800 MOVE "RECORD NOT ON FILE" TO MESSAGE-FIELD.
0688 024900 PERFORM DISPLAY-MESSAGE.
0689 025000 NOT-PROCESSED.
0690 025100 MOVE "NOT PROCESSED" TO MESSAGE-FIELD.
0691 025200 PERFORM DISPLAY-MESSAGE.
0692 025300 VERIFY-LISTDATA.
0693 025400 MOVE "IS THIS THE RIGHT RECORD ?" TO MESSAGE-FIELD.
0694 025500 PERFORM ASK-QUESTION.
0695 025600

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0696      025700/
0697      025800* *****      UTILITY ROUTINES      *****
0698      025900
0699      026000 UTILITY-ROUTINES SECTION.
0700      026100      COPY "PROXISCF01.PL".
0701C     010000
0702C
0703C     010010 ANY-CHANGE.
0704C
0705C     010020      DISPLAY ANY-CHANGE-SCREEN.
0706C
0707C     010030      ACCEPT ANY-CHANGE-SCREEN.
0708C
0709C     010040      ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0710C
0711C     010050      IF ESCAPE-CODE IS NOT = ESCAPE-KEY
0712C
0713C     010060          IF ANY-CHANGE-ANSWER = SPACE
0714C
0715C     010070              MOVE ANY-CHANGE-DEFAULT TO ANY-CHANGE-ANSWER
0716C
0717C     010080              DISPLAY ANY-CHANGE-DEFAULT-SCREEN.
0718C
0719C     010090      IF ESCAPE-CODE IS NOT = ESCAPE-KEY
0720C
0721C     010100          IF ANY-CHANGE-CHAR-1 NUMERIC
0722C
0723C     010110              MOVE ANY-CHANGE-ANSWER TO WHAT-NUMBER-ANSWER
0724C
0725C     010120              MOVE "Y" TO ANY-CHANGE-ANSWER
0726C
0727C     010130          ELSE
0728C
0729C     010140              MOVE SPACE TO WHAT-NUMBER-ANSWER
0730C
0731C     010150          IF (ANY-CHANGE-ANSWER = SPACE) OR
0732C
0733C     010160              (ANY-CHANGE-ANSWER IS NOT = "Y" AND
0734C
0735C     010170              ANY-CHANGE-ANSWER IS NOT = "N")
0736C
0737C     010180              GO TO ANY-CHANGE.
0738C
0739C     010190 DISPLAY-MESSAGE.
0740C
0741C     010200      DISPLAY MESSAGE-SCREEN.
0742C
0743C     010210      ACCEPT MESSAGE-SCREEN.
0744C
0745C     010220      DISPLAY CLEAR-MESSAGE.
0746C
0747C     010230
0748C
0749C     010240 ASK-QUESTION.
0750C
0751C     010250      DISPLAY CLEAR-MESSAGE.
0752C
0753C     010260      DISPLAY QUESTION-SCREEN.
0754C
0755C     010270      ACCEPT QUESTION-SCREEN.
0756C
0757C     010280      IF QUESTION-ANSWER IS NOT = "Y" AND
0758C

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

```

0759C 010290      QUESTION-ANSWER IS NOT = "N"
0760C
0761C 010300      GO TO ASK-QUESTION.
0762C
0763C 010310
0764C
0765C 010320      DISPLAY CLEAR-MESSAGE.
0766C
0767C 010330
0768C
0769C 010340 WHAT-NUMBER.
0770C
0771C 010350      IF WHAT-NUMBER-ANSWER = SPACE
0772C
0773C 010360      DISPLAY WHAT-NUMBER-SCREEN
0774C
0775C 010370      ACCEPT WHAT-NUMBER-SCREEN
0776C
0777C 010380      ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0778C
0779C 010390      IF ESCAPE-CODE IS NOT = ESCAPE-KEY
0780C
0781C 010400      INSPECT WHAT-NUMBER-ANSWER
0782C
0783C 010410      REPLACING LEADING SPACE BY ZEROS
0784C
0785C 010420      IF WHAT-NUMBER-CHAR (2) = SPACE
0786C
0787C 010430      MOVE WHAT-NUMBER-CHAR (1) TO WHAT-NUMBER-CHAR (2)
0788C
0789C 010440      MOVE ZERO TO WHAT-NUMBER-CHAR (1).
0790C
0791C 010450      IF ESCAPE-CODE IS NOT = ESCAPE-KEY
0792C
0793C 010460      IF WHAT-NUMBER-ANSWER NOT NUMERIC
0794C
0795C 010470      MOVE SPACE TO WHAT-NUMBER-ANSWER
0796C
0797C 010480      GO TO WHAT-NUMBER
0798C
0799C 010490      ELSE
0800C
0801C 010500      MOVE WHAT-NUMBER-ANSWER TO CHANGE-NO
0802C
0803C 010510      MOVE SPACE TO WHAT-NUMBER-ANSWER.
0804C
0805C 010520
0806C
0807C 010530 GET-DATES.
0808C
0809C 010540      ACCEPT SYSTEM-DATE FROM DATE.
0810C
0811C 010550      MOVE SYSTEM-DAY TO TODAYS-DAY.
0812C
0813C 010560      MOVE SYSTEM-MONTH TO TODAYS-MONTH.
0814C
0815C 010570      MOVE SYSTEM-YEAR TO TODAYS-YEAR.
0816C
0817
CROSS REFERENCE:
01          524
06          524

```

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

1	353	787	789					
2	356	785	787					
3	359							
4	362							
ADD-LISTDATA	390	638						
ADD-RECORDS	355	370						
ADD-RECORDS-BEGIN								
371	376	381	385	389	391			
ADD-RECORDS-EXIT								
378	392							
ALREADY-ON-FILE								
637	640	642	670					
ANY-CHANGE	514	703	737					
ANY-CHANGE-ANSWER								
61	211	217	339	516	713	715	723	
725	731	733	735					
ANY-CHANGE-CHAR-1								
62	721							
ANY-CHANGE-DEFAULT								
64	513	715						
ANY-CHANGE-DEFAULT-SCREEN								
215	717							
ANY-CHANGE-SCREEN								
205	705	707						
APPLICATION-NAME								
49	146	276	286					
ASK-QUESTION	694	749	761					
AT-END	86							
BEGIN	330							
BOUNDARY-ERROR	90							
CHANGE-LISTDATA-RECORD								
418	653							
CHANGE-LISTSCR-01								
528	538							
CHANGE-LISTSCR-02								
529	542							
CHANGE-LISTSCR-03								
530	546							
CHANGE-LISTSCR-04								
531	550							
CHANGE-LISTSCR-05								
532	554							
CHANGE-LISTSCR-06								
533	558							
CHANGE-NO	65	524	524	534	801			
CHANGE-NOT-ALLOWED								
673								
CHANGE-RECORDS	358	394						
CHANGE-RECORDS-BEGIN								
398	399	404	417	419				
CHANGE-RECORDS-EXIT								
406	420							
CHANGE-RECORDS-START-FILE								
395	412							
CLEAR-MESSAGE	242	745	751	765				
CLEAR-PROMPT	256	581	589	602	620			
CLOSE-ERROR	94							
CLOSE-FILES	452	459	663					
CS-20	9	13						
CURRENT-PRICE	43	183	199	615	618			
DELETE-LISTDATA-RECORD								
445	656							
DELETE-RECORDS	361	422						

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

DELETE-RECORDS-BEGIN	426	427	432	444	446			
DELETE-RECORDS-EXIT	434	447						
DELETE-RECORDS-START-FILE	423	440						
DESCRIPTION	41	178	195					
DISK-FULL	92	316						
DISPLAY-MESSAGE	571	672	676	679	682	685	688	691
	739							
DUPLICATE-KEY	88							
END-KEY	77	349	377	405	433	484		
END-OF-PROGRAM	351	456						
END-OF-PROGRAM-BEGIN	457							
END-OF-PROGRAM-EXIT	462							
END-SCREEN	155	350						
ENTER-KEY	476							
ERROR-MESSAGE-LINE	241							
ERROR-PROGRAM	333	464						
ERROR-PROGRAM-BEGIN	465							
ERROR-PROGRAM-EXIT	469							
ESCAPE-CODE	69	346	347	349	374	375	377	383
	387	403	405	407	415	431	433	435
	480	482	484	496	499	502	505	515
	522	562	579	587	593	600	606	613
	614	709	711	719	777	779	791	
ESCAPE-KEY	71	347	374	383	387	403	415	431
	480	496	499	502	505	515	522	562
	614	711	719	779	791			
EXIT-DECLARATIVES	324							
F1-KEY	72	375	407	435	482			
F2-KEY	73							
F3-KEY	74							
F7-KEY	75							
F8-KEY	76							
FILE-ACCESS-ERROR-SCREEN	221	320						
FILE-ERROR-NAME	81	233	319					
FILE-ERROR-STATUS	80	229	318					
FILE-NOT-AVAILABLE	632	683						
FILE-STATUS-CODE-TABLE	83							
FULL-PRINTER-TABLE	102							
GET-DATES	337	807						
HARDWARE-ERROR	91	314						
HASH-W	104							
I-O-OK	85	332	411	439	658			
I-O-SECTION	628							
INVALID-KEY	87							
LIST-DATE	39	172	191					
LISTDATA	21	34	312	396	400	424	428	630
	634	644	649	655	657	660	664	

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

LISTDATA-DELETED	659	677						
LISTDATA-ERROR	311							
LISTDATA-IN-USE	647	652	680					
LISTDATA-KEY	25	38	170	189	334	397	425	
LISTDATA-STATUS	26	53	314	315	316	317	318	332
	380	411	439	631	635	636	641	646
	651	658						
LISTING-RECORD	37	639	654					
LISTSCR-01	169	578	578					
LISTSCR-01-ENTRY	478	575						
LISTSCR-02	171	586	586					
LISTSCR-02-ENTRY	494	543	583					
LISTSCR-03	174	592	592					
LISTSCR-03-ENTRY	497	547	591					
LISTSCR-04	177	599	599					
LISTSCR-04-ENTRY	500	551	596					
LISTSCR-05	180	605	605					
LISTSCR-05-ENTRY	503	555	604					
LISTSCR-06	182	612	612					
LISTSCR-06-ENTRY	506	559	609					
LISTSCR-ANY-CHANGE	386	414	510					
LISTSCR-ANY-CHANGE-END	544	548	552	556	560	561		
LISTSCR-ANY-CHANGE-EXIT	517	565						
LISTSCR-ANY-QUESTION	512	523	540	563				
LISTSCR-CHANGE-SELECTION	519	525	536					
LISTSCR-D-01	188	580						
LISTSCR-D-02	190	588						
LISTSCR-D-03	192	594						
LISTSCR-D-04	194	601						
LISTSCR-D-05	196	607						
LISTSCR-D-06	198	619						
LISTSCR-DISPLAY-SCREEN	187	413	441					
LISTSCR-END-SCREEN	185	485						
LISTSCR-ENTER	382	490						
LISTSCR-ENTER-DATA	492							
LISTSCR-ENTER-EXIT	508							
LISTSCR-ENTER-KEY	373	402	430	474				
LISTSCR-ENTER-KEY-EXIT	481	483	486	488				
LISTSCR-ENTRY-LOGIC	573							
LISTSCR-ENTRY-SCREEN	168							
LISTSCR-FORMAT-SCREEN								

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

	159	372	401	429				
LISTSCR-NO-CHANGE	539	567						
LISTSCR-VALIDATE	622							
MAIN-LOGIC	329							
MAINTENANCE-MENU	343	348	365					
MENU-SCREEN	144	344	345					
MENU-SELECTION	113	148	353	356	359	362		
MESSAGE-FIELD	106	244	264	570	671	675	678	681
	684	687	690	693				
MESSAGE-SCREEN	239	741	743					
MESSAGE-WAIT-ENTRY	107	249						
MESSAGES	669							
MODE-ERROR	96							
MONTH-NAME	134							
MONTH-NAMES	129	133						
MONTH-TABLE	133							
NEXT-LISTDATA-RECORD	408	436	648					
NO-CHANGE	568							
NO-CONTIGUOUS-SPACE	101							
NO-DIRECTORY	99							
NO-FILE	95							
NOT-ON-FILE	645	650	686					
NOT-PROCESSED	384	388	416	689				
OC-STATUS	51							
OPEN-ERROR	93	315						
OPEN-LISTDATA	331	629						
ORIGINAL-PRICE	42	181	197	617				
OVER-LOCK-LIMIT	100	317						
PRINT-RECORDS	364	449						
PRINT-RECORDS-BEGIN	450							
PRINT-RECORDS-EXIT	454							
PROCESS-LISTDATA-ERROR	313							
PROMPT-LINE	112	258	577	585	598	611		
PROMPT-SCREEN	254	577	585	598	611			
QUESTION-ANSWER	109	268	443	757	759			
QUESTION-SCREEN	262	753	755					
READ-LISTDATA-RECORD	410	438	643					
RECORD-LOCKED	97	636	641	646	651			
RECORD-NOT-FOUND	89							
RECORD-ON-FILE	84	380	635					
SELECTION-TYPE	114	352	354	357	360	363		
SYSTEM-DATE	117	809						
SYSTEM-DAY	120	811						
SYSTEM-HOUR	137							
SYSTEM-HUNDRETH	140							
SYSTEM-MINUTE	138							
SYSTEM-MONTH	119	813						
SYSTEM-SECOND	139							

Figure C-5. Compilation Listing of the File Maintenance Program (continued)

SYSTEM-TIME	136						
SYSTEM-YEAR	118	815					
TODAYS-DATE	122	124					
TODAYS-DATE-REDEFINE							
	124						
TODAYS-DAY	126	811					
TODAYS-MONTH	125	813					
TODAYS-YEAR	127	815					
TOWN	40	175	193				
ULINES	111						
USE-ERROR	98	631					
UTILITY-ROUTINES							
	699						
VERIFY-LISTDATA							
	442	692					
VERIFY-NEW-LISTDATA							
	379	633					
WAIT-LINE-24	292	451	458	466			
WAIT-PRINTING	272						
WAIT-PROCESSING							
	282						
WHAT-NUMBER	521	769	797				
WHAT-NUMBER-ANSWER							
	66	304	723	729	771	781	793
	801	803					795
WHAT-NUMBER-CHAR							
	67	785	787	787	789		
WHAT-NUMBER-SCREEN							
	300	773	775				

Figure C-5. Compilation Listing of the File Maintenance Program (concluded)

```

0001 000100/
0002 000200
0003 000300
0004 000400 IDENTIFICATION DIVISION.
0005 000500 PROGRAM-ID.          SHOWLIST.
0006 000600 AUTHOR.                      PROXI REV 1.10.
0007 000700
0008 000800 ENVIRONMENT DIVISION.
0009 000900 CONFIGURATION SECTION.
0010 001000 SOURCE-COMPUTER. CS-20.
0011 001100*                CS-30.
0012 001200*                CS-40.
0013 001300*                CS-60.
0014 001400 OBJECT-COMPUTER. CS-20.
0015 001500*                CS-30.
0016 001600*                CS-40.
0017 001700*                CS-60.
0018 001800 INPUT-OUTPUT SECTION.
0019 001900
0020 002000 FILE-CONTROL.
0021 002100
0022 002200 COPY "PROXI$CF03.SL".
0023C 000100 SELECT PRINTER-FILE
0024C 000200 ASSIGN TO PRINTER, PRINTER-NAME
0025C 000300 FILE STATUS IS PRINTER-STATUS.
0026
0027 002300 COPY "PROXI$CF04.SL".
0028C 000400 SELECT INSTNAME-FILE
0029C 000500 ASSIGN DISK "FIRST$NAME"
0030C 000600 ORGANIZATION SEQUENTIAL
0031C 000700 ACCESS MODE IS SEQUENTIAL
0032C 000800 FILE STATUS IS INSTNAME-STATUS.
0033
0034 002400
0035 002500 COPY "LISTDATA.SL".
0036C 000000
0037C 000100 SELECT LISTDATA
0038C 000200 ASSIGN TO DISK "LISTDATA"
0039C 000300 ORGANIZATION IS INDEXED
0040C 000400 ACCESS MODE IS DYNAMIC
0041C 000500 RECORD KEY IS LISTDATA-KEY
0042C 000600 FILE STATUS IS LISTDATA-STATUS.
0043C 000700
0044
0045 002600
0046 002700
0047 002800 SELECT SORTFILE
0048 002900 ASSIGN TO DISK SORTFILE-NAME
0049 003000 ORGANIZATION IS INDEXED
0050 003100 ACCESS MODE IS DYNAMIC
0051 003200 RECORD KEY IS SORTFILE-KEY
0052 003300 FILE STATUS IS SORTFILE-STATUS.

