

# Professional<sup>TM</sup> 300 series

DATATRIEVE  
For Beginners

SHREWSBURY LIBRARY  
DIGITAL EQUIPMENT CORPORATION  
SHR1-3/G18  
DTN 237-2256

DECLIT  
AA  
PRO  
U439A

digital

---

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The information and drawings, herein, are the property of Digital Equipment Corporation and shall not be reproduced or copied or used in whole or in part as the basis for the manufacture or sale of items without written permission.

Copyright © 1983 by Digital Equipment Corporation.  
All Rights Reserved.

The following are trademarks of Digital Equipment Corporation:

**digital**

CTI BUS  
DEC  
DECmate  
DECsystem-10  
DECSYSTEM-20  
DECUS  
DECwriter  
DIBOL

MASSBUS  
PDP  
P/OS  
Professional  
PROSE  
PRO/BASIC  
PRO/DATATRIEVE  
PRO/FMS  
PRO/RMS  
Tool Kit

Rainbow  
RSTS  
RSX  
UNIBUS  
VAX  
VMS  
VT  
Work Processor

94-0031049/31

#### WHO SHOULD READ THIS BOOK?

If you know little or nothing about using a computer to manage information, do not have a lot of time to spare, and want to learn to use PRO/DATATRIEVE quickly, then this book is for you.

This book is a self-paced guide that leads you through the process of using PRO/DATATRIEVE to manage information, create mailing lists, and write reports. As you read this book, you learn to use some of PRO/DATATRIEVE's most useful features and to feel comfortable "speaking" PRO/DATATRIEVE. In addition, this book introduces you to key information management concepts. You can then go on to the *PRO/DATATRIEVE Handbook* to learn more about PRO/DATATRIEVE and information management.



29170

## WHAT IS INSIDE?

The first two chapters introduce you to PRO/DATATRIEVE, provide an overview of information management terms and concepts, and show you how to get started using PRO/DATATRIEVE on your Professional 350.

The remaining five chapters take you through the process of using PRO/DATATRIEVE to manage customer information:

- In Chapter 3, you learn to organize information and create a *database*. A database is a collection of records maintained on a computer and organized for fast and accurate inquiry and update.
- Chapter 4 shows you how to find, display, and sort records stored in a database and Chapter 5 shows you how to create groups of related records.
- Chapter 6 shows you how to change and delete information to keep your database up-to-date.
- In Chapter 7, you learn to create meaningful reports from a database.

Finally, the last chapter in this book introduces you to PRO/DATATRIEVE features that save you typing time.

Throughout this book, you have many opportunities to try using PRO/DATATRIEVE. To get the full benefit of this book, read the first chapter and then sit down at your Professional 350 and work through the following chapters. In the examples, text in red indicates what you should type. PRO/DATATRIEVE's responses should match the black text in the examples.

# Contents

---

<b>CHAPTER 1</b>	<b>YOU, PRO/DATATRIEVE, AND THE PROFESSIONAL 350</b>	
<hr/>		
What Is Information Management? .....		1
What Is PRO/DATATRIEVE? .....		3
Speaking the Language .....		3
Fields, Records, and Files .....		3
Record Definitions, Domains, Files, and ADT .....		5
RSEs, Collections, and the Report Writer .....		6
Summary .....		7
<b>CHAPTER 2</b>	<b>GETTING STARTED WITH PRO/DATATRIEVE</b>	
<hr/>		
Installing PRO/DATATRIEVE .....		9
Getting Into and Out of PRO/DATATRIEVE .....		9
Using PRO/DATATRIEVE Commands and Statements .....		10
Showing Dictionary Information .....		11
Working with Domains .....		12
Using PRO/DATATRIEVE Dictionaries .....		14
Using PRO/DATATRIEVE Function Keys .....		16
DO, CANCEL, MAIN SCREEN, and HELP .....		16
EXIT and INTERRUPT/DO .....		18
Summary .....		18
<b>CHAPTER 3</b>	<b>SETTING UP THE DATA BASE</b>	
<hr/>		
Analyzing the Data .....		22
Identifying the Fields .....		23
Determining Field Characteristics .....		23
Naming the Domain and Data File .....		25

Defining Data with ADT .....	27
Getting Domains Ready .....	32
Storing Information .....	33
Storing One Record .....	33
Storing More Than One Record At a Time .....	35
Displaying Information .....	37
Finishing Domains .....	38
Summary .....	38

#### **CHAPTER 4           GETTING INFORMATION**

---

Forming Record Streams .....	40
Forming an RSE .....	41
Restricting the Number of Records in a Record Stream .....	43
Establishing Conditions for the Record Stream .....	44
Using Relational Operators .....	45
Using Boolean Operators .....	48
Sorting Records in a Record Stream .....	51
Summary .....	53

#### **CHAPTER 5           CREATING RECORD COLLECTIONS**

---

Forming and Naming Collections .....	55
Deleting Collections with the Release Command .....	59
Sorting Collections .....	60
Deleting Collections with the Finish Command .....	62
Summary .....	63

#### **CHAPTER 6           CHANGING AND DELETING INFORMATION**

---

Selecting Records from a Collection .....	66
Using Collections to Modify and Erase Records .....	68
Modifying the Selected Record .....	69
Erasing the Selected Record .....	71
Modifying All Records in a Collection .....	72
Erasing All Records in a Collection .....	72
Using Record Streams to Modify and Erase Records .....	73
Summary .....	75

---

**CHAPTER 7                    CREATING REPORTS**

---

Forming a Simple Report Specification .....	78
Titling a Report .....	80
Writing Reports to a File .....	81
Setting Report Width .....	82
Formatting Report Lines .....	83
Formatting Detail Lines .....	84
Formatting Summary Lines .....	87
Summary .....	89

---

**CHAPTER 8                    SAVING TYPING TIME**

---

Using Guide Mode .....	91
Creating and Using Procedures .....	92
Editing Items in Your Dictionary .....	97
Summary .....	99
Conclusion .....	99

---

**GLOSSARY AND INDEX**

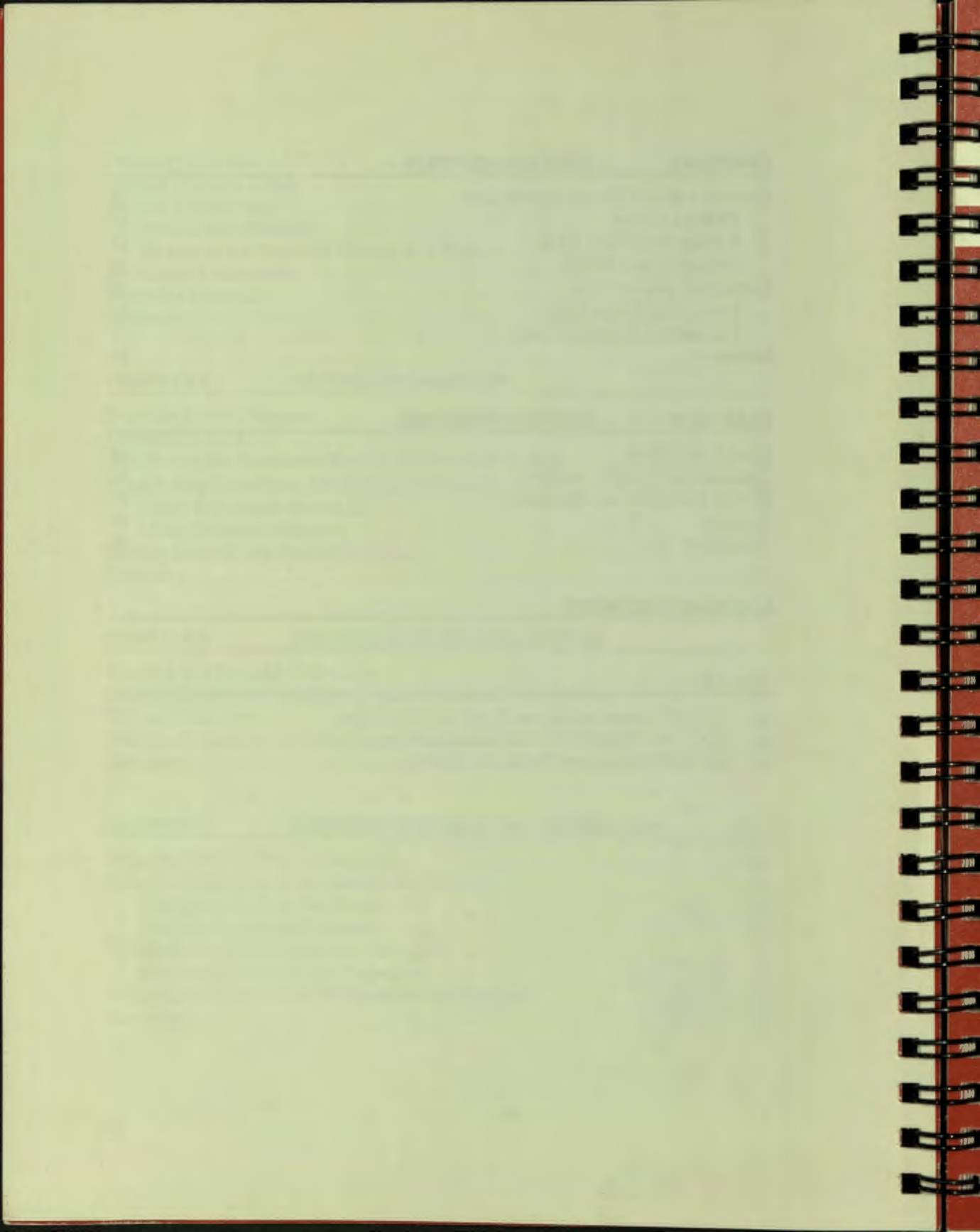
---

---

**TABLES**

---

3-1    READY Command Access Types and Privileges .....	32
4-1    PRO/DATATRIEVE Relational Operators .....	46
6-1    SELECT Statement Forms and Effects .....	66