```

Figure C-6. Compilation Listing of the Report Writer Program (continues)

```

0053 003400/
0054 003500 DATA DIVISION.
0055 003600 FILE SECTION.
0056 003700 COPY "PROXISCF03.FD".
0057C 000100 FD PRINTER-FILE
0058C 000200 LABEL RECORDS ARE OMITTED.
0059C 000300 01 PRINTER-RECORD.
0060C 000400 10 PRINTER-FIELD-1-1 PIC X(1).
0061C 000500 10 PRINTER-FIELD-2-12 PIC X(11).
0062C 000600 10 PRINTER-FIELD-13-24 PIC X(12).
0063C 000700 10 PRINTER-FIELD-25-36 PIC X(12).
0064C 000800 10 PRINTER-FIELD-37-48 PIC X(12).
0065C 000900 10 PRINTER-FIELD-49-60 PIC X(12).
0066C 001000 10 PRINTER-FIELD-61-72 PIC X(12).
0067C 001100 10 PRINTER-FIELD-73-84 PIC X(12).
0068C 001200 10 PRINTER-FIELD-85-96 PIC X(12).
0069C 001300 10 PRINTER-FIELD-97-108 PIC X(12).
0070C 001400 10 PRINTER-FIELD-109-120 PIC X(12).
0071C 001500 10 PRINTER-FIELD-121-132 PIC X(12).
0072C 001600
0073C 001700 01 PRINTER-RECORD-1 PIC X(1).
0074C 001800 01 PRINTER-RECORD-12 PIC X(12).
0075C 001900 01 PRINTER-RECORD-24 PIC X(24).
0076C 002000 01 PRINTER-RECORD-36 PIC X(36).
0077C 002100 01 PRINTER-RECORD-48 PIC X(48).
0078C 002200 01 PRINTER-RECORD-60 PIC X(60).
0079C 002300 01 PRINTER-RECORD-72 PIC X(72).
0080C 002400 01 PRINTER-RECORD-84 PIC X(84).
0081C 002500 01 PRINTER-RECORD-96 PIC X(96).
0082C 002600 01 PRINTER-RECORD-108 PIC X(108).
0083C 002700 01 PRINTER-RECORD-120 PIC X(120).
0084
0085 003800 COPY "PROXISCF04.FD".
0086C 002900 FD INSTNAME-FILE
0087C 003000 RECORDING MODE IS FIXED
0088C 003100 LABEL RECORDS ARE STANDARD.
0089C 003200 01 INSTNAME-RECORD.
0090C 003300 10 INSTNAME-NAME PIC X(50).
0091
0092 003900
0093 004000 COPY "LISTDATA.FD".
0094C 000100 FD LISTDATA
0095C 000200 RECORDING MODE IS VARIABLE
0096C 000300 LABEL RECORDS ARE OMITTED.
0097C 000400 01 LISTING-RECORD.
0098C 000500 03 LISTDATA-KEY PIC 9(5).
0099C 000600 03 LIST-DATE PIC X(8).
0100C 000700 03 TOWN PIC X(12).
0101C 000800 03 DESCRIPTION PIC X(30).
0102C 000900 03 ORIGINAL-PRICE PIC 9(6).
0103C 001000 03 CURRENT-PRICE PIC 9(6).
0104
0105 004100
0106 004200
0107 004300 FD SORTFILE
0108 004400 LABEL RECORDS ARE STANDARD.
0109 004500 01 SORTFILE-RECORD.
0110 004600 05 SORTFILE-KEY.
0111 004700 10 SORTFILE-KEY-1 PIC X(12).
0112 004800 10 SORTFILE-KEY-2 PIC 9(6).
0113 004900 10 SORTFILE-KEY-3 PIC 9(5).
0114 005000 10 SORTFILE-KEY-KEY PIC 9(5).
0115 005100 05 SORTFILE-DATA PIC XX.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0116      005200/
0117      005300 WORKING-STORAGE SECTION.
0118      005400
0119      005500      COPY "LISTDATA.WS".
0120C     000100 01 LISTDATA-STATUS          PIC XX.
0121C     000200
0122
0123      005600
0124      005700
0125      005800 01 SORTFILE-STATUS          PIC XX VALUE SPACES.
0126      005900 01 SORTFILE-NAME.
0127      006000      10 FILLER              PIC X(8) VALUE "SORTFILE".
0128      006100      10 SORTFILE-LINE-NO PIC 99.
0129      006200
0130      006300 01 PASS-FLAG      PIC 9 VALUE 1.
0131      006400
0132      006500      COPY "PROXISCF01.WS".
0133C     000100
0134C     000200 01 ANY-CHANGE-ANSWER.
0135C     000300      05 ANY-CHANGE-CHAR-1   PIC X.
0136C     000400      05 FILLER              PIC X.
0137C     000500 01 ANY-CHANGE-DEFAULT     PIC X.
0138C     000600 01 CHANGE-NO              PIC 99.
0139C     000700 01 WHAT-NUMBER-ANSWER.
0140C     000800      03 WHAT-NUMBER-CHAR   PIC X OCCURS 2 TIMES.
0141C     000900
0142C     001000 01 ESCAPE-CODE      PIC 99.
0143C     001100
0144C     001200 01 ESCAPE-KEY      PIC 99 VALUE 01.
0145C     001300 01 F1-KEY          PIC 99 VALUE 02.
0146C     001400 01 F2-KEY          PIC 99 VALUE 03.
0147C     001500 01 F3-KEY          PIC 99 VALUE 04.
0148C     001600 01 F7-KEY          PIC 99 VALUE 08.
0149C     001700 01 F8-KEY          PIC 99 VALUE 09.
0150C     001800 01 END-KEY         PIC 99 VALUE 09.
0151C     001900
0152C     002000
0153C     002100 01 FILE-ERROR-STATUS     PIC XX.
0154C     002200 01 FILE-ERROR-NAME     PIC X(10) VALUE SPACES.
0155C     002300
0156C     002400 01 FILE-STATUS-CODE-TABLE.
0157C     002500      03 RECORD-ON-FILE     PIC XX VALUE "00".
0158C     002600      03 I-O-OK            PIC XX VALUE "00".
0159C     002700      03 AT-END            PIC XX VALUE "10".
0160C     002800      03 INVALID-KEY       PIC XX VALUE "21".
0161C     002900      03 DUPLICATE-KEY     PIC XX VALUE "22".
0162C     003000      03 RECORD-NOT-FOUND PIC XX VALUE "23".
0163C     003100      03 BOUNDARY-ERROR   PIC XX VALUE "24".
0164C     003200      03 HARDWARE-ERROR   PIC XX VALUE "30".
0165C     003300      03 DISK-FULL        PIC XX VALUE "34".
0166C     003400      03 OPEN-ERROR       PIC XX VALUE "91".
0167C     003500      03 CLOSE-ERROR     PIC XX VALUE "91".
0168C     003600      03 NO-FILE         PIC XX VALUE "91".
0169C     003700      03 MODE-ERROR     PIC XX VALUE "92".
0170C     003800      03 RECORD-LOCKED  PIC XX VALUE "94".
0171C     003900      03 USE-ERROR      PIC XX VALUE "94".
0172C     004000      03 NO-DIRECTORY   PIC XX VALUE "96".
0173C     004100      03 OVER-LOCK-LIMIT PIC XX VALUE "97".
0174C     004200      03 NO-CONTIGUOUS-SPACE PIC XX VALUE "98".
0175C     004300      03 FULL-PRINTER-TABLE PIC XX VALUE "99".
0176C     004400
0177C     004410 01 HASH-W          PIC XX      VALUE "#W".
0178C     004420

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0179C 004500 01 MESSAGE-FIELD PIC X(45).
0180C 004600 01 MESSAGE-WAIT-ENTRY PIC X.
0181C 004700
0182C 004800 01 QUESTION-ANSWER PIC X.
0183C 004900
0184C 005000 01 U LINES PIC X(79) VALUE ALL " ".
0185C 005100 01 PROMPT-LINE PIC X(79) VALUE SPACES.
0186C 005200 01 MENU-SELECTION PIC 9.
0187C 005300 01 SELECTION-TYPE PIC X(14).
0188C 005400
0189C 005410
0190C 005420 01 SYSTEM-DATE.
0191C 005430 03 SYSTEM-YEAR PIC 99 VALUE ZERO.
0192C 005440 03 SYSTEM-MONTH PIC 99 VALUE ZERO.
0193C 005450 03 SYSTEM-DAY PIC 99 VALUE ZERO.
0194C 005460
0195C 005470 01 TODAYS-DATE PIC 999999 VALUE ZERO.
0196C 005472
0197C 005474 01 TODAYS-DATE-REDEFINE REDEFINES TODAYS-DATE.
0198C 005480 03 TODAYS-MONTH PIC 99.
0199C 005490 03 TODAYS-DAY PIC 99.
0200C 005500 03 TODAYS-YEAR PIC 99.
0201C 005600
0202C 005700 01 MONTH-NAMES.
0203C 005800 10 FILLER PIC X(36)
0204C 005900 VALUE "JANFEBMARAPR MAYJUNJUL AUGSEP OCTNOVDEC".
0205C 006000
0206C 006100 01 MONTH-TABLE REDEFINES MONTH-NAMES.
0207C 006200 10 MONTH-NAME PIC XXX OCCURS 12 TIMES.
0208C 006300
0209C 006400 01 SYSTEM-TIME.
0210C 006500 10 SYSTEM-HOUR PIC 99 VALUE ZERO.
0211C 006600 10 SYSTEM-MINUTE PIC 99 VALUE ZERO.
0212C 006700 10 SYSTEM-SECOND PIC 99 VALUE ZERO.
0213C 006800 10 SYSTEM-HUNDRETH PIC 99 VALUE ZERO.
0214
0215 006600 COPY "PROXISCF04.WS".
0216C 000100
0217C 000200 01 INSTNAME-STATUS PIC XX VALUE SPACE.
0218
0219 006700
0220 006800 01 APPLICATION-NAME PIC X(30)
0221 006900 VALUE "Current Listings".
0222 007000
0223 007100
0224 007200 01 SPOOL-CODE PIC X VALUE "0".
0225 007300
0226 007400 01 SPOOLER-FILE-NAME.
0227 007500 10 FILLER PIC X(08) VALUE "SPOOLIST".
0228 007600 10 SPOOL-LINE-NUMBER PIC 99.
0229 007700
0230 007800 01 PAGE-WIDTH PIC 999 VALUE 080.
0231 007900
0232 008000 01 PAGE-DEPTH PIC 99 VALUE 60.
0233 008100
0234 008200
0235 008300 01 REPORT-TITLE.
0236 008400 10 FILLER PIC X(21) VALUE SPACES.
0237 008500 10 FILLER PIC X(30) VALUE
0238 008600 "BIRKS & ABERGELDIE REAL ESTATE".
0239 008700 10 FILLER PIC X(8) VALUE
0240 008800 " COMPANY".
0241 008900

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0242 009000
0243 009100 01 LEGENDS.
0244 009200 10 LEGEND-1.
0245 009300 20 FILLER PIC X(33) VALUE
0246 009400 "This report shows the active list".
0247 009500 20 FILLER PIC X(33) VALUE
0248 009600 "ings for this office.".
0249 009700 20 FILLER PIC X(33) VALUE
0250 009800 SPACE.
0251 009900 20 FILLER PIC X(33) VALUE
0252 010000 SPACE.
0253 010100 10 LEGEND-2.
0254 010200 20 FILLER PIC X(33) VALUE
0255 010300 "For information about old listing".
0256 010400 20 FILLER PIC X(33) VALUE
0257 010500 "s, see the Blue Book.".
0258 010600 20 FILLER PIC X(33) VALUE
0259 010700 SPACE.
0260 010800 20 FILLER PIC X(33) VALUE
0261 010900 SPACE.
0262 011000
0263 011100 01 LEGEND-TABLE REDEFINES LEGENDS.
0264 011200 10 LEGEND-LINE OCCURS 2 TIMES
0265 011300 INDEXED BY LEGEND-INDEX
0266 011400 PIC X(132).
0267 011500 01 LEGENDS-TO-PRINT PIC 9 VALUE 2.
0268 011600 01 LEGENDS-FROM-PAGE-2-ONWARD PIC 9 VALUE 0.
0269 011700
0270 011800
0271 011900
0272 012000 01 HEADERS.
0273 012100 10 HEADER-1.
0274 012200 20 FILLER PIC X(33) VALUE
0275 012300 " Town Listing".
0276 012400 20 FILLER PIC X(33) VALUE
0277 012500 " Description Pric".
0278 012600 20 FILLER PIC X(33) VALUE
0279 012700 "e".
0280 012800 20 FILLER PIC X(33) VALUE
0281 012900 SPACE.
0282 013000 10 HEADER-2.
0283 013100 20 FILLER PIC X(33) VALUE
0284 013200 " -----".
0285 013300 20 FILLER PIC X(33) VALUE
0286 013400 " -----".
0287 013500 20 FILLER PIC X(33) VALUE
0288 013600 "-".
0289 013700 20 FILLER PIC X(33) VALUE
0290 013800 SPACE.
0291 013900
0292 014000 01 HEADER-TABLE REDEFINES HEADERS.
0293 014100 10 HEADER-LINE OCCURS 2 TIMES
0294 014200 INDEXED BY HEADER-INDEX
0295 014300 PIC X(132).
0296 014400 01 HEADERS-TO-PRINT PIC 9 VALUE 2.
0297 014500
0298 014600
0299 014700
0300 014800 COPY "PROXISCF03.WS".
0301C 000100
0302C 000300 01 PRINTER-STATUS PIC XX VALUE SPACE.
0303C 000400
0304C 000500 01 PRINTER-NAME.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0305C 000600 10 PRINTER-DEVICE PIC XXXX.
0306C 000700 10 FILLER PIC X(6) VALUE LOW-VALUES.
0307C 000800
0308C 000900 01 TOP-LINE.
0309C 001000 10 FILLER PIC X(5) VALUE "RUN: ".
0310C 001100 10 TOP-DAY PIC Z9.
0311C 001200 10 FILLER PIC X VALUE "-".
0312C 001300 10 TOP-MONTH PIC XXX.
0313C 001400 10 FILLER PIC X VALUE "-".
0314C 001500 10 TOP-YEAR PIC 99.
0315C 001600 10 TOP-TIME PIC X(6) VALUE SPACE.
0316C 001700 10 TOP-TIME-REDEF REDEFINES TOP-TIME.
0317C 001800 20 FILLER PIC X.
0318C 001900 20 TOP-HOUR PIC Z9.
0319C 002000 20 TOP-COLON PIC X.
0320C 002100 20 TOP-MINUTE PIC 99.
0321C 002200 10 FILLER PIC X(21) VALUE SPACES.
0322C 002300 10 TOP-NAME PIC X(50) VALUE
0323C 002400 " INTEGRATED SOFTWARE SYSTEMS PTY. LTD. ".
0324C 002500
0325C 002600 10 FILLER PIC X(33) VALUE SPACE.
0326C 002700 10 FILLER PIC X(5) VALUE "PAGE ".
0327C 002800 10 TOP-NUMBER PIC ZZ9.
0328C 002900 01 TOP-LINE-80 REDEFINES TOP-LINE.
0329C 003000 10 FILLER PIC X(21).
0330C 003100 10 TOP-NAME-80 PIC X(51).
0331C 003200 10 TOP-PAGE-80 PIC X(5).
0332C 003300 10 TOP-NUMBER-80 PIC ZZ9.
0333C 003400 10 TOP-81-132 PIC X(52).
0334C 003410
0335C 003420 01 PRINT-LINE-SAVE REDEFINES TOP-LINE.
0336C 003430 10 FILLER PIC X(132).
0337C 003440
0338C 003440 01 DETAIL-PRINT PIC 9 VALUE ZERO.
0339C 003500
0340C 003600 01 PAGE-NUMBER PIC 999 VALUE ZERO.
0341C 003700
0342C 003800 01 LINE-COUNT PIC 999 VALUE 999.
0343C 003900
0344C 004000 01 NUMBER-OF-LEGENDS-PAGE-1 PIC 9 VALUE ZERO.
0345C 004100
0346C 004200 01 PRINT-LINE PIC X(132) VALUE SPACES.
0347
0348 014900
0349 015000 01 DETAIL-LINE-1
0350 015100 REDEFINES PRINT-LINE.
0351 015200 05 FILLER PIC X(69).
0352 015300 05 DETAIL-1-FIELD-1 PIC X(7).
0353 015400 01 DETAIL-LINE-2
0354 015500 REDEFINES PRINT-LINE.
0355 015600 05 FILLER PIC X(4).
0356 015700 05 DETAIL-2-FIELD-1 PIC X(12).
0357 015800 05 FILLER PIC X(3).
0358 015900 05 DETAIL-2-FIELD-2 PIC ZZZZ9.
0359 016000 05 FILLER PIC X(3).
0360 016100 05 DETAIL-2-FIELD-3 PIC X(30).
0361 016200 05 FILLER PIC X(3).
0362 016300 05 DETAIL-2-FIELD-4 PIC $Z,ZZ9,999.
0363 016400 05 PRINT-TOTAL-1 REDEFINES DETAIL-2-FIELD-4
0364 016500 PIC $Z,ZZ9,999.
0365 016600 01 TOTAL-LINE-1
0366 016700 REDEFINES PRINT-LINE.
0367 016800 05 FILLER PIC X(31).

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)


```
0368 016900      05 TOTAL-1-FIELD-1  PIC X(26).
0369 017000
0370 017100 01 TOTAL-KEYS.
0371 017200      10 TOTAL-KEY-1          VALUE LOW-VALUES.
0372 017300          20 FILLER          PIC X(12).
0373 017400
0374 017500 01 TOTAL-SUB  PIC 99.
0375 017600 01 TOTAL-MAX  PIC 99  VALUE 1.
0376 017700 01 TOTALS    VALUE ZERO.
0377 017800      10 TOTALS-LEVEL OCCURS 1 TIMES.
0378 017900          20 TOTAL-1  PIC 99999999.
0379 018000
```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0380 018100/
0381 018200 SCREEN SECTION.
0382 018300
0383 018400 COPY "PROXI$CF01.SD".
0384C 000100
0385C
0386C 000200 01 ANY-CHANGE-SCREEN.
0387C
0388C 000300 05 LINE 24 BLANK LINE.
0389C
0390C 000400 05 LINE 24 COL 1 "ANY CHANGE ?".
0391C
0392C 000500 05 LINE 24 COL 15 PIC XX FROM SPACE TO ANY-CHANGE-ANSWER.
0393C
0394C 000600
0395C
0396C 000700 01 ANY-CHANGE-DEFAULT-SCREEN.
0397C
0398C 000800 05 LINE 24 COL 15 PIC X FROM ANY-CHANGE-ANSWER.
0399C
0400C 000900
0401C
0402C 001000 01 FILE-ACCESS-ERROR-SCREEN.
0403C
0404C 001100 05 BLANK SCREEN.
0405C
0406C 001200 05 BLINK LINE 12 COL 20 "E R R O R".
0407C
0408C 001300 05 " . . . FILE ACCESS ERROR: ".
0409C
0410C 001400 05 PIC X(2) FROM FILE-ERROR-STATUS.
0411C
0412C 001500 05 LINE 13 COL 20 "FROM".
0413C
0414C 001600 05 LINE 13 COL 26 PIC X(10) FROM FILE-ERROR-NAME.
0415C
0416C 001700 05 LINE 13 COL 37 "FILE".
0417C
0418C 001800
0419C
0420C 001900 01 MESSAGE-SCREEN.
0421C
0422C 001900 05 ERROR-MESSAGE-LINE.
0423C 002000 10 CLEAR-MESSAGE LINE 24 BLANK LINE.
0424C
0425C 002100 10 LINE 24 COL 5 PIC X(45) FROM MESSAGE-FIELD.
0426C 002110 10 BELL.
0427C
0428C 002200 05 LINE 24 COL 52 "TYPE "CR" TO CONTINUE".
0429C
0430C 002300 05 LINE 24 COL 78 PIC X(1) TO MESSAGE-WAIT-ENTRY.
0431C
0432C
0433C 002500
0434C
0435C 002600 01 PROMPT-SCREEN.
0436C
0437C 002700 10 CLEAR-PROMPT LINE 24 BLANK LINE.
0438C
0439C 002800 10 LINE 24 COL 1 PIC X(79) FROM PROMPT-LINE.
0440C
0441C 002900
0442C

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0443C 003000 01 QUESTION-SCREEN.
0444C
0445C 003100 03 LINE 24 COLUMN 01 PIC X(45) FROM MESSAGE-FIELD.
0446C
0447C 003200 03 LINE 24 COLUMN 31 PIC X
0448C
0449C 003300 TO QUESTION-ANSWER.
0450C
0451C 003400
0452C
0453C 003500 01 WAIT-PRINTING.
0454C
0455C 003600 10 BLANK SCREEN.
0456C
0457C 003700 10 LINE 1 COL 1 PIC X(30) FROM APPLICATION-NAME.
0458C
0459C 003800 10 LINE 12 COL 20 "PRINTING ..... PLEASE WAIT".
0460C
0461C 003900
0462C
0463C 004000 01 WAIT-PROCESSING.
0464C
0465C 004100 10 BLANK SCREEN.
0466C
0467C 004200 10 LINE 1 COL 1 PIC X(30) FROM APPLICATION-NAME.
0468C
0469C 004300 10 LINE 12 COL 20 "PROCESSING ..... PLEASE WAIT".
0470C
0471C 004400
0472C
0473C 004500 01 WAIT-LINE-24.
0474C
0475C 004600 10 LINE 24 BLANK LINE.
0476C
0477C 004700 10 LINE 24 COL 1 "PLEASE WAIT".
0478C
0479C 004800
0480C
0481C 004900 01 WHAT-NUMBER-SCREEN.
0482C
0483C 005000 05 LINE 24 COL 18 "WHAT NUMBER ?".
0484C
0485C 005100 05 LINE 24 COL 33 PIC XX FROM SPACES TO WHAT-NUMBER-ANSWER.
0486C
0487
0488 018500 COPY "PROXISCF03.SD".
0489C 000100
0490C 000200 01 SPOOL-OPTION-SCREEN.
0491C 000300 10 BLANK SCREEN.
0492C 000400 10 LINE 1 COL 1 PIC X(30) FROM APPLICATION-NAME.
0493C 000500 10 LINE 12 COL 20 "OUTPUT TO THE PRINTER ?".
0494C 000600 10 LINE 12 COL 44 PIC X TO QUESTION-ANSWER.
0495C 000700 01 SPOOL-OPTION-END.
0496C 000800 10 LINE 12 COL 44 "END".
0497
0498 018600

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0499 018700/
0500 018800 PROCEDURE DIVISION.
0501 018900 DECLARATIVES.
0502 019000
0503 019100     COPY "PROXI$CF03.DS".
0504C 000100
0505C 000200 PRINTER-ERROR SECTION.
0506C 000300     USE AFTER ERROR PROCEDURE ON PRINTER-FILE.
0507C 000400 PROCESS-PRINTER-ERROR.
0508C 000500     IF PRINTER-STATUS = HARDWARE-ERROR
0509C 000600         MOVE PRINTER-STATUS TO FILE-ERROR-STATUS
0510C 000700         MOVE SPOOLER-FILE-NAME TO FILE-ERROR-NAME
0511C 000800         DISPLAY FILE-ACCESS-ERROR-SCREEN
0512C 000900         STOP RUN.
0513
0514 019200     COPY "PROXI$CF04.DS".
0515C 001000
0516C 001100 INSTNAME-ERROR SECTION.
0517C 001200     USE AFTER ERROR PROCEDURE ON INSTNAME-FILE.
0518C 001300 PROCESS-INSTNAME-ERROR.
0519C 001400     IF INSTNAME-STATUS = HARDWARE-ERROR
0520C 001500         OR INSTNAME-STATUS = DISK-FULL
0521C 001600         OR INSTNAME-STATUS = OVER-LOCK-LIMIT
0522C 001700         MOVE INSTNAME-STATUS TO FILE-ERROR-STATUS
0523C 001800         MOVE "FIRST$NAME" TO FILE-ERROR-NAME
0524C 001900         DISPLAY FILE-ACCESS-ERROR-SCREEN
0525C 002000         STOP RUN.
0526
0527 019300
0528 019400     COPY "LISTDATA.DS".
0529C 000100 LISTDATA-ERROR SECTION.
0530C 000200     USE AFTER ERROR PROCEDURE ON LISTDATA.
0531C 000300 PROCESS-LISTDATA-ERROR.
0532C 000400     IF LISTDATA-STATUS = HARDWARE-ERROR OR
0533C 000500         LISTDATA-STATUS = OPEN-ERROR OR
0534C 000600         LISTDATA-STATUS = DISK-FULL OR
0535C 000700         LISTDATA-STATUS = OVER-LOCK-LIMIT
0536C 000800         MOVE LISTDATA-STATUS TO FILE-ERROR-STATUS
0537C 000900         MOVE "LISTDATA" TO FILE-ERROR-NAME
0538C 001000         DISPLAY FILE-ACCESS-ERROR-SCREEN
0539C 001100         STOP RUN.
0540
0541 019500
0542 019600
0543 019700 SORTFILE-ERROR SECTION.
0544 019800     USE AFTER ERROR PROCEDURE ON SORTFILE.
0545 019900 PROCESS-SORTFILE-ERROR.
0546 020000     IF SORTFILE-STATUS = HARDWARE-ERROR OR
0547 020100         SORTFILE-STATUS = DISK-FULL OR
0548 020200         SORTFILE-STATUS = OVER-LOCK-LIMIT
0549 020300         MOVE SORTFILE-STATUS TO FILE-ERROR-STATUS
0550 020400         MOVE "SORTFILE" TO FILE-ERROR-NAME
0551 020500         DISPLAY FILE-ACCESS-ERROR-SCREEN
0552 020600         STOP RUN.
0553 020700
0554 020800 END DECLARATIVES.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0555 020900/
0556 021000* ***** LEVEL 1 *****
0557 021100
0558 021200 MAIN-LOGIC SECTION.
0559 021300 BEGIN.
0560 021400 PERFORM OPEN-FILES.
0561 021500 PRINT-RECORDS.
0562 021600 DISPLAY WAIT-PROCESSING.
0563 021700 PERFORM OPEN-SORTFILE.
0564 021800 IF SORTFILE-STATUS IS NOT = I-O-OK
0565 021900 PERFORM CALL-ERROR-PROGRAM.
0566 022000 PERFORM PRINT-THE-LINES.
0567 022100 PERFORM CLOSE-SORTFILE.
0568 022200 PERFORM LINE-FEED 2 TIMES.
0569 022300 END-OF-REPORT.
0570 022400 PERFORM TOP-OF-FORM.
0571 022500 DISPLAY WAIT-PROCESSING.
0572 022600 PERFORM CLOSE-FILES.
0573 022700 PERFORM CALL-NEXT-PROGRAM.
0574 022800 STOP RUN.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0575 022900/
0576 023000* ***** LEVEL 2 *****
0577 023100
0578 023200 OPEN-FILES.
0579 023300 PERFORM PRINT-SYSTEM-DATE.
0580 023400 PERFORM OPEN-LISTDATA.
0581 023500 IF LISTDATA-STATUS IS NOT = I-O-OK
0582 023600 PERFORM CALL-ERROR-PROGRAM.
0583 023700
0584 023800
0585 023900 PERFORM PRINTER-OPEN.
0586 024000 IF PRINTER-STATUS IS NOT = I-O-OK
0587 024100 IF ESCAPE-CODE = END-KEY
0588 024200 PERFORM CALL-NEXT-PROGRAM
0589 024300 ELSE
0590 024400 PERFORM CALL-ERROR-PROGRAM.
0591 024500
0592 024600 PRINT-THE-LINES.
0593 024700 IF PASS-FLAG = 1
0594 024800 PERFORM READ-NEXT-LISTDATA-RECORD
0595 024900 IF LISTDATA-STATUS = AT-END
0596 025000 CLOSE SORTFILE
0597 025100 OPEN INPUT SORTFILE
0598 025200 MOVE LOW-VALUES TO SORTFILE-KEY
0599 025300 START SORTFILE KEY NOT < SORTFILE-KEY
0600 025400 MOVE 2 TO PASS-FLAG
0601 025500 DISPLAY WAIT-PRINTING
0602 025600 ELSE
0603 025700 PERFORM WRITE-SORTFILE
0604 025800 GO TO PRINT-THE-LINES.
0605 025900 PERFORM READ-NEXT-SORTFILE-RECORD.
0606 026000 PERFORM CHECK-FOR-CONTROL-BREAK.
0607 026100 IF LISTDATA-STATUS NOT = AT-END
0608 026200 PERFORM PRINT-DETAIL-LINE-1
0609 026300 PERFORM PRINT-DETAIL-LINE-2
0610 026400 GO TO PRINT-THE-LINES.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0611 026500/
0612 026600* ***** LEVEL 3 *****
0613 026700
0614 026800
0615 026900 CALL-NEXT-PROGRAM.
0616 027000     DISPLAY WAIT-LINE-24.
0617 027100     CALL PROGRAM "LOGON".
0618 027200     STOP RUN.
0619 027300
0620 027400 CALL-ERROR-PROGRAM.
0621 027500     DISPLAY WAIT-LINE-24.
0622 027600     CALL PROGRAM "LOGON".
0623 027700     STOP RUN.
0624 027800
0625 027900 WRITE-SORTFILE.
0626 028000     MOVE TOWN TO SORTFILE-KEY-1
0627 028100     MOVE CURRENT-PRICE TO SORTFILE-KEY-2
0628 028200     MOVE LISTDATA-KEY TO SORTFILE-KEY-3
0629 028300     MOVE LISTDATA-KEY TO SORTFILE-KEY-KEY
0630 028400     PERFORM WRITE-SORTFILE-RECORD.
0631 028500
0632 028600
0633 028700 PRINT-DETAIL-LINE-1.
0634 028800     IF CURRENT-PRICE <
0635 028900         ORIGINAL-PRICE
0636 029000         PERFORM DETAIL-LINE-1-LOGIC.
0637 029100 PRINT-DETAIL-LINE-2.
0638 029200     PERFORM DETAIL-LINE-2-LOGIC.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0639 029300/
0640 029400* ***** LEVEL 4 *****
0641 029500
0642 029600
0643 029700 CHECK-FOR-CONTROL-BREAK.
0644 029800 IF LISTDATA-STATUS = AT-END
0645 029900     PERFORM CONTROL-BREAK-1 THRU CONTROL-BREAK-1.
0646 030000 IF TOWN NOT = TOTAL-KEY-1
0647 030100     IF TOTAL-KEY-1 = LOW-VALUES
0648 030200         MOVE TOWN
0649 030300             TO TOTAL-KEY-1
0650 030400     ELSE
0651 030500         PERFORM CONTROL-BREAK-1 THRU CONTROL-BREAK-1.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)