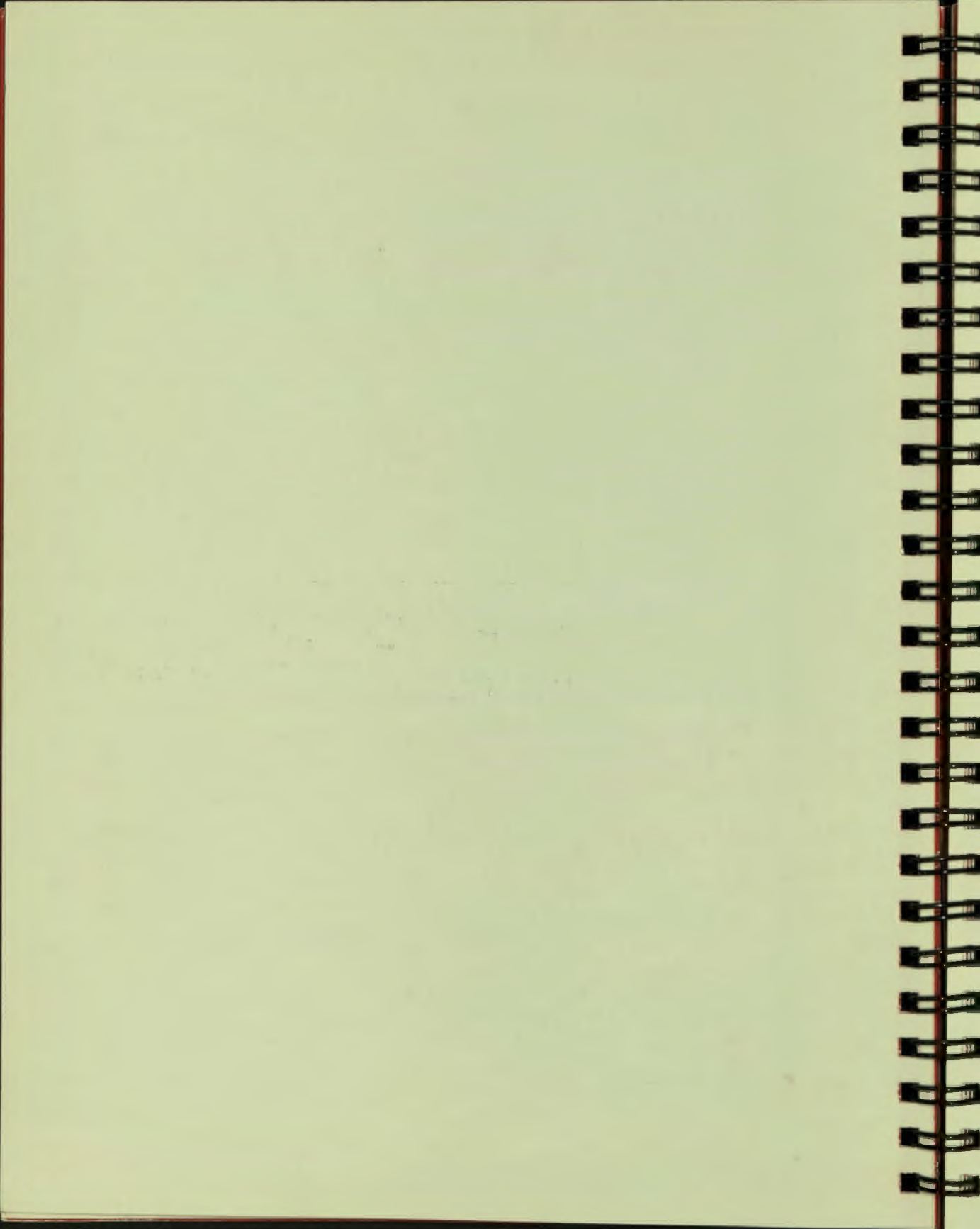




# 1

---

You, PRO/DATATRIEVE,  
and the Professional 350



## Chapter 1

---

# You, PRO/DATATRIEVE, and the Professional 350

### WHAT IS INFORMATION MANAGEMENT?

Whatever your job, you have one thing in common with all other professionals — you have to manage information. If you use your Professional 350 filing system to store memos, create reports, and maintain accounting data or customer records, you are using the Professional 350 to manage information.

Although some of the terminology may be new to you, the process of *information management* is probably familiar. If you have an address book, for instance, you go through the process of information management when you:

- Pick the style of address book that best meets your needs
- Write the original names and addresses in the book
- Look up the address of a restaurant in the book
- Add your new dentist's name and address
- Delete your previous dentist's name and address
- Change a friend's address when the family moves
- Create a mailing list



In more general terms, information management consists of the following tasks:

- Setting up a file to hold related information
- Storing original information in the file
- Getting information from the file
- Adding new information to the file
- Deleting old information from the file
- Making changes to information stored in the file
- Organizing information in the file into meaningful reports

Information management tools vary greatly in their efficiency and accuracy. Your address book, for example, can become messy and disorganized if you keep making changes and additions to it. Or, if you accidentally drop a stack of alphabetically arranged address cards, you could waste valuable time putting the cards back in order.

If you want to create a mailing list from cards or an address book, you have to rewrite information on a sheet of paper or on address labels. You not only lose time, but could lose or forget an important address. If you keep budget information on ledger sheets in a file folder, you have to recalculate the entire budget every time you change a budget figure. As with the mailing list, you increase the risk of making a mistake, while also losing time.



## WHAT IS PRO/DATATRIEVE?

PRO/DATATRIEVE is an interactive language and report writing tool that can help you keep information organized and up-to-date efficiently, quickly, and accurately. You tell PRO/DATATRIEVE what you want to do, and PRO/DATATRIEVE does it. If you do not give PRO/DATATRIEVE enough information to do what you want, PRO/DATATRIEVE responds with messages to help you along.

You communicate with PRO/DATATRIEVE by typing commands and statements on your keyboard. *Commands* and *statements* are familiar English words that you combine in sentence-like form so that you can “talk” to PRO/DATATRIEVE. You tell PRO/DATATRIEVE to update information, store information, or create a report much as you would ask someone to do these jobs for you. PRO/DATATRIEVE, though, can do these jobs faster, more efficiently, and more accurately than you or any other human being can.

PRO/DATATRIEVE stores information in *files* on your Professional 350 disk or diskette. The information you need — John Smith’s phone number, addresses for all customers living in Boston, or a complete mailing list — is never more than a few keystrokes away. Furthermore, you can keep information up-to-date and accurate without ever leaving your chair or your Professional to search for missing files or lost index cards.

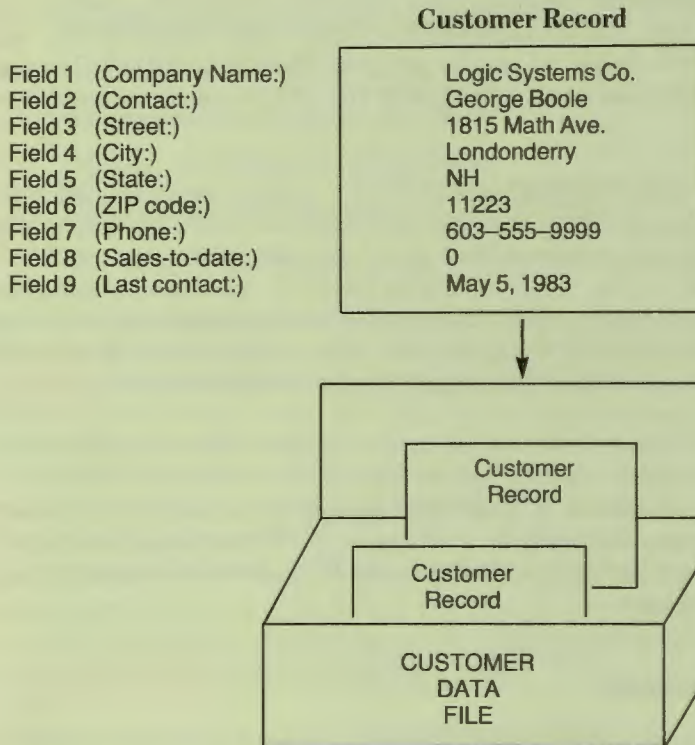
## SPEAKING THE LANGUAGE

Before you start learning PRO/DATATRIEVE, you need to become acquainted with some PRO/DATATRIEVE terms and concepts. New terms are italicized and then briefly explained. For your convenience, a glossary at the end of this book lists and describes PRO/DATATRIEVE terms and concepts, as well as the commands and statements introduced in this book.