```
0652 030600/
0653 030700* ***** LEVEL 5 *****
0654 030800
0655 030900 CONTROL-BREAK-1.
0656 031000 MOVE 1 TO TOTAL-SUB.
0657 031100 PERFORM LINE-FEED 1 TIMES.
0658 031200 PERFORM MOVE-TOTALS-1.
0659 031300 PERFORM TOTAL-LINE-1-LOGIC.
0660 031400 PERFORM LINE-FEED 2 TIMES.
0661 031500 MOVE TOWN TO TOTAL-KEY-1.
0662 031600 MOVE ZEROS TO TOTALS-LEVEL (1).
```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0663 031700/
0664 031800* ***** LEVEL 6 *****
0665 031900
0666 032000 MOVE-TOTALS-1.
0667 032100 MOVE TOTAL-1 (TOTAL-SUB) TO PRINT-TOTAL-1.
0668 032200 IF TOTAL-SUB < TOTAL-MAX
0669 032300 ADD TOTAL-1 (TOTAL-SUB) TO TOTAL-1 (TOTAL-SUB + 1).
0670 032400
0671 032500 DETAIL-LINE-1-LOGIC.
0672 032600 MOVE "REDUCED"
0673 032700 TO DETAIL-1-FIELD-1.
0674 032800 DETAIL-LINE-2-LOGIC.
0675 032900 MOVE TOWN
0676 033000 TO DETAIL-2-FIELD-1.
0677 033100 MOVE LISTDATA-KEY
0678 033200 TO DETAIL-2-FIELD-2.
0679 033300 MOVE DESCRIPTION
0680 033400 TO DETAIL-2-FIELD-3.
0681 033500 MOVE CURRENT-PRICE
0682 033600 TO DETAIL-2-FIELD-4.
0683 033700 ADD CURRENT-PRICE TO TOTAL-1 (1).
0684 033800 PERFORM PRINT.
0685 033900 TOTAL-LINE-1-LOGIC.
0686 034000 MOVE "Total value for this town:"
0687 034100 TO TOTAL-1-FIELD-1.
0688 034200 PERFORM PRINT.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0689 034300/
0690 034400* ***** INPUT/OUTPUT ROUTINES *****
0691 034500
0692 034600 I-O-SECTION SECTION.
0693 034700 OPEN-LISTDATA.
0694 034800 OPEN INPUT LISTDATA.
0695 034900 IF LISTDATA-STATUS = USE-ERROR
0696 035000 PERFORM LISTDATA-IN-USE.
0697 035100 READ-NEXT-LISTDATA-RECORD.
0698 035200 READ LISTDATA NEXT RECORD.
0699 035300 IF LISTDATA-STATUS = RECORD-LOCKED
0700 035400 PERFORM LISTDATA-RECORD-IN-USE
0701 035500 GO TO READ-NEXT-LISTDATA-RECORD.
0702 035600 READ-LISTDATA-RECORD.
0703 035700 READ LISTDATA RECORD.
0704 035800 IF LISTDATA-STATUS = RECORD-LOCKED
0705 035900 PERFORM LISTDATA-RECORD-IN-USE
0706 036000 GO TO READ-LISTDATA-RECORD.
0707 036100 LISTDATA-IN-USE.
0708 036200 MOVE "LISTDATA FILE IN USE - TRY AGAIN LATER" TO
0709 036300 MESSAGE-FIELD.
0710 036400 PERFORM DISPLAY-MESSAGE.
0711 036500 LISTDATA-RECORD-IN-USE.
0712 036600 MOVE "LISTDATA RECORD IN USE" TO
0713 036700 MESSAGE-FIELD.
0714 036800 DISPLAY ERROR-MESSAGE-LINE.
0715 036900 CALL PROGRAM HASH-W.
0716 037000 CALL PROGRAM HASH-W.
0717 037100 DISPLAY CLEAR-MESSAGE.
0718 037200
0719 037300
0720 037400 CLOSE-FILES.
0721 037500 PERFORM CLOSE-PRINTER-FILE.
0722 037600 CLOSE LISTDATA.
0723 037700
0724 037800
0725 037900 OPEN-SORTFILE.
0726 038000 ACCEPT SORTFILE-LINE-NO FROM LINE NUMBER.
0727 038100 DELETE FILE SORTFILE.
0728 038200 OPEN OUTPUT SORTFILE.
0729 038300 IF SORTFILE-STATUS NOT = I-O-OK
0730 038400 PERFORM SORTFILE-IN-USE.
0731 038500 WRITE-SORTFILE-RECORD.
0732 038600 WRITE SORTFILE-RECORD.
0733 038700 READ-NEXT-SORTFILE-RECORD.
0734 038800 READ SORTFILE NEXT.
0735 038900 MOVE SORTFILE-STATUS TO LISTDATA-STATUS
0736 039000 IF SORTFILE-STATUS NOT = AT-END
0737 039100 MOVE SORTFILE-KEY-KEY TO LISTDATA-KEY
0738 039200 PERFORM READ-LISTDATA-RECORD
0739 039300 IF LISTDATA-STATUS = RECORD-NOT-FOUND
0740 039400 GO TO READ-NEXT-SORTFILE-RECORD.
0741 039500 CLOSE-SORTFILE.
0742 039600 CLOSE SORTFILE.
0743 039700 DELETE FILE SORTFILE.
0744 039800 SORTFILE-IN-USE.
0745 039900 MOVE "SORTFILE FILE IN USE - TRY AGAIN LATER" TO
0746 040000 MESSAGE-FIELD.
0747 040100 PERFORM DISPLAY-MESSAGE.
0748 040200
0749 040300 COPY "PROXI$CF03.PL".
0750C 000100
0751C 000200 PRINTER-OPEN SECTION.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0752C 000300 GET-TOP-NAME.
0753C 000400 OPEN INPUT INSTNAME-FILE.
0754C 000500 IF INSTNAME-STATUS = I-O-OK
0755C 000600     READ INSTNAME-FILE
0756C 000700     IF INSTNAME-STATUS = I-O-OK
0757C 000800         MOVE INSTNAME-RECORD TO TOP-NAME.
0758C 000900     CLOSE INSTNAME-FILE.
0759C 001000 MOVE LEGENDS-TO-PRINT TO NUMBER-OF-LEGENDS-PAGE-1.
0760C 001100 IF PAGE-WIDTH = 80
0761C 001200     MOVE TOP-NAME TO TOP-NAME-80
0762C 001300     MOVE "PAGE" TO TOP-PAGE-80
0763C 001400     MOVE SPACE TO TOP-81-132.
0764C 001500 IF SPOOL-CODE = "S"
0765C 001600     GO TO OPEN-SPOOL-PRINTER-FILE
0766C 001700 ELSE
0767C 001800     IF SPOOL-CODE = "P"
0768C 001900         GO TO OPEN-LPT-PRINTER-FILE.
0769C 002000 SPOOL-OPTION.
0770C 002100 DISPLAY SPOOL-OPTION-SCREEN.
0771C 002200 ACCEPT SPOOL-OPTION-SCREEN.
0772C 002300 ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0773C 002400 IF ESCAPE-CODE = END-KEY
0774C 002500     DISPLAY SPOOL-OPTION-END
0775C 002600     MOVE SPACE TO PRINTER-NAME
0776C 002700     GO TO PRINTER-OPEN-EXIT
0777C 002800 ELSE
0778C 002900     IF QUESTION-ANSWER = "N"
0779C 003000         ACCEPT SPOOL-LINE-NUMBER FROM LINE
0780C 003100         GO TO OPEN-SPOOL-PRINTER-FILE.
0781C 003200 OPEN-LPT-PRINTER-FILE.
0782C 003300 MOVE LOW-VALUES TO PRINTER-NAME.
0783C 003400 MOVE "@LPT" TO PRINTER-DEVICE.
0784C 003500 OPEN OUTPUT PRINTER-FILE.
0785C 003600 IF PRINTER-STATUS = USE-ERROR
0786C 003700     MOVE "PRINTER IN USE" TO MESSAGE-FIELD
0787C 003800     PERFORM DISPLAY-MESSAGE
0788C 003900     GO TO SPOOL-OPTION
0789C 004000 ELSE
0790C 004100     GO TO PRINTER-OPEN-EXIT.
0791C 004200 OPEN-SPOOL-PRINTER-FILE.
0792C 004300 MOVE SPOOLER-FILE-NAME TO PRINTER-NAME.
0793C 004400 OPEN OUTPUT PRINTER-FILE.
0794C 004500 IF PRINTER-STATUS = FULL-PRINTER-TABLE
0795C 004600     PERFORM SPOOLER-ACCESS-TABLE-FULL
0796C 004700     IF SPOOL-CODE = "S"
0797C 004800         GO TO OPEN-SPOOL-PRINTER-FILE
0798C 004900     ELSE
0799C 005000         GO TO SPOOL-OPTION
0800C 005100 ELSE
0801C 005200     GO TO PRINTER-OPEN-EXIT.
0802C 005300 SPOOLER-ACCESS-TABLE-FULL.
0803C 005400 MOVE "P.A.S.S. QUEUE FULL" TO MESSAGE-FIELD.
0804C 005500 PERFORM DISPLAY-MESSAGE.
0805C 005600 PRINTER-OPEN-EXIT. EXIT.
0806C 005700
0807C 005800 PRINTER-CLOSE SECTION.
0808C 005900 CLOSE-PRINTER-FILE.
0809C 006000 CLOSE PRINTER-FILE.
0810C 006100
0811C 006200 PRINT SECTION.
0812C 006300 PRINT-DETAIL.
0813C 006400 IF LINE-COUNT > PAGE-DEPTH
0814C 006500     PERFORM NEW-PAGE.

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0815C 006600 MOVE PRINT-LINE TO PRINTER-RECORD.
0816C 006700 PERFORM PRINT-RECORD.
0817C 006800 MOVE SPACE TO PRINT-LINE.
0818C 006900 GO TO PRINT-EXIT.
0819C 007000 NEW-PAGE.
0820C 007100 PERFORM TOP-OF-FORM.
0821C 007200 ADD 1 TO PAGE-NUMBER.
0822C 007300 IF PAGE-NUMBER = 1
0823C 007400 MOVE NUMBER-OF-LEGENDS-PAGE-1 TO LEGENDS-TO-PRINT.
0824C 007500 IF PAGE-WIDTH = 80
0825C 007600 MOVE PAGE-NUMBER TO TOP-NUMBER-80
0826C 007700 ELSE
0827C 007800 MOVE PAGE-NUMBER TO TOP-NUMBER.
0828C 007850 MOVE ZERO TO LINE-COUNT.
0829C 007900 MOVE TOP-LINE TO PRINTER-RECORD.
0830C 008000 PERFORM PRINT-RECORD.
0831C 008050 PERFORM LINE-FEED.
0832C 008100 MOVE REPORT-TITLE TO PRINTER-RECORD.
0833C 008200 PERFORM PRINT-RECORD.
0834C 008300 PERFORM LINE-FEED.
0835C 008400 PERFORM PRINT-LEGEND VARYING LEGEND-INDEX FROM 1 BY 1
0836C 008500 UNTIL LEGEND-INDEX > LEGENDS-TO-PRINT.
0837C 008600 PERFORM LINE-FEED.
0838C 008700 MOVE LEGENDS-FROM-PAGE-2-ONWARD TO LEGENDS-TO-PRINT.
0839C 008800
0840C 008900 PERFORM PRINT-HEADERS VARYING HEADER-INDEX FROM 1 BY 1
0841C 009000 UNTIL HEADER-INDEX > HEADERS-TO-PRINT.
0842C 009100 PERFORM LINE-FEED 2 TIMES.
0843C 009200 TOP-OF-FORM.
0844C 009300 MOVE SPACE TO PRINTER-RECORD.
0845C 009400 WRITE PRINTER-RECORD-1 BEFORE ADVANCING PAGE.
0846C 009500 PRINT-LEGEND.
0847C 009600 MOVE LEGEND-LINE (LEGEND-INDEX) TO PRINTER-RECORD.
0848C 009700 PERFORM PRINT-RECORD.
0849C 009800 PRINT-HEADERS.
0850C 009900 MOVE HEADER-LINE (HEADER-INDEX) TO PRINTER-RECORD.
0851C 010000 PERFORM PRINT-RECORD.
0852C 010100 LINE-FEED.
0853C 010110 IF LINE-COUNT > PAGE-DEPTH
0854C 010120 PERFORM NEW-PAGE.
0855C 010200 MOVE SPACE TO PRINTER-RECORD.
0856C 010300 WRITE PRINTER-RECORD-1 BEFORE ADVANCING 1 LINE.
0857C 010400 ADD 1 TO LINE-COUNT.
0858C 010500 PRINT-EXIT. EXIT.
0859C 010600
0860C 010700 PRINT-RECORD SECTION.
0861C 010800 SCAN-PRINT-LINE.
0862C 010900 IF PRINTER-FIELD-121-132 NOT = SPACE
0863C 011000 WRITE PRINTER-RECORD BEFORE ADVANCING 1 LINE
0864C 011100 GO TO PRINT-RECORD-EXIT.
0865C 011200 IF PRINTER-FIELD-109-120 NOT = SPACE
0866C 011300 WRITE PRINTER-RECORD-120 BEFORE ADVANCING 1 LINE
0867C 011400 GO TO PRINT-RECORD-EXIT.
0868C 011500 IF PRINTER-FIELD-97-108 NOT = SPACE
0869C 011600 WRITE PRINTER-RECORD-108 BEFORE ADVANCING 1 LINE
0870C 011700 GO TO PRINT-RECORD-EXIT.
0871C 011800 IF PRINTER-FIELD-85-96 NOT = SPACE
0872C 011900 WRITE PRINTER-RECORD-96 BEFORE ADVANCING 1 LINE
0873C 012000 GO TO PRINT-RECORD-EXIT.
0874C 012100 IF PRINTER-FIELD-73-84 NOT = SPACE
0875C 012200 WRITE PRINTER-RECORD-84 BEFORE ADVANCING 1 LINE
0876C 012300 GO TO PRINT-RECORD-EXIT.
0877C 012400 IF PRINTER-FIELD-61-72 NOT = SPACE

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0878C 012500 WRITE PRINTER-RECORD-72 BEFORE ADVANCING 1 LINE
0879C 012600 GO TO PRINT-RECORD-EXIT.
0880C 012700 IF PRINTER-FIELD-49-60 NOT = SPACE
0881C 012800 WRITE PRINTER-RECORD-60 BEFORE ADVANCING 1 LINE
0882C 012900 GO TO PRINT-RECORD-EXIT.
0883C 013000 IF PRINTER-FIELD-37-48 NOT = SPACE
0884C 013100 WRITE PRINTER-RECORD-48 BEFORE ADVANCING 1 LINE
0885C 013200 GO TO PRINT-RECORD-EXIT.
0886C 013300 IF PRINTER-FIELD-25-36 NOT = SPACE
0887C 013400 WRITE PRINTER-RECORD-36 BEFORE ADVANCING 1 LINE
0888C 013500 GO TO PRINT-RECORD-EXIT.
0889C 013600 IF PRINTER-FIELD-13-24 NOT = SPACE
0890C 013700 WRITE PRINTER-RECORD-24 BEFORE ADVANCING 1 LINE
0891C 013800 GO TO PRINT-RECORD-EXIT.
0892C 013900 IF PRINTER-FIELD-2-12 NOT = SPACE
0893C 014000 WRITE PRINTER-RECORD-12 BEFORE ADVANCING 1 LINE
0894C 014100 GO TO PRINT-RECORD-EXIT.
0895C 014200 WRITE PRINTER-RECORD-1 BEFORE ADVANCING 1 LINE.
0896C 014300 PRINT-RECORD-EXIT.
0897C 014400 ADD 1 TO LINE-COUNT.
0898C 014450
0899C 014500 PRINT-SYSTEM-DATE-AND-TIME SECTION.
0900C 014600 PRINT-SYSTEM-DATE.
0901C 014700 ACCEPT SYSTEM-DATE FROM DATE.
0902C 014800 MOVE SYSTEM-DAY TO TOP-DAY.
0903C 014900 MOVE MONTH-NAME (SYSTEM-MONTH) TO TOP-MONTH.
0904C 015000 MOVE SYSTEM-YEAR TO TOP-YEAR.
0905C 015100 PRINT-SYSTEM-TIME.
0906C 015200 ACCEPT SYSTEM-TIME FROM TIME.
0907C 015300 MOVE ":" TO TOP-COLON.
0908C 015400 MOVE SYSTEM-MINUTE TO TOP-MINUTE.
0909C 015500 MOVE SYSTEM-HOUR TO TOP-HOUR.
0910
0911 040400

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0912 040500/
0913 040600* ***** UTILITY ROUTINES *****
0914 040700
0915 040800 UTILITIES SECTION.
0916 040900 COPY "PROXISCF01.PL".
0917C 010000
0918C
0919C 010010 ANY-CHANGE.
0920C
0921C 010020 DISPLAY ANY-CHANGE-SCREEN.
0922C
0923C 010030 ACCEPT ANY-CHANGE-SCREEN.
0924C
0925C 010040 ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0926C
0927C 010050 IF ESCAPE-CODE IS NOT = ESCAPE-KEY
0928C
0929C 010060 IF ANY-CHANGE-ANSWER = SPACE
0930C
0931C 010070 MOVE ANY-CHANGE-DEFAULT TO ANY-CHANGE-ANSWER
0932C
0933C 010080 DISPLAY ANY-CHANGE-DEFAULT-SCREEN.
0934C
0935C 010090 IF ESCAPE-CODE IS NOT = ESCAPE-KEY
0936C
0937C 010100 IF ANY-CHANGE-CHAR-1 NUMERIC
0938C
0939C 010110 MOVE ANY-CHANGE-ANSWER TO WHAT-NUMBER-ANSWER
0940C
0941C 010120 MOVE "Y" TO ANY-CHANGE-ANSWER
0942C
0943C 010130 ELSE
0944C
0945C 010140 MOVE SPACE TO WHAT-NUMBER-ANSWER
0946C
0947C 010150 IF (ANY-CHANGE-ANSWER = SPACE) OR
0948C
0949C 010160 (ANY-CHANGE-ANSWER IS NOT = "Y" AND
0950C
0951C 010170 ANY-CHANGE-ANSWER IS NOT = "N")
0952C
0953C 010180 GO TO ANY-CHANGE.
0954C
0955C 010190 DISPLAY-MESSAGE.
0956C
0957C 010200 DISPLAY MESSAGE-SCREEN.
0958C
0959C 010210 ACCEPT MESSAGE-SCREEN.
0960C
0961C 010220 DISPLAY CLEAR-MESSAGE.
0962C
0963C 010230
0964C
0965C 010240 ASK-QUESTION.
0966C
0967C 010250 DISPLAY CLEAR-MESSAGE.
0968C
0969C 010260 DISPLAY QUESTION-SCREEN.
0970C
0971C 010270 ACCEPT QUESTION-SCREEN.
0972C
0973C 010280 IF QUESTION-ANSWER IS NOT = "Y" AND
0974C

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