### Fields, Records, and Files

PRO/DATATRIEVE organizes data into a system of fields, records, and files. Simply speaking, *data* is unprocessed information. It can range from the smallest meaningful piece of information, such as a character or number, to a huge *database* that contains hundreds of records.

*Fields* contain one or more pieces of data, *records* contain fields, and *files* contain records, like this:



PRO/DATATRIEVE stores pieces of data, like a state or ZIP code, in a field. A *field* can consist of a very simple piece of data like a two-character state abbreviation or a combination of data items like a street number and name.

A single field of data, though, does not usually have much meaning by itself, so PRO/DATATRIEVE stores fields in a record. A *record* consists of pieces of related data that you consider a unit of information. You create a customer record, for example, when you decide what customer data goes together to make up a unit of customer information.

Finally, PRO/DATATRIEVE stores records that contain similar pieces of information in a *file*. You decide what information a file contains when you decide what pieces of data a record contains. You might decide, for example, to keep inventory information in one file, customer orders in another file, and customer contact information in another file. The choices are entirely yours.

This field-record-file filing system corresponds to the familiar physical filing system you probably now use. Each data item on an index card, for example, corresponds to a field, the card corresponds to a record, and the index card box in which you store the cards corresponds to the file.

When you use PRO/DATATRIEVE, you select the files, records, and fields you need. If you want mailing list information about a customer, you tell PRO/DATATRIEVE to get data from the company name, street, city, state, and ZIP code fields and to ignore all other fields in the customer record. If you want to know the contact person and phone number for a customer, you tell PRO/DATATRIEVE to look at customer records another way.

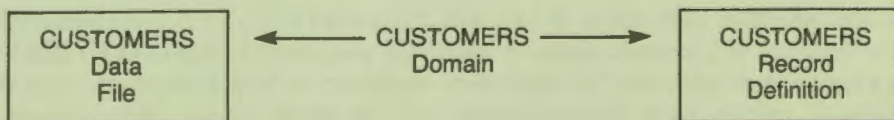
### Record Definitions, Domains, Files, and ADT

Record definitions and domains tell PRO/DATATRIEVE how to organize and interpret information stored in a data file. A *record definition* lists and names all the fields in a record and describes the type of data (number, letters, money, a percentage, and so on) that each field contains. A record definition tells PRO/DATATRIEVE what pieces of data a record can contain. PRO/DATATRIEVE stores record definitions separately from the records in a data file that contain your information.

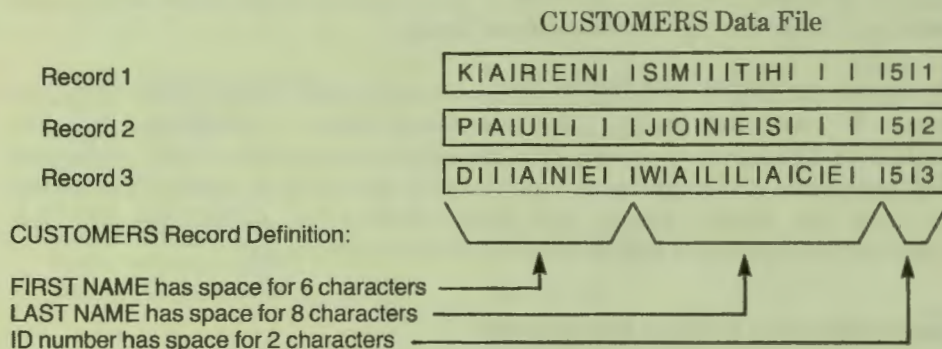
A *domain* associates a data file on your disk or diskette with a record definition. Think of a domain as an efficient office assistant who knows exactly how to fill out a particular form and where to file it when it's complete. Just as this efficient assistant knows what record goes in which file cabinet, a PRO/DATATRIEVE domain "knows" which record definition goes with which data file. Also, just as you wouldn't confuse your assistant with a filing cabinet, remember not to confuse a PRO/DATATRIEVE domain with a computer data file.

When you use a domain name, you tell PRO/DATATRIEVE to use a particular record definition to interpret information stored in a specific file. The domain itself does not contain data.

In order to work with a file that contains customer information, you need a domain that links a record definition with the file, like this:



This diagram represents three records in the CUSTOMERS data file and the associated CUSTOMERS record definition:



The CUSTOMERS domain tells PRO/DATATRIEVE which record definition, or map, to use to store and interpret the pieces of information stored in the CUSTOMERS file. You will learn much more about domains, records, and files as you read through this book. For now, just remember that domains link the data file with the record definition and do not themselves contain data.

In Chapter 4, you learn to use the PRO/DATATRIEVE *Application Design Tool* (ADT) to define domains, simple records, and data files. ADT helps you by prompting for all of the information needed. The *PRO/DATATRIEVE Handbook* shows how to define more complex records.

### RSEs, Collections, and the Report Writer

So much for the structure of information in PRO/DATATRIEVE. The heart of information management is the ability to get the information you need in a meaningful format. To create a customer mailing list from index cards, for example, you probably look at each card to find the group of customers you want to include on a mailing list. You may then sort the cards and type address information for each customer on a sheet of paper or on mailing labels. You can do the same thing in PRO/DATATRIEVE by using record selection expressions (RSEs), collections, and the PRO/DATATRIEVE Report Writer.

A *record selection expression* (RSE) tells PRO/DATATRIEVE the characteristics of the records you want to use. With RSEs, you can tell PRO/DATATRIEVE you want to work with only the records of customers in New Hampshire, only the records of customers in Massachusetts with no sales, or any other group of records you choose.