```

0975C 010290 QUESTION-ANSWER IS NOT = "N"
0976C
0977C 010300 GO TO ASK-QUESTION.
0978C
0979C 010310
0980C
0981C 010320 DISPLAY CLEAR-MESSAGE.
0982C
0983C 010330
0984C
0985C 010340 WHAT-NUMBER.
0986C
0987C 010350 IF WHAT-NUMBER-ANSWER = SPACE
0988C
0989C 010360 DISPLAY WHAT-NUMBER-SCREEN
0990C
0991C 010370 ACCEPT WHAT-NUMBER-SCREEN
0992C
0993C 010380 ACCEPT ESCAPE-CODE FROM ESCAPE KEY.
0994C
0995C 010390 IF ESCAPE-CODE IS NOT = ESCAPE-KEY
0996C
0997C 010400 INSPECT WHAT-NUMBER-ANSWER
0998C
0999C 010410 REPLACING LEADING SPACE BY ZEROS
1000C
1001C 010420 IF WHAT-NUMBER-CHAR (2) = SPACE
1002C
1003C 010430 MOVE WHAT-NUMBER-CHAR (1) TO WHAT-NUMBER-CHAR (2)
1004C
1005C 010440 MOVE ZERO TO WHAT-NUMBER-CHAR (1).
1006C
1007C 010450 IF ESCAPE-CODE IS NOT = ESCAPE-KEY
1008C
1009C 010460 IF WHAT-NUMBER-ANSWER NOT NUMERIC
1010C
1011C 010470 MOVE SPACE TO WHAT-NUMBER-ANSWER
1012C
1013C 010480 GO TO WHAT-NUMBER
1014C
1015C 010490 ELSE
1016C
1017C 010500 MOVE WHAT-NUMBER-ANSWER TO CHANGE-NO
1018C
1019C 010510 MOVE SPACE TO WHAT-NUMBER-ANSWER.
1020C
1021C 010520
1022C
1023C 010530 GET-DATES.
1024C
1025C 010540 ACCEPT SYSTEM-DATE FROM DATE.
1026C
1027C 010550 MOVE SYSTEM-DAY TO TODAYS-DAY.
1028C
1029C 010560 MOVE SYSTEM-MONTH TO TODAYS-MONTH.
1030C
1031C 010570 MOVE SYSTEM-YEAR TO TODAYS-YEAR.
1032C
1033
CROSS REFERENCE:
1 593 656 657 662 669 683 821 822
835 835 840 840 856 857 863 866

```

Figure C-6. Compilation Listing of the Report Writer Program (continued)

	869	872	875	878	881	884	887	890
2	893	895	897	1003	1005			
	568	600	660	842	1001	1003		
80	760	824						
ANY-CHANGE	919	953						
ANY-CHANGE-ANSWER								
	134	392	398	929	931	939	941	947
	949	951						
ANY-CHANGE-CHAR-1								
	135	937						
ANY-CHANGE-DEFAULT								
	137	931						
ANY-CHANGE-DEFAULT-SCREEN								
	396	933						
ANY-CHANGE-SCREEN								
	386	921	923					
APPLICATION-NAME								
	220	457	467	492				
ASK-QUESTION	965	977						
AT-END	159	595	607	644	736			
BEGIN	559							
BOUNDARY-ERROR	163							
CALL-ERROR-PROGRAM								
	565	582	590	620				
CALL-NEXT-PROGRAM								
	573	588	615					
CHANGE-NO	138	1017						
CHECK-FOR-CONTROL-BREAK								
	606	643						
CLEAR-MESSAGE	423	717	961	967	981			
CLEAR-PROMPT	437							
CLOSE-ERROR	167							
CLOSE-FILES	572	720						
CLOSE-PRINTER-FILE								
	721	808						
CLOSE-SORTFILE	567	741						
CONTROL-BREAK-1								
	645	645	651	651	655			
CS-20	10	14						
CURRENT-PRICE	103	627	634	681	683			
DESCRIPTION	101	679						
DETAIL-1-FIELD-1								
	352	673						
DETAIL-2-FIELD-1								
	356	676						
DETAIL-2-FIELD-2								
	358	678						
DETAIL-2-FIELD-3								
	360	680						
DETAIL-2-FIELD-4								
	362	363	682					
DETAIL-LINE-1	349							
DETAIL-LINE-1-LOGIC								
	636	671						
DETAIL-LINE-2	353							
DETAIL-LINE-2-LOGIC								
	638	674						
DETAIL-PRINT	338							
DISK-FULL	165	520	534	547				
DISPLAY-MESSAGE								
	710	747	787	804	955			
DUPLICATE-KEY	161							
END-KEY	150	587	773					

Figure C-6. Compilation Listing of the Report Writer Program (continued)

END-OF-REPORT	569							
ERROR-MESSAGE-LINE	422	714						
ESCAPE-CODE	142	587	772	773	925	927	935	993
	995	1007						
ESCAPE-KEY	144	927	935	995	1007			
F1-KEY	145							
F2-KEY	146							
F3-KEY	147							
F7-KEY	148							
F8-KEY	149							
FILE-ACCESS-ERROR-SCREEN	402	511	524	538	551			
FILE-ERROR-NAME	154	414	510	523	537	550		
FILE-ERROR-STATUS	153	410	509	522	536	549		
FILE-STATUS-CODE-TABLE	156							
FULL-PRINTER-TABLE	175	794						
GET-DATES	1023							
GET-TOP-NAME	752							
HARDWARE-ERROR	164	508	519	532	546			
HASH-W	177	715	716					
HEADER-1	273							
HEADER-2	282							
HEADER-INDEX	294	840	841	850				
HEADER-LINE	293	850						
HEADER-TABLE	292							
HEADERS	272	292						
HEADERS-TO-PRINT	296	841						
I-O-OK	158	564	581	586	729	754	756	
I-O-SECTION	692							
INSTNAME-ERROR	516							
INSTNAME-FILE	28	86	517	753	755	758		
INSTNAME-NAME	90							
INSTNAME-RECORD	89	757						
INSTNAME-STATUS	32	217	519	520	521	522	754	756
INVALID-KEY	160							
LEGEND-1	244							
LEGEND-2	253							
LEGEND-INDEX	265	835	836	847				
LEGEND-LINE	264	847						
LEGEND-TABLE	263							
LEGENDS	243	263						
LEGENDS-FROM-PAGE-2-ONWARD	268	838						
LEGENDS-TO-PRINT	267	759	823	836	838			
LINE-COUNT	342	813	828	853	857	897		
LINE-FEED	568	657	660	831	834	837	842	852
LIST-DATE	99							
LISTDATA	37	94	530	694	698	703	722	
LISTDATA-ERROR	529							
LISTDATA-IN-USE	696	707						
LISTDATA-KEY	41	98	628	629	677	737		
LISTDATA-RECORD-IN-USE	700	705	711					

Figure C-6. Compilation Listing of the Report Writer Program (continued)

LISTDATA-STATUS	42	120	532	533	534	535	536	581
	595	607	644	695	699	704	735	739
LISTING-RECORD	97							
MAIN-LOGIC	558							
MENU-SELECTION	186							
MESSAGE-FIELD	179	425	445	709	713	746	786	803
MESSAGE-SCREEN	420	957	959					
MESSAGE-WAIT-ENTRY								
	180	430						
MODE-ERROR	169							
MONTH-NAME	207	903						
MONTH-NAMES	202	206						
MONTH-TABLE	206							
MOVE-TOTALS-1	658	666						
NEW-PAGE	814	819	854					
NO-CONTIGUOUS-SPACE								
	174							
NO-DIRECTORY	172							
NO-FILE	168							
NUMBER-OF-LEGENDS-PAGE-1								
	344	759	823					
OPEN-ERROR	166	533						
OPEN-FILES	560	578						
OPEN-LISTDATA	580	693						
OPEN-LPT-PRINTER-FILE								
	768	781						
OPEN-SORTFILE	563	725						
OPEN-SPOOL-PRINTER-FILE								
	765	780	791	797				
ORIGINAL-PRICE	102	635						
OVER-LOCK-LIMIT								
	173	521	535	548				
PAGE-DEPTH	232	813	853					
PAGE-NUMBER	340	821	822	825	827			
PAGE-WIDTH	230	760	824					
PASS-FLAG	130	593	600					
PRINT	684	688	811					
PRINT-DETAIL	812							
PRINT-DETAIL-LINE-1								
	608	633						
PRINT-DETAIL-LINE-2								
	609	637						
PRINT-EXIT	818	858						
PRINT-HEADERS	840	849						
PRINT-LEGEND	835	846						
PRINT-LINE	346	350	354	366	815	817		
PRINT-LINE-SAVE								
	335							
PRINT-RECORD	816	830	833	848	851	860		
PRINT-RECORD-EXIT								
	864	867	870	873	876	879	882	885
	888	891	894	896				
PRINT-RECORDS	561							
PRINT-SYSTEM-DATE								
	579	900						
PRINT-SYSTEM-DATE-AND-TIME								
	899							
PRINT-SYSTEM-TIME								
	905							
PRINT-THE-LINES								
	566	592	604	610				
PRINT-TOTAL-1	363	667						

Figure C-6. Compilation Listing of the Report Writer Program (continued)

PRINTER-CLOSE	807						
PRINTER-DEVICE	305	783					
PRINTER-ERROR	505						
PRINTER-FIELD-1-1							
	60						
PRINTER-FIELD-109-120							
	70	865					
PRINTER-FIELD-121-132							
	71	862					
PRINTER-FIELD-13-24							
	62	889					
PRINTER-FIELD-2-12							
	61	892					
PRINTER-FIELD-25-36							
	63	886					
PRINTER-FIELD-37-48							
	64	883					
PRINTER-FIELD-49-60							
	65	880					
PRINTER-FIELD-61-72							
	66	877					
PRINTER-FIELD-73-84							
	67	874					
PRINTER-FIELD-85-96							
	68	871					
PRINTER-FIELD-97-108							
	69	868					
PRINTER-FILE	23	57	506	784	793	809	
PRINTER-NAME	24	304	775	782	792		
PRINTER-OPEN	585	751					
PRINTER-OPEN-EXIT							
	776	790	801	805			
PRINTER-RECORD	59	815	829	832	844	847	850 855
	863						
PRINTER-RECORD-1							
	73	845	856	895			
PRINTER-RECORD-108							
	82	869					
PRINTER-RECORD-12							
	74	893					
PRINTER-RECORD-120							
	83	866					
PRINTER-RECORD-24							
	75	890					
PRINTER-RECORD-36							
	76	887					
PRINTER-RECORD-48							
	77	884					
PRINTER-RECORD-60							
	78	881					
PRINTER-RECORD-72							
	79	878					
PRINTER-RECORD-84							
	80	875					
PRINTER-RECORD-96							
	81	872					
PRINTER-STATUS	25	302	508	509	586	785	794
PROCESS-INSTNAME-ERROR							
	518						
PROCESS-LISTDATA-ERROR							
	531						
PROCESS-PRINTER-ERROR							
	507						

Figure C-6. Compilation Listing of the Report Writer Program (continued)

PROCESS-SORTFILE-ERROR	545								
PROMPT-LINE	185	439							
PROMPT-SCREEN	435								
QUESTION-ANSWER	182	449	494	778	973	975			
QUESTION-SCREEN	443	969	971						
READ-LISTDATA-RECORD	702	706	738						
READ-NEXT-LISTDATA-RECORD	594	697	701						
READ-NEXT-SORTFILE-RECORD	605	733	740						
RECORD-LOCKED	170	699	704						
RECORD-NOT-FOUND	162	739							
RECORD-ON-FILE	157								
REPORT-TITLE	235	832							
SCAN-PRINT-LINE	861								
SELECTION-TYPE	187								
SORTFILE	47	107	544	596	597	599	727	728	
	734	742	743						
SORTFILE-DATA	115								
SORTFILE-ERROR	543								
SORTFILE-IN-USE	730	744							
SORTFILE-KEY	51	110	598	599					
SORTFILE-KEY-1	111	626							
SORTFILE-KEY-2	112	627							
SORTFILE-KEY-3	113	628							
SORTFILE-KEY-KEY	114	629	737						
SORTFILE-LINE-NO	128	726							
SORTFILE-NAME	48	126							
SORTFILE-RECORD	109	732							
SORTFILE-STATUS	52	125	546	547	548	549	564	729	
	735	736							
SPOOL-CODE	224	764	767	796					
SPOOL-LINE-NUMBER	228	779							
SPOOL-OPTION	769	788	799						
SPOOL-OPTION-END	495	774							
SPOOL-OPTION-SCREEN	490	770	771						
SPOOLER-ACCESS-TABLE-FULL	795	802							
SPOOLER-FILE-NAME	226	510	792						
SYSTEM-DATE	190	901	1025						
SYSTEM-DAY	193	902	1027						
SYSTEM-HOUR	210	909							
SYSTEM-HUNDRETH	213								
SYSTEM-MINUTE	211	908							
SYSTEM-MONTH	192	903	1029						
SYSTEM-SECOND	212								
SYSTEM-TIME	209	906							

Figure C-6. Compilation Listing of the Report Writer Program (continued)

SYSTEM-YEAR	191	904	1031				
TODAYS-DATE	195	197					
TODAYS-DATE-REDEFINE	197						
TODAYS-DAY	199	1027					
TODAYS-MONTH	198	1029					
TODAYS-YEAR	200	1031					
TOP-81-132	333	763					
TOP-COLON	319	907					
TOP-DAY	310	902					
TOP-HOUR	318	909					
TOP-LINE	308	328	335	829			
TOP-LINE-80	328						
TOP-MINUTE	320	908					
TOP-MONTH	312	903					
TOP-NAME	322	757	761				
TOP-NAME-80	330	761					
TOP-NUMBER	327	827					
TOP-NUMBER-80	332	825					
TOP-OF-FORM	570	820	843				
TOP-PAGE-80	331	762					
TOP-TIME	315	316					
TOP-TIME-REDEF	316						
TOP-YEAR	314	904					
TOTAL-1	378	667	669	669	683		
TOTAL-1-FIELD-1	368	687					
TOTAL-KEY-1	371	646	647	649	661		
TOTAL-KEYS	370						
TOTAL-LINE-1	365						
TOTAL-LINE-1-LOGIC	659	685					
TOTAL-MAX	375	668					
TOTAL-SUB	374	656	667	668	669	669	
TOTALS	376						
TOTALS-LEVEL	377	662					
TOWN	100	626	646	648	661	675	
ULINES	184						
USE-ERROR	171	695	785				
UTILITIES	915						
WAIT-LINE-24	473	616	621				
WAIT-PRINTING	453	601					
WAIT-PROCESSING	463	562	571				
WHAT-NUMBER	985	1013					
WHAT-NUMBER-ANSWER	139	485	939	945	987	997	1009
	1017	1019					1011
WHAT-NUMBER-CHAR	140	1001	1003	1003	1005		
WHAT-NUMBER-SCREEN	481	989	991				
WRITE-SORTFILE	603	625					
WRITE-SORTFILE-RECORD	630	731					

Figure C-6. Compilation Listing of the Report Writer Program (concluded)

End of Appendix

Appendix **D**

To the Program Operator

This appendix is designed to introduce you to the PROXI system. We've gathered in one place all the general information that you'll need - you don't have to read through the earlier sections of this manual.

PROXI programs are custom made. Therefore you'll have to get the details about a particular program from the person who built it. PROXI programs follow a basic format that is easy to use and often self-explanatory. After you've run a PROXI program a few times you probably won't need any reference material.

The Keyboard

PROXI programs often require you to use special function keys. We provide a terminal keyboard template which you place over the row of function keys at the top. There are two different templates: one for the D2 keyboard and the other for the D200. You'll need only one of these templates for your terminal. See Figure D-1. The only keys you will need to use are the NEXT RECORD and the FINISHED keys. We'll explain the use of these keys shortly.

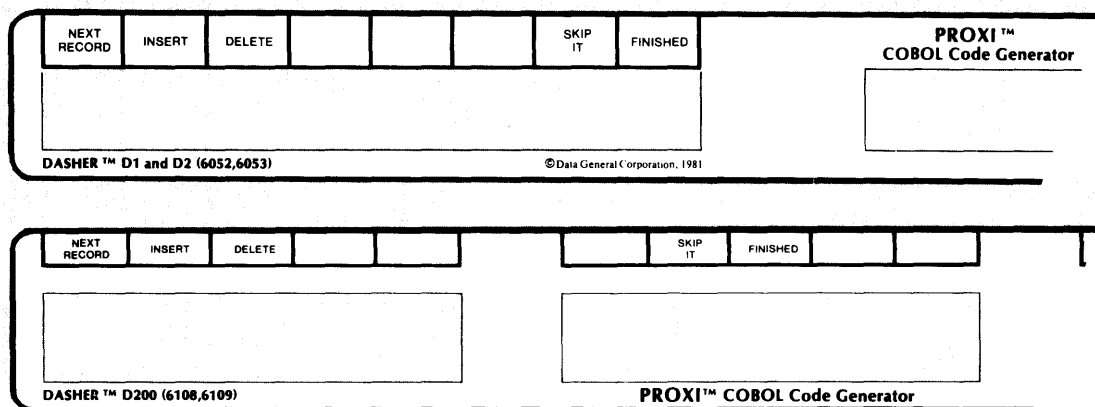


Figure D-1. The PROXI Terminal Templates

The Screen

You communicate with the PROXI program through the terminal screen. The program's data-entry screens list prompts or questions to which you must supply a value, a name, or a "Yes or No" response. When the system is awaiting an entry, the cursor will position itself at the beginning of the appropriate field. Enter your response; the cursor will then move to the next field (if any), or you will receive the next data-entry screen.

Making Entries

When responding to a prompt you should know the range of values that the field may contain, and the field's default value, if any. The programmer who developed the program should provide you with this information either in written form (program documentation) or on the screen itself (help messages). Help messages appear on the last line of the screen.

If you make an invalid entry, that is, one that does not conform to what the program expects, your response will disappear and the cursor will position itself once again at the beginning of the field. Note that the program may not recognize an *incorrect* entry. For example, if you misspell a customer's name, the program will probably be unable to detect the error. If you notice that you've made an error after entering the field, you still have a chance to correct it. You can do this through the Any Change cycle which we describe later.

Default Values