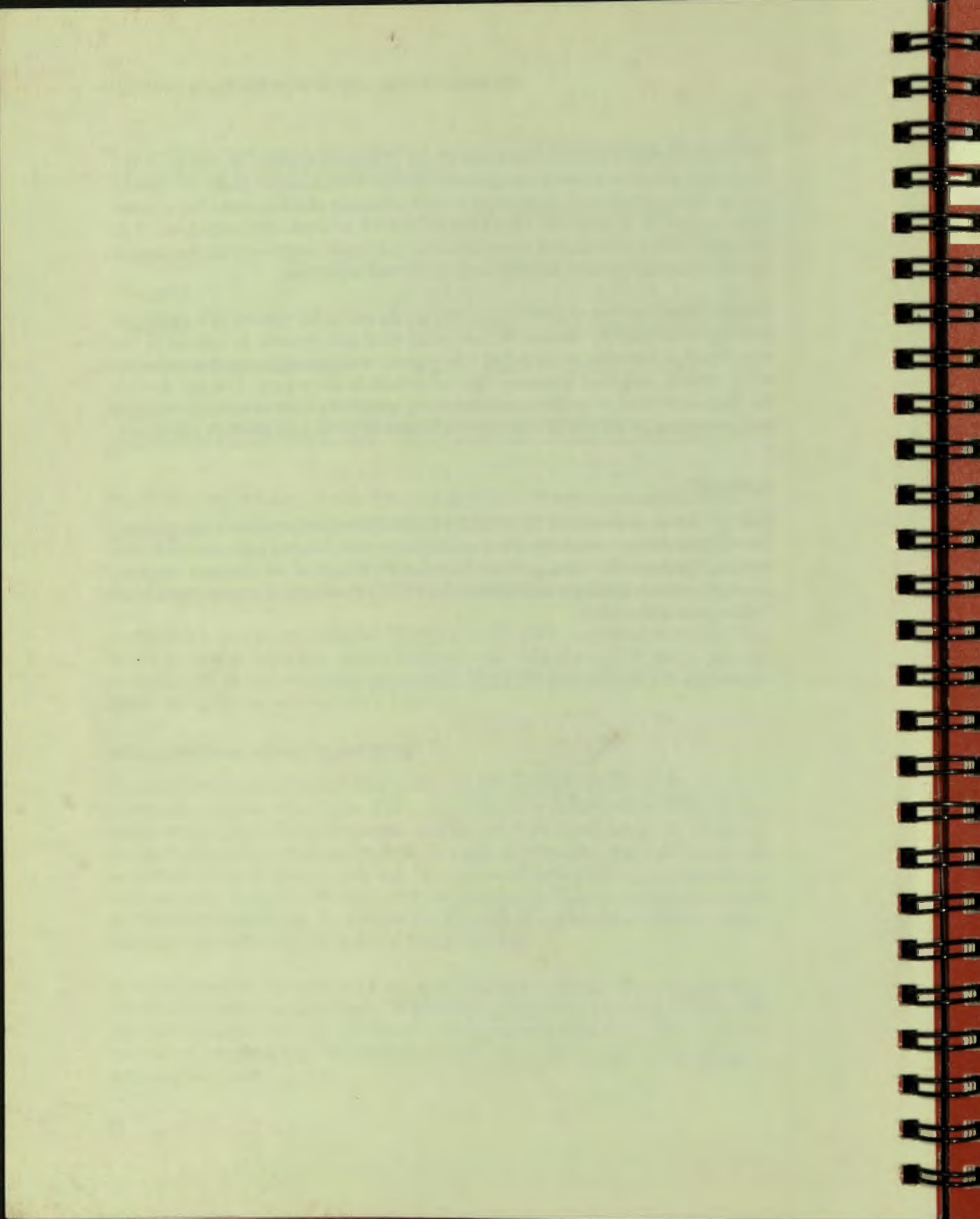


You can also use RSEs to create collections of related records. A *collection* is a temporary group of records you gather together from a larger group of records. Just as it is easier for you to organize a small collection of index cards into a meaningful report, it is easier for PRO/DATATRIEVE to work with collections. You can work with a collection of records just as you would work with all the records in a file. Chapter 5 describes how to use RSEs and collections.

Finally, with the *Report Writer*, you can create and print reports in a variety of formats. You tell the Report Writer what data and records to include in the report and, if you want, what to title the report, what headings to print at the top of the report, and what summary lines to include in the report. You can also tell the Report Writer to perform mathematical operations such as finding averages and computing totals and can include the results of these operations in the report.

### **SUMMARY**

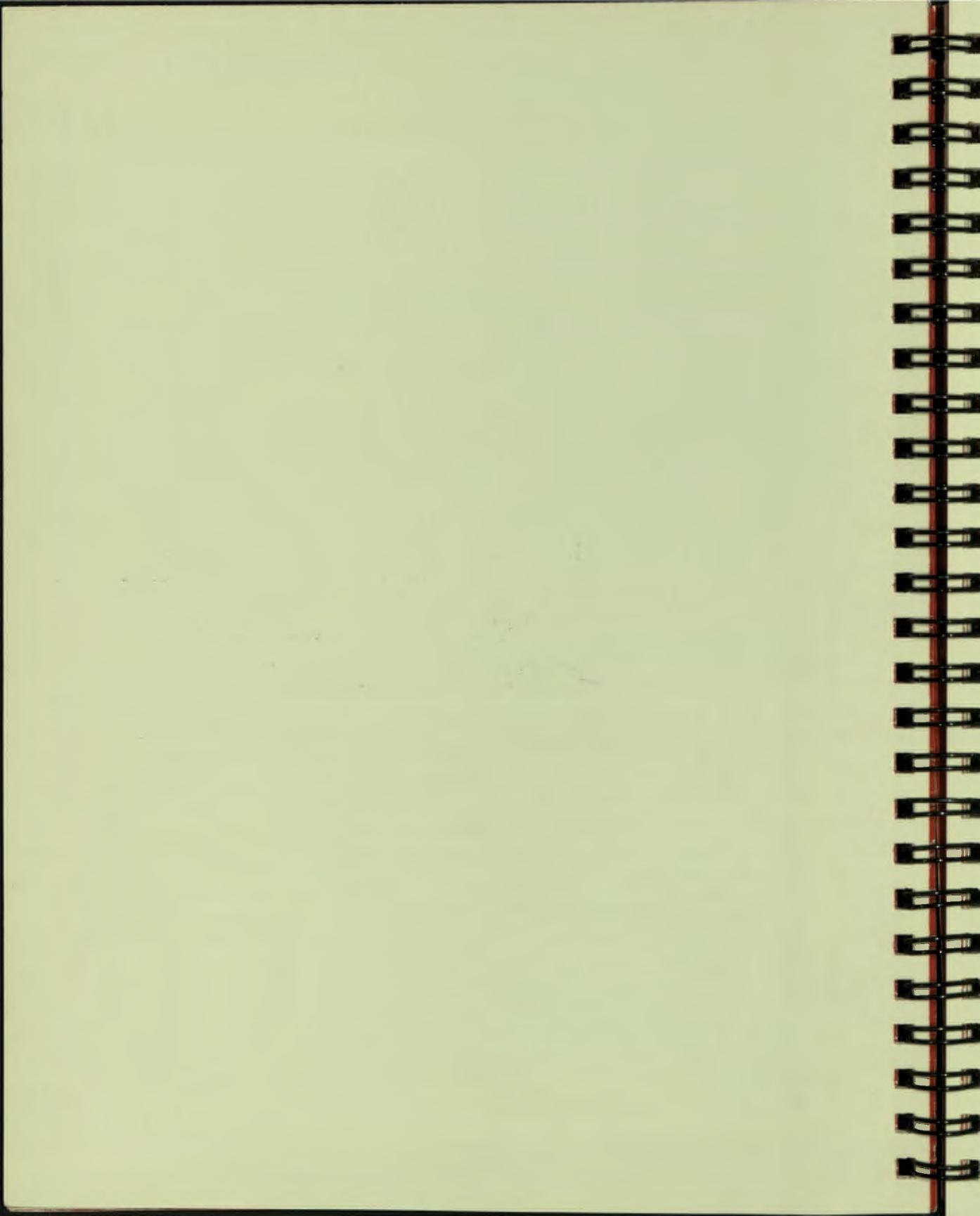
That's it for an overview of PRO/DATATRIEVE and information management. You will learn more about the terms and concepts described in this chapter as you continue through this book and use PRO/DATATRIEVE. In the next chapter, you start communicating with PRO/DATATRIEVE, so pull your chair up to your Professional and read on.



# 2

---

Getting Started with  
PRO/DATATRIEVE



## Chapter 2

---

# Getting Started With PRO/DATATRIEVE

### INSTALLING PRO/DATATRIEVE

Before you can start using PRO/DATATRIEVE, you must install it on your Professional 350, if it has not been installed by someone else. Check your Main Menu and Additional Application group menus to see whether PRO/DATATRIEVE has been installed. If not, follow the instructions on the *PRO/DATATRIEVE Installation Card*, contained in the Media and Documentation box, to install PRO/DATATRIEVE.

Although this book refers to PRO/DATATRIEVE installed on the Main Menu, you may give PRO/DATATRIEVE any name you choose and you may install it on an Additional Applications group menu instead of on the Main Menu.

### GETTING INTO AND OUT OF PRO/DATATRIEVE

To use PRO/DATATRIEVE, select the PRO/DATATRIEVE application from the Main Menu and press DO. If you gave PRO/DATATRIEVE a different name or installed it on an Additional Applications group menu, use your choice of name and menu instead.

PRO/DATATRIEVE displays a startup heading and a prompt to indicate that you must make the next move:

```
PRO/Datatrieve, DEC Query and Report System
Version: 1.0, 05-SEP-83
Copyright © 1983 Digital Equipment Corporation
Type HELP for help
DTR>
```

The DTR> prompt indicates that PRO/DATATRIEVE is waiting for you to tell it what to do. A *prompt* is a character or phrase that PRO/DATATRIEVE displays on your screen to let you know that it is waiting to do your bidding. When you see the DTR> prompt, you can type in commands and statements that tell PRO/DATATRIEVE what to do next.

Other types of prompts let you know what PRO/DATATRIEVE expects from you:

- The CON> prompt lets you know that PRO/DATATRIEVE is waiting for you to continue a command or statement.
- The RW> prompt tells you that the Report Writer expects you to type in a Report Writer statement.

You learn about other types of PRO/DATATRIEVE prompts as you work through this book.

When you finish working with PRO/DATATRIEVE and want to return to the Main Menu, press the EXIT key or type EXIT and press RETURN. Do not exit from PRO/DATATRIEVE now, though, unless you want to quit and return to this book later.

## USING PRO/DATATRIEVE COMMANDS AND STATEMENTS

A *command* consists of one or more keywords that let you define data structures and control PRO/DATATRIEVE. A *statement* also consists of one or more keywords and lets you manipulate data stored in data files. *Keywords* are the words that make up the PRO/DATATRIEVE language. Because keywords have special meaning to PRO/DATATRIEVE, you cannot use them as names for domains, records, or other data structures. In this book, keywords, as well as words you supply, are printed in uppercase letters.