Earlier we mentioned default values. These are entries that the program automatically supplies when you do not respond to a question and only press NEW LINE. This feature can save you time, particularly if the default values are the values you most often use. In some cases you can press a series of NEW LINES without actually keying in a single entry. The default value sometimes appears in the entry field. If you don't want to use the default value, you must overwrite it with another response.

Field Characteristics

Certain fields have special characteristics that affect the way you enter information. For example, some fields are "required": you *must* supply an answer (or take the default, if one exists). If you do not make an entry to such a field, the cursor will not move until you supply a response. This characteristic could apply to a field requesting a customer's name. If you don't supply a name the program will not be able to locate the appropriate record in the data file.

Other fields must be *completely* filled in before the program will accept your entry. A program requesting a customer's account number won't accept a partial entry; it needs every digit of the number.

Some programs use "Secure" fields. When you make a response to this type of field, your entry appears as one or more asterisks. This prevents privileged information from appearing on the screen where unauthorized persons may see it.

When changing information on a record, you may find that certain fields cannot be modified. If you try to alter this type of field you will receive an error message. Move on to the next field.

You will be able to edit certain fields; you must replace others.

Controlling the Cursor

When you are making entries to screen fields, you may use a variety of cursor control keys and key combinations to position the cursor. The following tables list these control keys. Note that you work on one field at a time within a screen format. Once you complete a field, you'll move on to the the next. You cannot return to a previous field until you complete the entire screen. See the description of the Any Change Cycle (below) for details about modifying a previous entry.

Table D-1. Control and Editing Characters

Key	Action
NEW LINE	Moves the cursor to the beginning of the next line.
CR	Blanks the characters to the right of the cursor and moves the cursor to the beginning of the next line.
TAB	(Same as NEW LINE.)
ERASE PAGE	(Same as NEW LINE.)
→	Moves the cursor right one column.
←	Moves the cursor left one column.
↑	Moves the cursor to the beginning of the current input field.
ESC	Restores the screen to its former state. Newly created screens are cleared. Modified screens appear as they did before this set of changes.

The Types of PROXI Program

There are two main types of PROXI program: the printing program and the interactive program.

The Printing Program

PROXI printing programs generate reports or forms. These programs require very little of your attention; they do most of the work on their own.

Key Range Selection

The program may, however, request that you supply a range of keys to identify which records in the data file you want to process. For example, you may want to print a set of invoices for only the first fifty orders that are on file. You provide a starting key and an ending key, naming the first and last possible records to be output by the printing program.

After you specify a key range, the program will again prompt you to enter another range. This feature allows you to print selected portions of a data file. If you do not want to print any more records, press the FINISHED key to terminate the program.

If you want to process the entire file, simply press NEW LINE when the program asks for each key. The printing program will then print the complete data file.

Output Method

You may be asked to specify how the program should output its reports or forms. There are two choices: either direct the output to the printer, or spool it to a disk file for printing at a later time. If you attempt to use the printer, but it is unavailable, you'll see the message

PRINTER IN USE

OUTPUT TO THE PRINTER?

Enter Y if you want to retry the printer; otherwise, enter N to have the output spooled to a disk file. If you press FINISHED in response to the OUTPUT TO THE PRINTER question, the program will terminate and return you to the calling program (or the CLI).

The Interactive Program

PROXI Interactive programs allow you to work with records in a data file. Some programs, called File Maintenance Programs, allow you to add new records, change existing records, delete records, and print the contents of the data file. You perform one function for a series of records, then press FINISHED to return to the function menu. At this point you may select another function, or press FINISHED again to terminate the program.

Other programs allow you to display data-file information, but not to change it; these are called File Inquiry Programs.

All interactive programs use screen formats to communicate with you. File Maintenance Programs begin by displaying a menu of functions. Figure D-2 illustrates a typical menu.

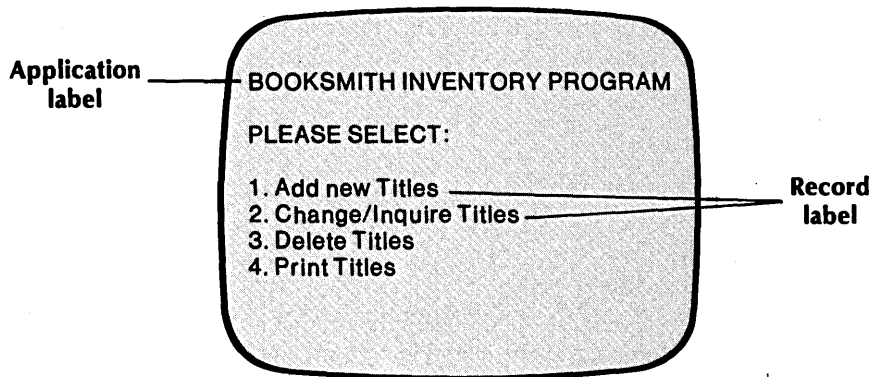


Figure D-2. A Typical File Maintenance Program Menu

As you can see, you have four choices. To enter your selection, simply press the appropriate number depending on the operation you want to perform. The program will execute your request, but first it may need additional information. Another screen may appear asking for information that identifies a particular record within the data file. It may include a number of prompts to which you make appropriate entries. See Figure D-3 for an example.

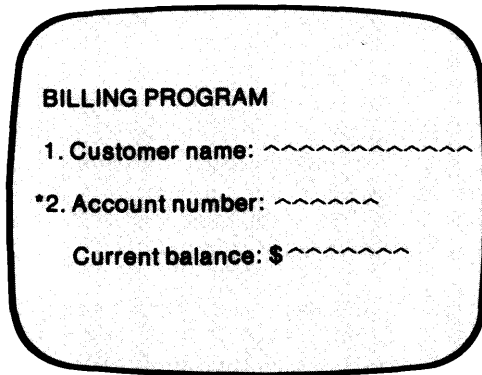


Figure D-3. A Sample Screen Format

This screen requests you to enter the account number, name, and order amount for a customer. Notice that each prompt has a number preceding it. The account number prompt also has an asterisk before the number. Whenever you see an asterisk before a prompt it indicates that your response to the question is what the program uses to locate the record in the data file. In this case the file must be organized by account number.

The Any Change Cycle

The Any Change cycle is a PROXI feature that asks you to confirm your entries before passing from one screen to the next. The question

ANY CHANGE ?

appears at the bottom of the screen. (See Figure D-4.) If you are satisfied with your answers, enter N; you'll move on to the next screen.

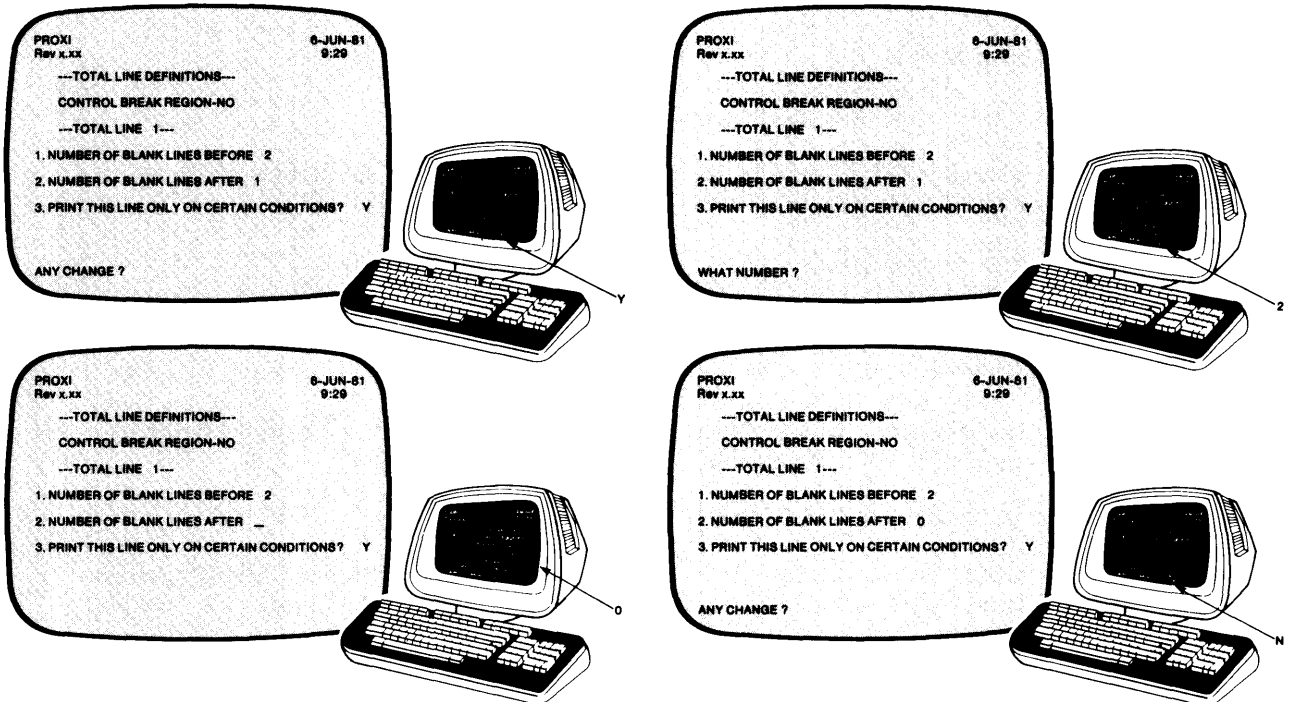


Figure D-4. The Any Change Cycle

If you want to make a change, however, enter Y. The program will then ask

WHAT NUMBER ?

Enter the number of the entry you want to change. The cursor will move to the beginning of the field, allowing you to make any modifications. When you press NEW LINE the "Any Change" question appears once again. You can continue to modify your entries until you respond N to this question. ("No" is the default answer, so you may press NEW LINE alone.) Use uppercase characters when entering Y or N.

You may enter the prompt number immediately following the "Any Change" question, thereby skipping the "What Number" question. Note that certain screens may be so brief that the entry fields do not have prompt numbers. If you want to change an entry in this case, respond Y to the "Any Change" question, then re-enter all fields.

The NEXT RECORD Key

We've already explained that an asterisk preceding a prompt indicates a field used to locate a record in the data file. To make your job easier, we provide the NEXT RECORD key. If you want to change or delete data-file records sequentially, you can press this key instead of entering the next key field. Note that this is only for sequential operations. If you need to move around the file, working with scattered records, this key will not help you.

Operator Error Messages

This section lists the error messages you may receive while operating a PROXI program.

CHANGES NOT ALLOWED TO THESE FIELDS

The field you tried to modify will not accept changes. Move on to the next field.

datafile FILE IN USE or

datafile FILE IN USE - TRY AGAIN LATER

The program cannot gain access to the indicated data file because another program is currently using it. You may try to run this program at another time.

datafile RECORD DELETED

The program has performed the DELETE function on the specified record.

datafile RECORD IN USE

The program cannot gain access to the specified record because another program is using it. Move on to another record or function and come back to this record later.

ERROR: RECORD ALREADY ON FILE

You tried to add a record to a data file, but a record already exists with the RECORD KEY you used. You may want to delete the existing record and replace it, or you may only need to modify it.

IS THIS THE RIGHT RECORD ?

Before you delete a record, the program gives you one last chance to change your mind. Answer N if you want to cancel the delete operation.

NO RECORDS WITHIN RANGE SELECTED

The specified range within the data file contains no records. You may want to adjust the range or use the entire data file.

NOT PROCESSED

The program did not perform the operation you canceled.

RECORD NOT ENTERED

You pressed the ESC key while entering information for a new record. The program ignores your entries, does not alter the file, and awaits a new set of entries.

RECORD NOT ON FILE

You specified a record that does not exist on this data file. Check the RECORD KEY you supplied.

TYPE CR TO CONTINUE

This message usually follows another message. Press the NEW LINE or CR key to re-enter the field that caused the error.

End of Appendix

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

Appendix **E**

Screen Format Parameters

As we describe in Chapter 3, you build a parameter file for each screen format that a PROXI program will use. After completing the parameter file you must build two copy files, one for the Screen Section and the other for the Procedure Division. If you make any changes to the original screen format parameters (such as altering the screen image or modifying a field's definition) you will have to rebuild at least one of the copy files.

This appendix lists the screen format parameters and indicates which of the copy files uses each parameter. Using this list, you may find that you need to rebuild only one of the copy files, (assuming that the changes you made to the screen format affect only one copy file).

Remember that if you make any changes to the screen format parameter file (screen.SF), you may have to rebuild the corresponding Screen Section and Screen Procedure copy files. Certain parameters, however, apply to only one of these files. If your changes to the screen format are limited to particular entries, you may have to rebuild only one of these files. The following table, Table E-1, lists the field definition parameters that are used in the Screen copy files.

Table E-1. Screen Format Parameters

Screen Field Parameter	Used by .SD File	Used by .PL File
SCREEN ID	X	X
ENTRY SEQ #	X	X
NAME OF FIELD	X	X
TYPE OF DATA	X	
DISPLAY FORMAT	X	
CHANGE METHOD		X
BELL	X	
BLINK	X	
REQUIRED	X	
FULL	X	
SECURE	X	
PROMPT LINE		X
DEFAULT VALUE		X
VALIDATION		X
OWN CODE		X

If you have changed the contents of an Own Code paragraph or section, but not the name, the change will not affect the .PL copy file. Including Own Code where there was none, removing it, or renaming the section or paragraph will require you to redefine the parameters in the screen format file and to rebuild the screen procedure file.

End of Appendix

Appendix **F**

Using CRT Format PROXI Code

This appendix provides additional information for those who need to produce PROXI code in CRT, not card, format. The PROXI system uses card format, but allows you to create copy files and the main program framework in CRT format. Note, however, that the program framework uses COPY statements to incorporate prepared PROXI copy files into its structure. Because all files provided by PROXI use card format, you must remove the line numbers before you incorporate them into your program through the Program Generator.

You must alter the following PROXI files:

PROXI\$CF01.PL	PROXI\$CF04.DS
PROXI\$CF01.SD	PROXI\$CF04.FD
PROXI\$CF01.WS	PROXI\$CF04.PL
	PROXI\$CF04.SD
PROXI\$CF02.PL	PROXI\$CF04.SL
	PROXI\$CF04.WS
PROXI\$CF03.DS	
PROXI\$CF03.FD	PROXI\$CF05.PL
PROXI\$CF03.PL	
PROXI\$CF03.SD	PROXI\$CF06.DS
PROXI\$CF03.SL	PROXI\$CF06.FD
PROXI\$CF03.WS	PROXI\$CF06.PL
	PROXI\$CF06.SD
	PROXI\$CF06.SL
	PROXI\$CF06.WS

You may use any method available to remove the line numbers from the files listed. If your system includes AOS SPEED software, you can create a macro file to do this. We show how to do that here.

The first step is to create a file containing the SPEED command. Type the following CLI commands:

```
) CRE /I CRT.SCF)
)) VN<6DT1L>FB)
)) )
```

Now you have a file called CRT.SCF which contains the SPEED command. The next step is to create the macro that will call SPEED and execute the command. Enter the following CLI commands:

```
) CRE /I CRT.CLI)#  
) X SPEED /I=CRT.SCF %1%  
) RENAME %1%.BU %1%.CARD)  
) )
```

To use this macro, simply type CRT followed by the name of the file you want to remove line numbers from. For example,

```
) CRT PROXI$CF03.SD)
```

The macro will execute the SPEED command that strips the line number (the first six characters) from each line, and displays each line on the screen. When the operation is complete, the file PROXI\$CF03.SD will contain the PROXI code *without* line numbers, (i.e., in CRT format). The macro creates a backup file called filename.CARD which retains the original card format code. You should keep the card format code in case you need to create a card format program at a later time.

End of Appendix

Index

Within this index, the letter "f" following a page entry means "and the following page"; "ff" means "and the following pages." If you need a definition of a PROXI term, see Appendix A.

- * (asterisk),
 - comment 4-18f, 4-24f
 - conditional field 2-35, 2-59, 2-76, 2-90f, 2-100f, 2-108f, 2-116f
 - record key field 3-4f, 3-16f, D-5
- ^ (caret) 3-3f, 3-16f
- " (quotation marks),
 - when illegal 3-3, 3-17
- with nonnumeric literal 2-57, 2-61, 2-93, 2-95, 2-103, 2-105, 2-111, 2-119, 2-121, 3-25, 3-31
- / (slash),
 - editing character 2-50, 2-52, 3-18
 - form feed character 3-59, 4-18f, 4-24, 4-33

A

- access field, reference file 2-44f, 2-86f
- Access Method,
 - Data Base (DBAM) 1-2
 - Indexed Sequential (ISAM) 1-2, 2-7, 2-10, 2-21, 2-23, 2-37
- ACCESS MODE clause 4-10f
- accumulator,
 - auto-total 2-36f, 2-78
 - requiring own code 1-2, 1-10
 - specifying 2-36f, 2-62f, 2-70f, 2-78, 2-94f, 2-104f, 2-112f, 2-120f
- adding program parameters 2-126

- adding a record 1-3, 2-7, D-4
- adding a screen format 3-1, 3-11, 3-14ff
- alphanumeric field,
 - how represented 1-8
 - identifying 3-24f, 3-28f
- alternate key 1-2
- Any Change cycle 1-8f, 2-19, 2-27, 2-51, 2-53, 2-125f, 3-4, 3-23, 3-38, 3-40, D-5f
- application name (label) 2-4f, 2-9
- ASSIGN TO clause 4-10f
- asterisk,
 - comment 4-18f, 4-24f
 - conditional field 2-35, 2-59, 2-76, 2-90f, 2-100f, 2-108f, 2-116f
 - record key field 3-4f, 3-16f, D-5
- attribute,
 - full field 3-7, 3-24f, D-2
 - required field 3-7, 3-24f, D-2
 - secure field 3-7, 3-26, D-2
- automatic totaling 2-36f, 2-62f, 2-70f, 2-78, 2-104f, 2-120f
- auto-total accumulator 2-36f, 2-78

B

- beginning a PROXI session 1-12
- bell field 3-7, 3-28f
- binding a program 5-1
- blanking a screen 3-5f, 3-20f, 3-38
- blinking field 3-7, 3-28f
- break, control 2-33, 2-36f, 2-66ff, 2-75, 2-77, 2-114ff
- break line, page 2-75f, 2-77, 2-82, 2-108ff
- building a program 1-10, 2-9ff, 2-22ff, 2-38ff, 2-79ff

C

card format 1-10, 1-12, 3-61, 5-1, F-1f
specifying 2-18f, 2-30f, 2-72f, 2-122f, 2-129f, 3-50f,
3-54f, 4-8f, 4-14f, 4-22f, 4-28f
calculations (See "computations".)
calling a printing program 1-3, 2-7, 2-10, 2-15
caret 3-3f, 3-16f
change method (by operator) 3-7, 3-24f
changing
a datafile copy file 4-2, 4-15
a program 1-11, 2-2, 2-19, 2-40, 2-82, 2-125ff
a record (by operator) 1-3, 2-7, 2-9, D-4
a screen format 3-1f, 3-9, 3-10f, 3-36ff, E-1f
characters,
for naming files 1-11f
using uppercase 1-9, 1-11, 2-5
clause,
ACCESS MODE 4-10f
ASSIGN TO 4-10f
codes 4-19f (FD), 4-25f (Working Storage)
DATA SIZE 4-10f
FILE STATUS 4-10f
INDEX SIZE 4-10f
LABEL RECORDS 4-16f
ORGANIZATION 4-10f
RECORD KEY 1-3, 4-10f
RECORDING MODE 4-16f, 5-1
.CO file 1-12, 2-1
COBOL, Interactive (CS) 4-17
code, clause 4-19f (FD), 4-25f (Working Storage)
code,
compiling source 1-1, 1-10, 2-19, 2-123, 5-1
generating COBOL 1-10, 2-1f, 2-10, 2-12, 2-18f,
2-23, 2-24, 2-30f, 2-40, 2-72f, 2-82, 2-122f,
2-125, 2-128f, 3-2
Column 999 field 2-40, 2-82
specifying 2-62f, 2-70f, 2-94f, 2-104f, 2-112f,
2-120f
command format conventions v
comment line 4-19 (FD), 4-24 (Working Storage)
compiling source code 1-1, 1-10, 2-19, 2-123, 5-1
computations 2-33, 2-36, 2-40, 2-75f 2-78, 2-82
specifying 2-62f, 2-70f, 2-94f, 2-104f, 2-112f,
2-120f
condition, end-of-file 2-36, 2-66f, 2-77
conditional field 2-35f, 2-59, 2-76, 2-90f, 2-101, 2-109,
2-116f
identifying 2-35, 2-59, 2-76, 2-90f, 2-100f, 2-116f
conditional printing,
field 2-35f, 2-59, 2-76, 2-90f, 2-101, 2-109, 2-116f
line 2-33, 2-35, 2-37, 2-40, 2-75ff
detail 2-58ff, 2-100ff
page break 2-108ff
top-of-form 2-90f
total 2-116ff
constant (See "literal field".)
control break 1-4, 2-33, 2-36f, 2-40, 2-75, 2-77, 2-82
defining 2-66f, 2-114f

conventions,
command format v
file-naming 1-3, 1-11f
copy file,
data-file 1-10, 1-12, 4-1ff
declaratives 4-28ff
file descriptor entry 4-1f, 4-4f, 4-14ff
Own Code 1-1, 1-12, 2-1, 2-12, 2-14f, 2-23ff, 3-7,
3-40, 5-1
screen format 1-10, 3-1f, 3-9
screen procedure 1-10, 1-12, 2-10, 2-27, 3-9, 3-40,
3-44f, 5-1
screen section 1-10, 1-12, 2-10, 2-27, 3-9, 3-11,
3-40, 3-50f
SELECT statement 1-12, 4-8ff
COPY statement 1-1f, 1-11f, 2-1, 2-9f, 2-12, 2-21,
2-23f, 3-8, 4-1
correcting a mistake 1-8
counter, page number 2-59, 2-67, 2-69, 2-75, 2-78,
2-101
creating,
data-file copy files 4-1ff
a program 1-10, 2-2, 2-3ff
a screen format 1-10, 1-13, 2-10, 3-14ff
a screen procedure copy file 3-9, 3-54f
a screen section copy file 3-9, 3-50f
CRT format 1-10, F-1f
specifying 2-18f, 2-30f, 2-72f, 2-122f, 2-128f, 3-50f,
3-53f, 4-8f, 4-14f, 4-22f, 4-28f
CS PROXI 4-11
cursor control 1-5f, 2-51, 2-53, 3-17ff, D-2

D

D2 keyboard 1-4, D-1
D200 keyboard 1-4, D-1
Data Base Access Method (DBAM) 1-2
Data Dictionary 1-3, 1-11, 2-3, 2-14ff, 2-27, 2-29, 2-43,
2-45, 2-63, 2-71, 2-85, 2-87, 2-95, 2-105,
2-113, 2-121
data-entry screen 1-1f, 1-7, 2-1, 2-7
datafile,
copy file 1-10, 2-10, 4-1ff
definition 1-13, 4-1ff
principal 1-2, 1-4, 1-10, 2-7, 2-10, 2-21, 2-23, 2-33,
2-35ff, 2-37, 2-75, 2-79, 2-82, 2-87
identifying 2-14, 2-26f, 2-42f, 2-84f
record name 2-9, 2-12, 4-19
data name 4-18f, 4-24f
DATA SIZE clause 4-10f
DBAM 1-2
Declaratives Section 1-10, 1-12f, 2-10, 4-1f, 4-5
creating 4-28ff
default data name 1-3, 2-3, 2-15, 2-27, 2-43, 2-85
default print format 2-63, 2-70, 2-94, 2-105, 2-113,
2-121
defining fields,
print 2-62f, 2-70f, 2-94, 2-104f, 2-112f, 2-120f
screen 3-6ff, 3-22f, 3-24ff, 3-28f

- defining a program 2-3ff, 2-7ff, 2-21ff, 2-33ff, 2-75ff
- defining a screen format 1-10, 2-21, 3-1f, 3-14ff, 3-36ff
- DELETE key 1-5, 2-40, 2-51, 2-53, 2-82, 2-126, 3-9, 3-40
- deleting program parameters 2-126
- deleting a record 1-3, 2-7, D-4
- detail line 1-4, 2-33ff, 2-40ff, 2-58ff, 2-75f, 2-77f, 2-82f, 2-96ff
- dictionary, Data 1-3, 1-11, 2-3, 2-14ff, 2-27, 2-29, 2-43, 2-45, 2-63, 2-71, 2-85, 2-87, 2-95, 2-105, 2-113, 2-121
- display field 2-9, 2-21, 3-3, 3-7
 - defining 3-28f
 - specifying 3-22f
- displaying a record 1-4, 2-7, D-4
- documenting a program 5-2
- .DS file 1-10, 1-12, 2-14, 2-27, 2-43, 4-28ff, 4-33
- dynamic access 4-10f

E

- edit method (by operator) 3-25
- editing keys 1-5, 2-50, 2-52, 3-17f, D-2
- editing PROXI files 1-11, 4-2, 5-1
- end-of-file condition 2-36, 2-66f, 2-77
- ending key,range 2-37, 2-79, 3-60f, 4-34f
- entry field 2-9, 2-21, 3-3f, 3-7f, 3-39
 - specifying 3-22ff
- entry sequence number 3-4f, 3-7, 3-22f
- EOF (as control break) 2-66f
- error messages 1-7, B-1ff
 - operator D-6f
- ESC key 1-6, 1-8, 2-51, 2-53, 3-17f
- evaluating a logical test series 2-37, 2-79, 3-8
- extension, filename 1-11f, 2-5

F

- FD entry (See "File Descriptor entry".)
- .FD file 1-3, 1-10, 2-14f, 2-27, 2-29, 2-40, 4-2, 4-4f, 4-14f, 4-33, 5-1
- field, alphanumeric 1-8, 3-24f, 3-28f
 - bell 3-7, 3-28f
 - blinking 3-7, 3-28f
 - characteristics 3-7, 3-25, 3-29, D-2
 - Column 999 2-40, 2-62f, 2-70f, 2-82, 2-94f, 2-104f, 2-112f, 2-120f
 - conditional 2-35f, 2-59, 2-76, 2-90f, 2-100f, 2-109, 2-116f
 - control break 1-4, 2-33, 2-36f, 2-40, 2-66ff, 2-75, 2-77, 2-82, 2-114ff
 - display 2-9, 2-21, 3-3, 3-7, 3-22f, 3-28f
 - format 2-9, 2-21, 3-3, 3-7, 3-22f, 3-28f
 - full attribute 3-7, 3-24f, D-2
 - key 2-7f, 2-21, 2-37, 2-42f, 2-46f, 2-84f, 2-86f, 3-4, 3-16f, 4-10f, 4-23, 4-25
 - numeric 1-8, 3-24f, 3-28f

- print 2-33ff, 2-76ff
- prompt line (operator) 3-7, 3-24f
- record identifying 1-4, 2-7f, 2-12, 2-21, 2-37, 2-42f, 2-46f, 2-84f, 3-4, 3-16f, 4-10f, 4-23, 4-25
- required attribute 3-7, 3-24f, D-2
- screen 3-3f, 3-6f, 3-22ff, 3-28f
- secure attribute 3-7, 3-26, D-2
- unnumbered variable 3-22f
- variable 1-8, 2-9, 2-21, 3-3f, 3-6, 3-16f, 3-23ff

file,

- .CO 1-12, 2-1
- copy 1-1, 1-3, 1-10ff
- data, definition 1-13, 4-1ff
- data, record name 2-9, 2-12, 4-19
- Definer 1-8, 1-10, 1-13, 2-1, 2-3, 2-27, 2-38, 2-79, 4-1ff
- Definitions menu 1-6, 4-4f
- .DS 1-10, 1-12, 2-14, 2-27, 2-43, 4-28ff, 4-33
- editing a PROXI 1-11, 4-2, 5-1
- end of 2-36, 2-67, 2-77
- .FD 1-3, 1-10, 2-14ff,, 2-27, 2-29, 2-40, 4-2, 4-4f, 4-14f, 4-33, 5-1
- Inquiry program 1-1f, 1-4, 2-3ff, 2-21ff, 3-1, 3-4, 3-8f
- Maintenance program 1-1ff, 2-3ff, 2-7ff, 2-21, 3-1, 3-4, 3-8f, 3-17
- naming conventions 1-3, 1-11f
- parameter 1-2f, 1-10f, 2-1f, 2-14f, 2-23, 2-27, 2-125f
- .PL 1-12, 2-12, 2-14f, 2-24, 2-27, 3-33, 3-55, 3-59
- .PP 1-12, 2-1, 2-14, 2-27, 2-125, 2-129
- principal data 1-2, 1-4, 1-10, 2-7, 2-21, 2-23, 2-33, 2-35ff, 2-37, 2-42f, 2-75, 2-79, 2-82, 2-84f, 2-87, 2-115
- reference 1-2, 2-10, 2-14ff, 2-21, 2-23, 2-28f, 2-33, 2-37, 2-42ff, 2-75, 2-79, 2-82, 2-84ff, 2-125
- screen copy, printing 3-1f
- screen definition copy 1-10, 3-1f
- screen format parameter 1-11, 3-1, 3-15ff, E-1f
- screen procedure copy 1-10, 1-12, 2-10, 2-27, 3-9, 3-40, 3-40, 3-44f, 5-1, E-1f
- screen section copy 1-10, 1-12, 2-10, 2-27, 3-9, 3-11, 3-40, 3-50f, 5-1, E-1f
- .SD 1-12, 2-15, 2-27, 3-2, 3-51, 3-59
- .SF 1-12, 3-1f, 3-9, 3-15, 3-40, E-1f
- .SL 1-3, 1-10, 1-12, 2-14ff, 2-27, 2-29, 2-43, 3-37, 4-1f, 4-4ff, 4-8ff, 4-33
- status item 4-11, 4-23, 4-25
- .WS 1-10, 1-12, 2-12, 2-14f, 2-24, 2-27, 2-43, 4-2, 4-4ff, 4-22ff, 4-33
- File Definer (See "File Definitions module".)
- File Definitions menu 1-6, 4-4f
- File Definitions module 1-1f, 1-6, 1-8, 1-10, 1-13, 2-1, 2-3, 2-27, 2-38, 2-79, 4-1ff
- file descriptor entry 1-3, 1-10, 1-12f, 2-3, 2-10, 2-14f, 2-27, 2-40, 4-1f, 4-4f, 5-1
- building 4-14ff

File Inquiry program 1-1f, 1-4, 2-3ff, 2-21ff, 3-1, 3-4, 3-8f
 building 2-22ff
 overview 2-21
 parameters 2-23

File Maintenance program 1-1ff, 2-3ff, 2-7ff, 2-21, 3-1, 3-4, 3-8f, 3-17
 building 2-9ff
 overview 2-7f
 parameters 2-10ff

file-naming conventions 1-3, 1-11f

FILE STATUS clause 1-3, 4-10f

FILESTATUS item 4-10f, 4-23, 4-25

FINISHED key 1-5, 2-5, 2-17, 2-29, 2-40, 2-51ff, 2-82, 2-126, 3-9, 3-25, D-1

FIRST\$NAME 1-12

flag, Own Code status 2-12, 2-24

form feed,
 after total line 2-66f
 at page end 2-49
 before print line 2-58f, 2-68f, 2-100f
 in listing 3-59, 4-19, 4-24, 4-33

form input 2-79

form layout 1-4, 2-75f, 2-82

Form Printing program 1-1f, 1-4f, 2-3ff, 2-15, 2-75ff, 2-126
 building 2-79ff
 overview 2-75ff
 session 2-82

format,
 card 1-10, 1-12, 2-18f, 2-30f, 2-72f, 2-122f, 2-129f, 3-50f, 3-54f, 3-61, 4-8f, 4-14f, 4-22f, 4-28f, 5-1, F-1f
 CRT 1-10, 2-18f, 2-30f, 2-72f, 2-122f, 2-128f, 3-50f, 3-53f, 4-8f, 4-14f, 4-22f, 4-28f
 default print 2-63, 2-70, 2-94, 2-105, 2-113, 2-121
 field 2-9, 2-21, 3-3, 3-7, 3-22f, 3-28f
 report 1-4, 2-33f, 2-40, 2-48ff
 screen 1-10f, 1-13, 2-9f, 2-12, 2-14, 2-21, 2-23f, 3-1ff, 3-9, 3-11, 3-14ff, 3-36ff, 3-42ff, E-1f