This section shows how you can use commands and statements to get information about PRO/DATATRIEVE and from the sample domain, EMPLOYEES, that comes with PRO/DATATRIEVE. It also shows you how to store a record in the EMPLOYEES domain. Type the commands and statements as indicated in the **TRY IT** sections and observe what happens. All statements and commands introduced here are explained in more detail in later chapters in this book.

### Showing Dictionary Information

The SHOW command tells PRO/DATATRIEVE to display information about your PRO/DATATRIEVE environment. The PRO/DATATRIEVE *environment* consists of domains, records, and other dictionary objects available to you.

### TRY IT

To get a feel for PRO/DATATRIEVE, type SHOW and press DO in response to the DTR> prompt. You can press DO or RETURN whenever you are finished typing a line. This book uses DO instead of RETURN, but you can use whichever key is more convenient for you.

After you type SHOW and press DO, PRO/DATATRIEVE responds with a prompt that indicates you need to specify what you want to see. Type the word ALL and press DO to see the following display:

```
DTR> SHOW
[Looking for element to SHOW]
CON> ALL
Domains:
      EMPLOYEES
Records:
      EMPLOYEES_REC
Procedures:
Tables:
The current dictionary is DW1:[ZZAP00014]PRODTR.DIC;1
No established collections
No ready domains
```

This display shows you the names of dictionary objects (domains, records, procedures, and tables), the name of your current PRO/DATATRIEVE dictionary (DW1:[ZZAP00014]PRODTR.DIC;1), and other information about your PRO/DATATRIEVE environment. The system directory, [ZZAP00014], may be different than in this example, depending on when you installed PRO/DATATRIEVE on your Professional. The two messages at the end of the display provide information about records and domains currently in use. You will learn to interpret and use these messages as you go through this book.

A PRO/DATATRIEVE *dictionary* contains information about your data and its organization. PRO/DATATRIEVE automatically connects you to a dictionary file every time you choose PRO/DATATRIEVE from the Main Menu. In the next section, you learn to create dictionaries to help keep information orderly.

*Dictionary objects* are definitions and command sequences stored in a dictionary file that tell PRO/DATATRIEVE how to interpret data and what to do with data. As you can see from the display, dictionary objects include domains, records, tables, and procedures. Chapter 8 explains procedures and the *PRO/DATATRIEVE Handbook* discusses dictionary tables.

Other forms of the SHOW command let you display specific information about a particular category. For example, to show only the domains defined in your dictionary, you can type SHOW DOMAINS; to show only the records, you can type SHOW RECORDS.

### **TRY IT**

Type SHOW DOMAINS and press DO to see what domains are defined in your current dictionary. Domains, as explained in Chapter 1, link record definitions with data files. PRO/DATATRIEVE comes with a sample domain called EMPLOYEES. To see what the EMPLOYEES domain definition looks like, type SHOW EMPLOYEES and press DO.

The domain definition specifies the name of the domain (EMPLOYEES), the name of the record definition to use to interpret data (EMPLOYEES\_REC), and the name of the data file to use (EMPLOYEES.DAT).

Note that the files that actually contain data are not listed as dictionary objects. You can get information from a data file only by using the domain name associated with the data file. To find out the name of the data file associated with a particular domain, type SHOW and then the domain name.

If you want to see what a PRO/DATATRIEVE record definition looks like, type SHOW EMPLOYEES\_REC and press DO.

### **Working with Domains**

Before you can work with a domain, you have to tell PRO/DATATRIEVE to get the domain ready. You do this with the READY command. Because you cannot do anything with a domain that is not readied, READY is one of the first commands you use after you select PRO/DATATRIEVE from the Main Menu.

Once you have readied a domain, you can display information from it by using the PRINT statement. To display all records in the domain, you simply type PRINT and the domain name. To display only certain records, you type PRINT and a phrase that tells PRO/DATATRIEVE which records you want to see.



To store a record in a domain, you need to ready the domain for WRITE access. Then you type STORE, followed by the domain name and PRO/DATATRIEVE prompts you for data to store in the record.

### **TRY IT**

Use the READY command to get the EMPLOYEES domain ready:

```
DTR> READY EMPLOYEES
DTR>
```

The DTR> prompt tells you that the EMPLOYEES domain is now ready and that PRO/DATATRIEVE is waiting for you to tell it what to do next.

Type PRINT EMPLOYEES to see all the EMPLOYEES records, then type in the following PRINT statements to see different groups of employee records:

- PRINT EMPLOYEES WITH DEPT EQUAL "MATHEMATICS"
- PRINT EMPLOYEES WITH LAST-NAME EQUAL "BOOLE"
- PRINT EMPLOYEES WITH SALARY LESS-THAN 30000
- PRINT EMPLOYEES WITH STATUS EQUAL "TRAINEE"
- PRINT FIRST 2 EMPLOYEES WITH SALARY GREATER-THAN 30000

You can even have PRO/DATATRIEVE sort records before displaying them by typing PRINT statements like these:

- PRINT EMPLOYEES SORTED BY LAST-NAME
- PRINT EMPLOYEES WITH DEPT EQUAL "PHILOSOPHY" SORTED BY LAST-NAME
- PRINT EMPLOYEES SORTED BY LAST-NAME, FIRST-NAME

To store a new record in the EMPLOYEES domain, ready the domain again, only this time include the keyword WRITE after the domain name so that PRO/DATATRIEVE will let you write a new record to the data file. Then type STORE EMPLOYEES and type in the information PRO/DATATRIEVE asks you for, using yourself as the employee:

```
DTR> READY EMPLOYEES WRITE
DTR> STORE EMPLOYEES
Enter ID: 33333
Enter FIRST_NAME: YOUR NAME
Enter LAST_NAME: YOUR NAME
Enter DEPT: YOUR DEPT
Enter SALARY: 20000
DTR>
```

Note that you can store only ten characters in the FIRST-NAME and LAST-NAME fields. If your first or last name has more than 10 characters, type in only the first 10 characters, or PRO/DATATRIEVE will display an error message and prompt again for the name.

To see that PRO/DATATRIEVE has stored the new record, type a PRINT statement like this, putting your last name in the quotation marks

```
DTR> PRINT EMPLOYEES WITH LAST-NAME EQUAL "YOUR NAME"
```

ID	STATUS	FIRST NAME	LAST NAME	DEPT	SALARY
33333	TRAINEE	YOUR NAME	YOUR NAME	YOUR DEPT	\$20,000

These examples are intended to serve as a simple introduction to the power of PRO/DATATRIEVE. You can do many more things with the PRINT statement and with data in records:

- Chapter 3 provides more information on using the READY command and the PRINT statement, and explains how to store more than one record at a time.
- Chapter 4 explains how you identify the records with which you want to work.
- Chapter 6 explains how to modify and delete records.

### USING PRO/DATATRIEVE DICTIONARIES

To see the name of your current dictionary, type SHOW DICTIONARY and press DO:

```
DTR> SHOW DICTIONARY
The current dictionary is DW1:[ZZAP00014]PRODTR.DIC;1
```

PRODTR.DIC is the name of the dictionary supplied with PRO/DATATRIEVE. You can put all your record and domain definitions in this dictionary, or you can create your own dictionaries to store information by category.

Think about all the types of information you might want to manage with PRO/DATATRIEVE. You might want to create one dictionary for your payroll and employee definitions, another dictionary for customer and sales definitions, and another for supplier and purchasing definitions.

Just as you organize your directories by category, it is a good idea to organize your dictionaries the same way. If you store too many unrelated definitions in a dictionary, you may have trouble remembering which definitions you need to use to perform a specific task.

Before you create a dictionary, decide which directory you want to contain the dictionary and data files. If you want to store sales information in a SALES directory, for example, set your current directory to SALES, select PRO/DATATRIEVE from the Main Menu, and you are ready to define a dictionary for sales information.

You tell PRO/DATATRIEVE to define a dictionary with the DEFINE DICTIONARY command. The only thing you need to add to the command is the name of the dictionary. As with all other files on your Professional 350, a dictionary file name can contain from one to nine letters and/or numbers (A-Z, 0-9). Since this book shows you how to create a customer information application, create a dictionary called CUSTOMERS in your current directory.

### **TRY IT**

Type DEFINE DICTIONARY CUSTOMERS and press DO. After you press DO, PRO/DATATRIEVE creates a file named CUSTOMERS.DIC in your current directory and establishes the new dictionary as your current dictionary. The .DIC type assigned to the file identifies the file as a PRO/DATATRIEVE dictionary.

To see that your current dictionary is now CUSTOMERS.DIC, type the SHOW DICTIONARY command and press DO:

```
DTR> SHOW DICTIONARY
The current dictionary is DW1:[USERFILES]CUSTOMERS.DIC;1
```

The DEFINE DICTIONARY command automatically sets your current dictionary to the new dictionary. If you exit from PRO/DATATRIEVE and then choose it again from the Main Menu, your current dictionary is PRODTR.DIC. To set CUSTOMERS.DIC as your current dictionary, type SET DICTIONARY CUSTOMERS.DIC. To set the current dictionary back to PRODTR.DIC, type

SET DICTIONARY and press DO. When you do not specify a dictionary, PRO/DATATRIEVE assumes you want to return to PRODTR.DIC.

Type SET DICTIONARY to return to PRODTR.DIC, use the SHOW DICTIONARY command, then set the current dictionary to CUSTOMERS.DIC:

```
DTR> SET DICTIONARY
DTR> SHOW DICTIONARY
The current dictionary is DW1:[ZZAP00014]PRODTR.DIC;1
DTR> SET DICTIONARY CUSTOMERS
DTR> SHOW DICTIONARY
The current dictionary is DW1:[USERFILES]CUSTOMERS.DIC;1
```

If you type the SHOW ALL command, you see that your new dictionary is a clean slate, ready to store your domain and record definitions. After you read the next section to learn how your keyboard function keys operate in PRO/DATATRIEVE, you can begin storing definitions in CUSTOMERS.DIC.

### USING PRO/DATATRIEVE FUNCTION KEYS

You can use the MAIN SCREEN, DO, CANCEL, HELP, and EXIT function keys and the INTERRUPT/DO key sequence in PRO/DATATRIEVE much as you can when using Professional menus. Other function keys on your Professional keyboard beep when pressed, indicating that they do not perform a specific function in PRO/DATATRIEVE.

#### DO, CANCEL, MAIN SCREEN, and HELP

To execute a PRO/DATATRIEVE command or statement or to end a line typed in response to a PRO/DATATRIEVE prompt, press the DO key. If you prefer, you can use the RETURN key instead of the DO key to execute commands and end lines.

To cancel a command or statement before you have pressed DO, press the CANCEL key. PRO/DATATRIEVE ignores the line you just typed and prompts again for input. If you press DO after typing an incorrect line, PRO/DATATRIEVE displays an error message.

To exit from PRO/DATATRIEVE and return to the Main Menu, you can type EXIT and press DO or you can simply press the MAIN SCREEN key. If PRO/DATATRIEVE is performing a calculation or executing a command or statement when you press MAIN SCREEN, it finishes its current operation and then returns you to the Main Menu, with one exception: if you are in Guide Mode

(explained in Chapter 8), the MAIN SCREEN key returns you to the DTR> prompt.

To find out about using PRO/DATATRIEVE Help, press the HELP key. PRO/DATATRIEVE displays information on the type of help available. To get information on a particular topic:

- Type HELP HELP and press the DO key to display a list of topics.
- Type HELP and a topic, then press DO.

### TRY IT

Type in a sequence of nonsense characters and press DO. Then type in the same sequence of characters and press the CANCEL key:

```
DTR> JAJFLGHJGUJ(DO)
Expected statement, encountered "JAJFLGHJGUJ"
DTR> JAJFLGHJGUJ(CANCEL)
DTR>
```

Press the HELP key to display information on using PRO/DATATRIEVE Help, read the information, then display a list of available topics by typing HELP HELP. Select a topic, like DICTIONARY, from the list and type HELP DICTIONARY and press DO to see the following text:

```
DTR> HELP DICTIONARY
The Datatrieve dictionary is used to hold the definitions of
records, domains, description tables, and procedures. When
Datatrieve is initiated, it is linked to a central data
dictionary. To switch to a private dictionary, type the
command:
```

```
SET DICTIONARY file-spec
```

where "file-spec" is the file name of a private Datatrieve dictionary.

To display the file name of your current dictionary, type the command:

```
SHOW DICTIONARY
```

```
To build a private dictionary, use the DEFINE DICTIONARY
command. For additional information see HELP ADVANCED
DICTIONARY.
DTR>
```

**EXIT and INTERRUPT/DO**

The result of pressing the EXIT key depends on what PRO/DATATRIEVE is doing:

- Pressing the EXIT key in response to the DTR> prompt returns you to the Main Menu. In this instance, pressing EXIT is the same as pressing MAIN SCREEN or typing EXIT and pressing DO.
- Pressing the EXIT key in response to any PRO/DATATRIEVE prompt except DTR> returns you to the DTR> prompt.
- Pressing the EXIT key while PRO/DATATRIEVE is calculating or executing a command returns you to the Main Menu *after* PRO/DATATRIEVE finishes its current operation.

Pressing the CTRL key and the Z key at the same time (CTRL/Z) has the same effect as pressing the EXIT key.

The result of pressing INTERRUPT and then DO (INTERRUPT/DO) also depends on what PRO/DATATRIEVE is doing:

- Pressing INTERRUPT and then DO in response to *any* PRO/DATATRIEVE prompt returns you to the Main Menu.
- Pressing INTERRUPT and then DO while PRO/DATATRIEVE is calculating or executing a command returns you to PRO/DATATRIEVE command level and the DTR> prompt.

Pressing the CTRL key and the C key at the same time (CTRL/C) has the same effect as pressing INTERRUPT and then DO.

**SUMMARY**

In this chapter you learned to:

- Enter and exit from PRO/DATATRIEVE
- Use the SHOW, DEFINE DICTIONARY, and SET DICTIONARY commands
- Get help on PRO/DATATRIEVE topics
- Use PRO/DATATRIEVE function keys

You also saw what a domain and record definition look like and used the **READY** command and the **PRINT** and **STORE** statements to work with data in the **EMPLOYEES** domain.

When you feel comfortable with the information in this chapter, go on to Chapter 3 to learn how to analyze data and create a database.

### PRO/DATATRIEVE Function Keys

DO	CANCEL	HELP
Press to execute or end the current line	Press to cancel all characters typed on the current line	Press for general information on using PRO/DATATRIEVE on-line help
	In response to the DTR> or Guide prompt:	While calculating or executing commands:
EXIT	Press to return to the Main Menu from DTR> and to DTR> from Guide Mode	In response to any other prompt:
INTERRUPT	Press to return to the Main Menu	Press to finish the current operation and return to the Main Menu
DO	Press to return to the Main Menu	Press to return to the DTR> prompt without completing the current operation
MAIN SCREEN	Press to return to the Main Menu from DTR> and to DTR> from Guide Mode	Press to finish the current operation and return to the Main Menu

Faint, illegible text at the top of the page, possibly a header or introductory paragraph.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

Section of text, possibly a sub-header or a specific heading.

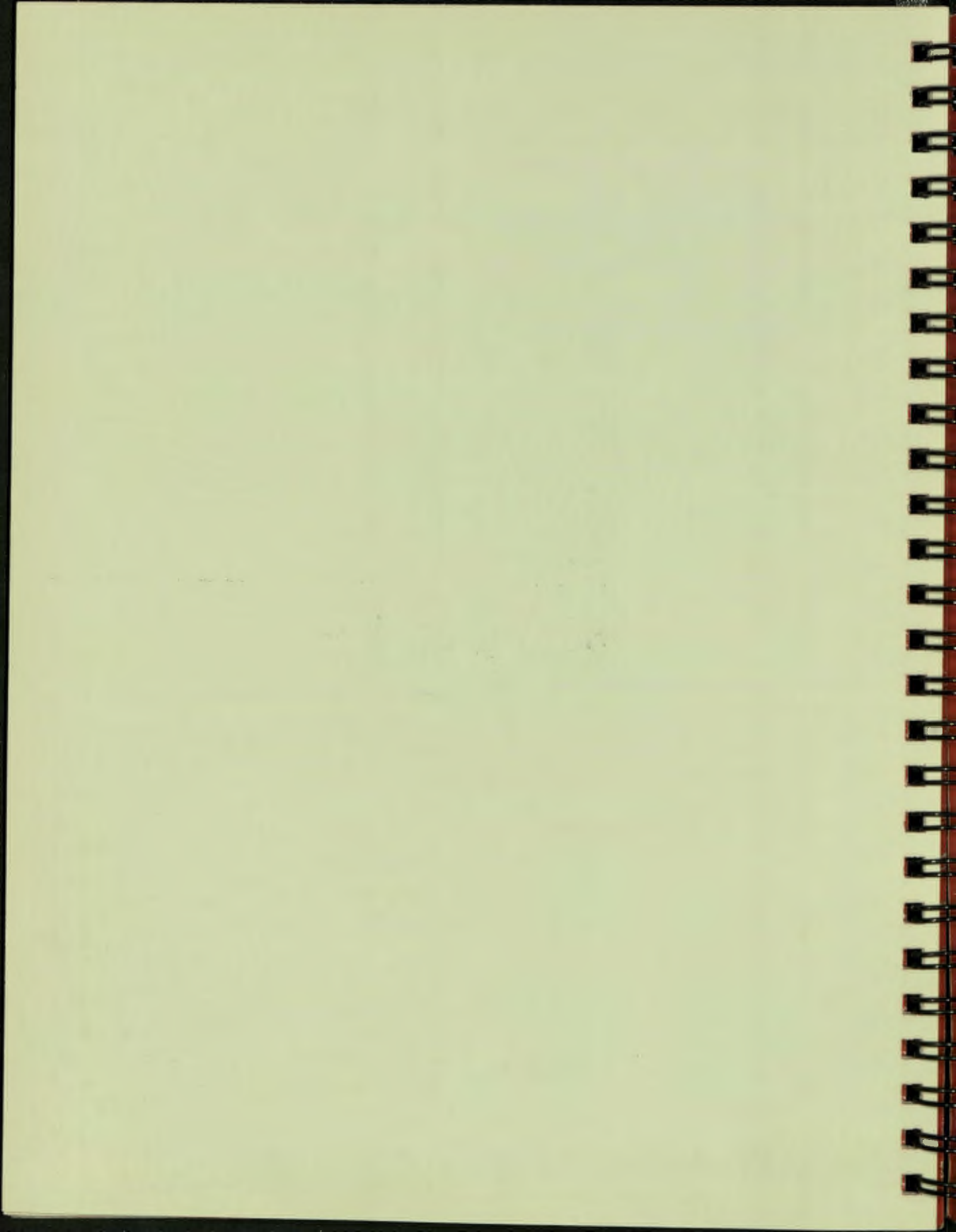
Section of text, possibly a sub-header or a specific heading.