full field attribute 3-7, 3-24f, D-2

function key 1-4f, 2-7, 2-40, 2-82, 3-9, D-1

G

generating
 COBOL code 1-10, 2-1f, 2-10, 2-12, 2-17ff, 2-24, 2-30f, 2-40, 2-72f, 2-82, 2-122f, 2-125, 2-128f, 3-2
 screen copy files 3-9

Generator, Program 1-3, 1-8, 1-10f, 1-13, 2-1ff

H

heading line 1-4, 2-33ff, 2-40, 2-48f, 2-52ff

hierarchy of evaluation 2-37, 2-79, 3-8

how to begin a session 1-12

I

ICALL 5-1

identifying a record 1-4, 2-7f, 2-12, 2-21, 2-42f, 2-84f, 2-86f, 3-4

INDEX SIZE clause 4-10f

indexed organization 1-2, 2-43, 4-10f, 4-25, 5-1

Indexed Sequential Access Method (ISAM) 1-2, 2-7, 2-10, 2-21, 2-23, 2-37

INFOS® II 1-12, 5-1

Inquiry Program, File 1-1f, 1-4, 2-3ff, 2-21ff, 3-1, 3-4, 3-8f

INSERT key 1-5, 2-40, 2-51, 2-53, 2-82, 2-126, 3-9, 3-40

installation name 1-12

installing PROXI 1-12

Interactive COBOL (CS) 4-17

interactive session 1-2, 1-6ff, 2-40, 2-82, D-4f

ISAM 1-2, 2-7, 2-10, 2-21, 2-23, 2-37

K

key,
 alternate 1-2
 cursor control 1-5f, 2-51, 2-53, 3-17ff, D-2

DELETE 1-5, 2-40, 2-51, 2-53, 2-82, 2-126, 3-9, 3-40

editing 1-5, 2-50, 2-52, 3-17f, D-2

FINISHED 1-5, 2-5, 2-17, 2-29, 2-40, 2-51ff, 2-82, 2-126, 3-9, 3-25, D-1

function 1-4f, 2-7, 2-40, 2-82, 3-9, D-1

INSERT 1-5, 2-40, 2-51, 2-53, 2-82, 2-126, 3-9, 3-40

NEXT RECORD 1-5

SKIP IT 1-5, 2-126

key field 2-7f, 2-21, 2-37, 2-42f, 2-46f, 2-84f, 2-86f, 3-4, 3-16f, 4-10f, 4-23, 4-25

key range,
 ending 2-37, 2-79, 3-60f, 4-34f
 selection 2-43, 2-67, 2-84f, D-3
 starting 2-37, 2-79, 3-60f, 4-34f

KEY, RECORD 1-3f, 1-10, 2-7f, 2-21, 2-42f, 2-84f, 2-86f, 3-4f, 4-25

keyboard 1-4f, D-1

L

LABEL RECORDS clause 4-16f

layout,
 form 1-4, 2-75f, 2-82
 report 1-4, 2-33f, 2-48f

legend line 1-4, 2-33f, 2-40, 2-48ff
 repeated 2-34, 2-49

level,
 control break 1-4, 2-33, 2-36f, 2-40, 2-67, 2-75, 2-77, 2-82
 FD entry 4-18f
 Working Storage entry 4-24f
 line, conditionally printed 2-33, 2-35, 2-37, 2-40,

2-58ff, 2-75ff, 2-100ff, 2-108ff, 2-116ff
 detail 1-4, 2-33ff, 2-40ff, 2-58ff, 2-75f, 2-77f, 2-82, 2-96ff
 heading 1-4, 2-33ff, 2-40, 2-48f, 2-52ff
 legend 1-4, 2-33ff, 2-40, 2-48ff
 number per page 1-4, 2-48f, 2-82, 2-88f
 numbers, stripping 1-10, F-1f
 page break 1-4, 2-75f, 2-77, 2-82, 2-108ff
 top-of-form 2-75f, 2-77, 2-82, 2-90ff
 total 1-4, 2-33f, 2-36, 2-40, 2-67ff, 2-75f, 2-77f, 2-82, 2-115f
 lines per page 1-4, 2-48f, 2-82, 2-88f
 literal field,
 entering on screen 3-3f, 3-17
 nonnumeric 2-57, 2-61, 2-93, 2-95, 2-103, 2-105, 2-111, 2-119, 2-121, 3-25, 3-31
 logical test,
 for printing 2-35, 2-40, 2-59ff, 2-76, 2-79, 2-82, 2-93, 2-102, 2-111, 2-119
 for record selection 2-37, 2-40, 2-55ff, 2-76, 2-79, 2-82, 2-99
 for validation 3-8, 3-30f
 LOGON 2-4f

M

Main menu 1-2, 1-6, 1-13
 main program framework 1-1, 1-3, 1-10f, 1-13, 2-1, 2-3, 2-9f, 2-21f, 2-38, 2-79, 2-129, 5-1
 maintenance menu 1-4, 2-9, 2-12
 Maintenance program, File 1-1ff, 2-3ff, 2-7ff, 2-21, 3-1, 3-4, 3-8f, 3-17
 menu, Main 1-2, 1-6, 1-13
 menu item name 2-14f
 message 1-2, 1-7, 3-25, D-2
 error 1-7
 modifying
 a data-file copy file 4-2, 4-15
 a program 1-11, 2-2, 2-19, 2-40, 2-82, 2-125ff
 a record (operator) 1-3, 2-7, 2-9, D-4
 a screen format 3-1f, 3-9, 3-10f, 3-36ff, E-1f
 module,
 File Definitions 1-1f, 1-6, 1-8, 1-10, 1-13, 2-1, 2-3, 2-27, 2-38, 2-79, 4-1ff
 Program Generator 1-3, 1-8, 1-10f, 1-13, 2-1ff
 Screen Generator 1-8, 1-10, 1-13, 2-1, 2-9, 2-21, 2-27, 3-1ff
 2-27, 3-1ff

N

name,
 application 2-4f, 2-9
 characters used in 1-11f
 data-file record 2-9, 2-12, 4-19
 default 1-3, 2-3, 2-15, 2-27, 2-43, 2-85
 file, conventions 1-3, 1-11f
 installation 1-12
 menu item 2-14f

paragraph, COBOL 1-2, 1-10, 2-12, 2-24, 3-32f, E-2
 record 2-14f
 section, COBOL 1-2, 1-10, 2-12, 2-24, 3-32f, E-2
 NEXT RECORD key 1-5, D-1
 no change method (by operator) 3-25
 nonnumeric literal 2-93, 2-95, 2-99, 2-103, 2-105, 2-111, 2-113, 2-119, 2-121, 3-31
 numbered prompts 3-4, 3-7, 3-16f
 numbering, page 2-59, 2-67, 2-69, 2-75, 2-78, 2-101
 numeric field 1-8, 3-24f, 3-28f

O

OC-STATUS 2-12, 2-24, 3-33
 opening a reference file 1-2, 1-10, 2-10, 2-15, 2-23, 2-27
 ORGANIZATION clause 4-10f
 output format 2-18f, 2-30f, 2-72f, 2-122f, 2-128f, 3-50f, 3-54f, 4-8f, 4-14f, 4-22f, 4-28f
 output method,
 datafile copy files 4-32ff
 operator selected D-4
 reports and forms 2-48f, 2-88f
 screen copy files 3-58ff
 screen format 3-42ff
 overlaying screens 3-5f, 3-21
 Own Code 1-1, 2-1, 2-12, 2-14f, 2-23ff, 3-7, 3-40, 5-1
 for accumulators 2-63, 2-71, 2-78, 2-94, 2-113, 2-121
 for reference files 1-2, 1-10, 2-10, 2-23
 for special processing 1-2, 1-10, 1-12, 2-23ff, 3-24ff, 3-32f
 status flag 2-12, 2-24

P

page,
 bottom 2-88f
 break line 1-4, 2-76, 2-77, 2-82, 2-108ff
 counter 2-59, 2-67, 2-69, 2-101
 lines per 1-4, 2-48f, 2-82, 2-88f
 numbering 2-75, 2-78
 paragraph name 1-2, 1-10, 2-12, 2-24, 3-32f, E-2
 parameter file 1-2f, 1-10f, 2-1ff, 2-23
 program 1-2, 1-10ff, 2-1f, 2-14, 2-27, 2-125, 2-129
 screen format 1-11, 3-1f, 3-15ff, 3-37, 3-40, E-1f
 picture
 for a print field 2-62f, 2-70f, 2-78, 2-94f, 2-104f, 2-112f, 2-120f
 for a screen field 3-7, 3-24, 3-28f
 .PL file 1-12, 2-12, 2-14f, 2-24, 2-27, 3-33, 3-55, 3-59, E-2
 .PP file 1-12, 2-1, 2-14, 2-27, 2-125, 2-129
 principal data file 1-2, 1-4, 1-10, 2-7, 2-10, 2-21, 2-23, 2-33, 2-35ff, 2-37, 2-75, 2-79, 2-82, 2-87, 2-115
 describing 2-14, 2-26f, 2-42f, 2-84f
 print field 2-33ff, 2-76ff
 print format, default 2-63, 2-70, 2-94, 2-105, 2-113, 2-121

printing,
 conditions 2-33, 2-35, 2-37, 2-82, 2-100ff
 data-file copy file 4-1, 4-4ff, 4-32ff
 functions, screen 3-1f, 3-9
 runtime information 2-48f
 a screen copy file 3-1f, 3-9, 3-11, 3-58ff
 a screen definition 3-1f, 3-11, 3-42ff
 printing program, file 1-3, 2-7, 2-10, 2-15, D-3f
 naming 2-14f
 Printing Program, Form 1-1f, 1-4f, 2-3f, 2-15, 2-75ff,
 2-126
 Procedure Division 1-2, 1-12, 2-12, 2-15, 2-24, 2-27,
 3-11, 3-55
 program,
 binding 1-1, 5-1
 building 1-10, 2-2, 2-3ff, 2-9ff, 2-22ff, 2-38ff,
 2-79ff
 changing 1-11, 2-2, 2-19, 2-40, 2-82, 2-125ff
 definition 2-3ff, 2-9, 2-21, 2-125
 documentation 5-2
 File Inquiry 1-1f, 1-4, 2-3ff, 2-21ff, 3-1, 3-4, 3-8f
 File Maintenance 1-1ff, 2-3ff, 2-7ff, 2-21, 3-1, 3-4,
 3-8f, 3-17
 Form Printing 1-1f, 1-4f, 2-3f, 2-15, 2-75ff, 2-126
 Generator 1-3, 1-8, 1-10f, 1-13, 2-1ff
 main framework 1-1, 1-3, 1-10, 1-13, 2-1, 2-3, 2-9f,
 2-21f, 2-38, 2-79, 2-129, 5-1
 operator 1-2ff, 1-10, 2-7ff, 2-12, 2-15, 2-21, 2-24,
 3-25, D-1ff
 parameter file 2-1ff
 printing 2-7, 2-10, 2-14f
 Report Writer 1-1f, 1-4f, 2-3ff, 2-15, 2-33ff, 2-126
 sample C-1ff
 Program Generator 1-3, 1-8, 1-10f, 1-13, 2-1ff
 menu 1-6, 2-2, 2-31, 2-73, 2-123
 prompt,
 line, screen 3-7, 3-24f
 number 1-9, 3-4, 3-7, 3-16f
 responding to 1-7
 PROXI,
 Commercial Systems' (CS) 4-11
 file, editing 1-11
 installing 1-12
 modules 1-1f, 1-6
 session 1-2, 1-6ff, 2-40, 2-82
 PROXI\$91 1-12

Q

quotation marks,
 when illegal 3-3, 3-17
 with nonnumeric literal 2-57, 2-61, 2-93, 2-95,
 2-103, 2-105, 2-111, 2-119, 2-121, 3-25, 3-31

R

random access 4-10f, 5-1
 range selection 2-37, 2-40, 2-79, 2-82, D-3
 reading a reference file 1-2, 1-10, 2-10, 2-15, 2-23, 2-27

record,
 adding 1-3, 2-7, D-4
 changing 1-3, 2-7, D-4
 deleting 1-3, 2-7, D-4
 displaying 1-4, 2-7, D-4
 identifying field 1-4, 2-7f, 2-12, 2-21, 2-37, 3-4,
 3-16f, 4-10f, 4-23, 4-25
 key 2-7f, 2-12, 2-24, 2-37, 2-43, 2-45ff, 2-84f, 2-87,
 3-4f, 3-17
 modifying 1-3, 2-7, D-4
 name 2-14f
 NEXT, key 1-5
 selection criteria 2-33, 2-35, 2-37, 2-40, 2-54ff,
 2-75, 2-77, 2-82, 2-96f, 2-106f
 RECORD KEY clause 1-3, 1-10, 2-87, 3-4f, 4-25
 RECORDING MODE clause 4-16f, 5-1
 reference file 1-2, 2-10, 2-14ff, 2-21, 2-23, 2-28f, 2-33,
 2-37, 2-42ff, 2-75, 2-79, 2-82, 2-84ff, 2-125
 access field 2-44ff, 2-86f
 access method 1-2, 2-10, 2-21, 2-23
 naming 2-16f, 2-28f, 2-44f, 2-86f
 opening and reading 1-2, 2-43, 4-10f, 4-25
 relative organization 1-2, 2-43, 4-10f, 4-25
 Release Notice 1-12
 repeated legend lines 2-34, 2-49
 repetitive entries (See "serial entry".)
 replacement method (by operator) 3-25
 report,
 format 1-4, 2-33f, 2-40, 2-48ff
 input 2-37
 layout 2-33f, 2-48ff
 title 1-4, 2-33ff, 2-48f
 Report Writer Program 1-1f, 1-4f, 2-3ff, 2-15, 2-33ff,
 2-126
 required field attribute 3-7, 3-24f
 responding to a prompt 1-7
 runtime information, printing 2-48f

S

sample programs C-1ff
 screen,
 blanking 3-5f, 3-20f, 3-38
 copy file, printing 3-1f, 3-9, 3-11, 3-58ff
 copy file, rebuilding 3-40, E-1
 data-entry 1-1f, 1-7, 2-1, 2-7
 fields, defining 3-6ff, 3-22f, 3-24ff, 3-28f
 numbers 1-8
 overlaying 3-5f, 3-21
 printing functions 3-1f, 3-9
 split 3-5f, 3-21
 screen definition
 copy file 1-10, 3-1f
 printing 3-1f, 3-11
 screen format
 adding 1-10, 1-13, 2-9, 3-1, 3-11, 3-14ff
 changing 3-1f, 3-9, 3-11, 3-36ff
 copy file 1-10, 3-1f, 3-9
 creating 1-10, 1-13, 2-9, 3-1, 3-11, 3-14ff

- defining 1-10, 2-21, 3-1f, 3-14ff, 3-36ff
- identifying 2-12, 2-14, 2-27
- representation 1-8
- parameter file 1-11, 3-1f, 3-15ff, 3-37, 3-40, E-1f
- printing 3-1f, 3-11, 3-42ff
- Screen Generator
 - menu 1-6, 3-1f
 - module 1-8, 1-10, 1-13, 2-1, 2-9, 2-21, 2-27, 3-1ff
- Screen Procedure copy file 1-10, 1-12, 2-10, 2-27, 3-9, 3-40, 3-44f, 5-1, E-1f
- Screen Section copy file 1-10, 1-12, 2-10, 2-27, 3-9, 3-11, 3-40, 3-50f, 5-1, E-1f
- .SD file 1-12, 2-15, 2-27, 3-2, 3-51, 3-59
- section name 1-2, 1-10, 2-12, 2-24, 3-32f, E-2
- secure field attribute 3-7, 3-26
- SELECT statement 1-3, 1-10, 1-12f, 2-3, 2-10, 2-14, 2-27, 2-43, 4-1f, 4-5, 4-25, 5-1
 - building 4-8ff
- selection,
 - key range 2-43, 2-67, 2-84f, D-3
 - record 2-33, 2-35, 2-37, 2-40, 2-54ff, 2-75, 2-77, 2-82, 2-96f, 2-106f
- SELECTION-TYPE 3-17
- sequence number, entry 3-4f, 3-7, 3-22f
 - multiple field 3-23
- sequential access 4-10f
- sequential organization 1-2, 2-37, 4-10f
- serial entry 1-5, 2-126
- .SF file 1-12, 3-1f, 3-9, 3-15, 3-40, E-1f
- SKIP IT key 1-5, 2-126
- .SL file 1-3, 1-10, 1-12, 2-14ff, 2-27, 2-29, 2-43, 3-37, 4-1f, 4-4ff, 4-8f, 4-33
- slash character
 - for editing 2-50, 2-52, 3-18
 - for form feed 3-59, 4-18f, 4-24, 4-33
- sort key 2-40, 2-43, 2-46f
- sorting 2-37, 2-42f, 2-46f
- source code 1-1
- SPEED F-1f
- split screen effect 3-5f, 3-21
- spooling output 2-48f, 2-82, 2-88f
- start key, range 2-37, 2-79, 3-60f, 4-34f

- statement,
 - FD 1-3, 1-10, 1-12f, 2-3, 2-10, 2-14f, 2-27, 2-40, 4-1f, 4-4f, 4-14ff, 5-1
 - SELECT 1-3, 1-10, 1-12f, 2-3, 2-10, 2-14, 2-27, 2-43, 4-1f, 4-5, 4-8ff, 4-25, 5-1
- status flag, Own Code 2-12, 2-24
- stripping line numbers 1-10, F-1f
- supplementary code (See "Own Code".)

T

- template 1-4f, 2-7, 2-126
- test,
 - logical 2-35, 2-37, 2-40, 2-55, 2-57, 2-76, 2-79, 2-82, 2-92f, 2-97ff, 2-102f, 2-109ff, 3-8, 3-30f
 - validation 3-7f, 3-25, 3-30f, 3-40
- title of a report 1-4, 2-40
- top-of-form line 2-75f, 2-77, 2-82, 2-90ff
- top-of-page 2-49, 2-59, 2-67, 2-69, 2-100f, 3-59, 4-19, 4-24, 4-33
- total line 1-4, 2-33f, 2-36, 2-40, 2-67ff, 2-75f, 2-77f, 2-82, 2-115ff
- totaling, automatic 2-63, 2-71, 2-78, 2-105, 2-121

U

- unnumbered variable field 3-22f
- uppercase letters, using 1-9, 2-5

V

- validation test 3-7f, 3-25, 3-30f, 3-40
- validity check (See "validation test".)
- variable field 2-9, 2-21, 3-3f, 3-6, 3-16f, 3-23ff

W

- Working Storage entries 1-2, 1-10, 1-12, 2-10, 4-1f, 5-1
 - data-file copy file 1-12, 4-22f
 - own code copy file 1-12, 2-12, 2-14f, 2-24, 2-26f
- .WS file 1-10, 1-12, 2-12, 2-14f, 2-24, 2-27, 2-43, 4-2, 4-4ff, 4-22ff, 4-33