# 3

---

Setting Up the  
Database



## Chapter 3

### Setting Up the Data Base

---

If you have ever sent an important letter to a group of customers and later discovered that you omitted one or two customers, you know how vital it is to maintain a current and orderly mailing list and contact file. Or, if you have ever lost an address or phone number, you know how frustrating an out-of-date or disorganized filing system can be.



In this chapter, you will use the Application Design Tool (ADT) to define a domain, record definition, and data file for customer information. You could just as easily use ADT to define a database for suppliers or friends, depending on your needs. Feel free to use field names that have meaning for you. This helps to create a database that you can use with ease.

ADT prompts you for all necessary information and sends you messages explaining your choices. Some information you need to supply includes:

- The domain name
- The data file name
- The name of each field in the record
- The type of information the field contains (MONEY, DATE, NUMBERS, CHARACTERS, or PERCENT)
- The format of fields containing dates, numbers, or money
- The length of the field

At the end of the ADT session, you can begin to store data.

ADT is especially helpful when you are beginning to use DATATRIEVE and want to define your own domains. As you gain experience, you can use ADT to create the basic definitions, then you can edit these definitions to include other features.

### **ANALYZING THE DATA**

The first step in creating a PRO/DATATRIEVE database involves identifying the information you want to store in each company record. To create a customer mailing list, the record should include:

- The company name, contact person, and address
- The customer's phone number, if you sometimes contact your customers by phone
- The date of your last contact with the customer, if you try to contact your customers on a regular basis
- The total sales to the customer for the current calendar year, to keep up-to-date on your sales

### Identifying the Fields

PRO/DATATRIEVE records consist of fields, so the next step in your data analysis is to identify the fields you want the record to contain. If you consider each piece of information to be a field, your record would have seven fields:

- Company name
- Contact person
- Address
- Phone number
- Last contact date
- Total sales for the current year

An address, though, usually consists of several items of information (street number and name, city, state, ZIP code, and maybe country) that can be considered separate fields. If you store all address information in one field (for example, 53 Webster Avenue, Westford, NH, 03062), you cannot sort records by state, city, or ZIP code, and you can never display anything but the entire address.

Since you may want to display only a portion of a company's address or sort records by city or state, make the parts of an address separate fields.

### Determining Field Characteristics

Now that you know what fields you want your record to contain, you need to determine the characteristics of each field. Characteristics include:

- The field's name
- The field's *query name*
- The field's size
- The field's data type

A field name must begin and end with a letter and may contain up to 31 numbers, hyphens, or underscores. It cannot contain spaces. PRO/DATATRIEVE changes hyphens in field names and other PRO/DATATRIEVE names to underscores. Since it is easier to type a hyphen than to type an underscore, the examples in this book use hyphens in names. When PRO/DATATRIEVE displays a name, though, you see underscores where you typed hyphens.

To make your record definition easier to understand, give fields meaningful names like COMPANY-NAME, CONTACT-PERSON, STREET, LAST-CONTACT, and SALES-THIS-YEAR.

If you give a field a name longer than seven characters, like COMPANY-NAME, CONTACT-PERSON, ZIP-CODE, LAST-CONTACT, or SALES-THIS-YEAR, ADT asks you for a query name. When you manipulate records, you can use query names such as COMPANY, CONTACT, LAST-C, or SALES as short and easy substitutes for long field names. When ADT asks for a query name, type in the name you have chosen. If you do not want to use a query name for a field, just press DO.

The size of a field determines how many characters you can store in the field. When you specify a field's size, make sure each field is long enough for the information you want it to hold, but not so long that the field wastes computer storage space. A company name, for example, usually consists of 20 characters at most, while a state abbreviation consists of only 2 characters.

The data type of a field determines how PRO/DATATRIEVE stores and manipulates data in the field. PRO/DATATRIEVE data types include numbers, characters, dates, percentages, and money.

PRO/DATATRIEVE allows you to choose from four date formats:

- MM/DD/YY (6/29/79)
- DD-MMM-YY (29-JUN-79)
- DD-MMM-YYYY (29-JUN-1979)
- DD.MM.YY (29.06.79)

PRO/DATATRIEVE allows you to specify the size of a money field and ADT formats the field so that values are displayed as money values, with a dollar sign in front and a decimal point. When ADT asks you how many digits to the left of the decimal point a money field contains, the number you supply determines the size of the field.

The first seven fields of the customer record consist of character data, while the LAST-CONTACT field consists of date values and SALES consists of money values, so an outline of your customer record fields might now look like this:

Field Name	Query Name	Size	Data Type
COMPANY-NAME	COMPANY	20	Character
CONTACT-PERSON	CONTACT	15	Character
STREET		15	Character
CITY		10	Character
STATE		2	Character
ZIP-CODE	ZIP	5	Character
PHONE-NUMBER	PHONE	12	Character
SALES-THIS-YEAR	SALES	7	Money
LAST-CONTACT	LAST-C	8	Date

The order of the fields in a record definition determines the way PRO/DATATRIEVE normally displays values stored in the record. If you give ADT the fields in the order shown above, PRO/DATATRIEVE displays the COMPANY-NAME field on the left side of the screen, then each successive field, ending with the LAST-CONTACT field. You can change this order for a particular report or screen display, but pay attention at this stage to the order of your record fields. If you want PRO/DATATRIEVE to routinely display the SALES field next to the COMPANY-NAME field, you have to give ADT the SALES field right after the COMPANY-NAME field.

#### **Naming the Domain and Data File**

The last steps you have to take in designing your database are choosing names for your customer domain and data file, and determining what type of file you want to use to store your records.

Domain names, like field names, must begin and end with a letter and may contain up to 31 numbers, hyphens, and underscores. PRO/DATATRIEVE converts hyphens to underscores.

Because you use domain names so frequently in PRO/DATATRIEVE, you should choose a meaningful name that is easy to remember and to type. The name CUSTOMER-INFO, for example, describes a customer information domain very clearly, but a shorter version of the name, like CUSTOMERS, can save you typing time and still have meaning.

Records also need names so that they can be defined in the data dictionary. You don't have to think up a record name, though. ADT creates a record name for your record by adding "-REC" to the domain name you choose. If you name the domain CUSTOMERS, ADT names the record CUSTOMERS-REC.

The domain links your record definition with a data file, so you have to provide a name for the data file. As with other files on your Professional 350, a data file name can contain from one to nine letters (A-Z) and/or numbers (0-9). ADT creates the file in your current directory and identifies it as a .DAT file in your directory listing. As with domain names, it is a good idea to give your data files meaningful names, so call the customer file CUSTOMERS. ADT names the file CUSTOMERS.DAT.

ADT next asks whether you want a *sequential data file* or an *indexed data file*. You cannot erase records from a sequential file, although you can erase all fields in a record. Consequently, records you no longer want or need waste memory space on a diskette or disk.

You can erase records from an indexed file, so choose an indexed file.

ADT then asks you for the name of the primary key field for your file. An *index key* is a field of your record that identifies the record in the file. You probably already use index keys to order your paper files. Paper records filed alphabetically by company name, for example, use the company name as the *primary index key*.

You cannot change the value of a primary key, so choose a field you do not expect to change very often. The only way you can change the value of a primary key field is to erase the entire record and store the data as a new record using the new primary key value.

ADT next asks if you want to allow duplicates for the primary key field. If you do not allow duplicates, you cannot have records with the same primary field value. Unless you are certain a primary key field will always contain unique values, such as an ID number, tell ADT to allow duplicates. When ADT asks, choose COMPANY-NAME as the primary key and allow duplicates.

ADT also asks if you want to name *alternate keys* for your file. An alternate key is another field that PRO/DATATRIEVE can use when organizing records in an indexed file. Unlike primary keys, you can change alternate key values. If you want to, you can specify alternate keys for your file, but they are not necessary. Alternate index keys also take up space in memory, so use them only when you really need them.



**DEFINING DATA WITH ADT**

The hard part of defining a database is now behind you. You know what fields you want the record definition to contain, what you want to name the domain and file, and what type of file you want to use to store the records. Now you just have to feed the results of your data analysis to ADT.

**TRY IT**

To start ADT, type ADT and press DO. ADT asks if you want help, so type YES (or just Y) and press DO.

Read the information ADT displays and answer the next question. Since this is your first time using ADT, you should answer this question YES so that ADT will ask you detailed questions.

ADT provides complete directions as you use it, prompts you for the information needed to define your domain, record, and file, and provides messages to explain the results of choices it asks you to make. You can press the HELP key at any point where ADT is waiting for a response from you and ADT will display helpful and explanatory information. The first time you use ADT, use the HELP key frequently to learn more about what ADT does.

From here on you can read ADT's instructions and type in the information ADT needs to define your domain, record, and data file. You can create the CUSTOMERS domain by responding to ADT prompts with the following information:

- The domain-name: CUSTOMERS
- The file name for the data: CUSTOMERS.DAT
- The record fields for the domain:

Field Name	Query Name	Data Type	Size
COMPANY-NAME	COMPANY	Character	20
CONTACT-PERSON	CONTACT	Character	15
STREET		Character	15
CITY		Character	10
STATE		Character	2
ZIP-CODE	ZIP	Character	5
PHONE-NUMBER	PHONE	Character	12
SALES-THIS-YEAR	SALES	Money	7
LAST-CONTACT	LAST-C	Date	8

- File type: Indexed
- Field name of the primary key: COMPANY
- Duplicates for the primary key field: Yes
- Alternate index keys: No
- The file name for the domain and record definitions: CUSTOMERS

When ADT asks how many digits you want to the left of the decimal point for the SALES-THIS-YEAR field, specify 5 so that your record looks like the one used in examples in this book. Likewise, choose the DD-MMM-YY format for the LAST-CONTACT field.

Remember that every time a field name exceeds seven characters, ADT asks for a query name. If you do not want to use a query name, just press DO in response to ADT's question. Remember that you can get help at any time by pressing the HELP key and that you can see your record definition by typing an exclamation point and pressing DO. To save typing time, you can type C instead of CHARACTERS and Y and N instead of YES and NO.

If you want to create a domain to contain data familiar to you, use the following outline to summarize the information you need to give ADT:

- The domain-name:
- The file name for the data:
- The record fields for the domain:

Field Name	Query Name	Data Type	Size
------------	------------	-----------	------

- File type:
- Field name of the primary key:
- Duplicates for the primary key field:
- Alternate index keys:
- The file name for the domain and record definitions:

Don't worry about making mistakes. You can use the delete key ( $\langle X \rangle$ ) to erase anything you have typed if you have not pressed DO. If you forget what you have already typed, type an exclamation point (!) and press DO to see the record definition you have defined so far. You cannot correct a mistake after you press DO. If you make a lot of mistakes, exit from ADT by pressing the EXIT key. If you do this, everything you have entered disappears and you can start over again.

The last part of your ADT session should look like this:

```
What is the name of the file where the DATATRIEVE
domain and field definitions should go? : CUSTOMERS
The DATATRIEVE definitions for your domain are
located in file DW1:[CUSTOMERS]CUSTOMERS.CMD;1
The record length is 94 bytes.
Do you want to define another domain? (YES or NO) : N
DTR>
```

ADT tells you that the definitions for your domain are stored in CUSTOMERS.CMD and that the record created by the record definition is 94 bytes, or 94 characters, long.

Note that CUSTOMERS.CMD is not the same as the CUSTOMERS data file that will contain your records. ADT puts all the information you provide for field names and characteristics into PRO/DATATRIEVE commands and writes these commands to CUSTOMERS.CMD, a *command file*. You can then "command" PRO/DATATRIEVE to execute the commands in the file just as though you typed them in yourself.

Do not confuse command files, which have a .CMD file type, with data files, which have a .DAT file type. PRO/DATATRIEVE can execute the command file, while a data file contains information stored in records.

The DTR> prompt indicates that you need to tell PRO/DATATRIEVE what to do next. Type an at sign (@) and the command file name (CUSTOMERS) and press DO. PRO/DATATRIEVE knows to look for CUSTOMERS.CMD and to execute the commands it contains. The PRO/DATATRIEVE commands created by ADT appear on your screen as they execute and you can see what PRO/DATATRIEVE *data definition commands* look like.

Data definition commands, like DEFINE DOMAIN, DEFINE RECORD, and DEFINE FILE, define PRO/DATATRIEVE dictionary objects. If you used the domain name, file name, and field names and characteristics outlined in the first part of this chapter, the DEFINE statements you see look something like this:

```

DTR> @CUSTOMERS
DEFINE DOMAIN CUSTOMERS USING CUSTOMERS-REC
ON CUSTOMERS;
DEFINE RECORD CUSTOMERS-REC USING
01 CUSTOMERS-REC.
  15 COMPANY-NAME PIC IS X(20) ←————— (1)
     QUERY-NAME IS COMPANY.
  15 CONTACT-PERSON PIC IS X(15)
     QUERY-NAME IS CONTACT.
  15 STREET PIC IS X(15).
  15 CITY PIC IS X(10).
  15 STATE PIC IS X(2).
  15 ZIP-CODE PIC IS X(5)
     QUERY-NAME IS ZIP.
  15 PHONE-NUMBER PIC IS X(12)
     QUERY-NAME IS PHONE.
  15 SALES-THIS-YEAR PIC IS S9(5)V99 ←————— (2)
     EDIT-STRING IS $$$,$$$,$$ ←————— (3)
     QUERY-NAME IS SALES.
  15 LAST-CONTACT USAGE IS DATE ←————— (4)
     EDIT-STRING IS DD-MMM-YY ←————— (5)
     QUERY-NAME IS LAST-C.
;
[Record CUSTOMERS_REC is 94 bytes long]
DEFINE FILE FOR CUSTOMERS KEY = COMPANY_NAME;
DTR>

```

The PIC, USAGE, EDIT-STRING, and QUERY-NAME phrases describe characteristics of a record field and are called *field definition clauses*. ADT created the PIC, USAGE, and EDIT-STRING clauses based on your responses to questions about the type of data contained in the field and the length of the field. Whenever you specified a QUERY-NAME, ADT included a QUERY-NAME clause with the query name you supplied.

Each PIC IS clause tells PRO/DATATRIEVE what type of data the field contains and how long the field can be. PIC is an abbreviation for PICTURE. An X in the PIC clause ((1)) indicates that the field contains character data. Numbers in parentheses indicate how many characters can be stored in each field. Thus, PIC IS X(20) tells PRO/DATATRIEVE that the COMPANY-NAME field can contain as many as 20 characters.

A 9 in a PIC clause ((2)) indicates that the field contains numeric digits. PIC IS S9(5)V99 clause tells PRO/DATATRIEVE that the SALES field can be positive or negative (S), that there may be five digits (5) to the left of the decimal point (V), and that there may be two digits to the right of the decimal point. The SALES field, then, can contain 7 digits, 9(5) and 99.

PRO/DATATRIEVE stores values in the SALES field without the sign and decimal point and with leading zeros. The value \$123.45, for example, is stored as 0012345. Since you need to know where the decimal point is and probably don't want to see the leading zeros in the field, ADT conveniently adds an EDIT-STRING clause (③) that tells PRO/DATATRIEVE to format sales values as money values. The edit string \$\$\$,\$\$\$.\$\$ tells PRO/DATATRIEVE to display SALES values with leading spaces instead of zeros, a leading dollar sign, a comma between hundreds and thousands, and a decimal point. Each dollar sign except the first reserves space for one digit, while the comma and decimal point reserve places for themselves. Thus, PRO/DATATRIEVE displays a value like 0012345 as \$123.45 and a value of 0000000 as a blank.

The USAGE IS DATE clause (④) tells PRO/DATATRIEVE that LAST-CONTACT is a date field. The DATE keyword reserves 8 places for date information — two places for the day, two for the month, and four for the year. You chose the EDIT-STRING clause for this field (⑤) when you told PRO/DATATRIEVE to use format 2. Thus, PRO/DATATRIEVE formats the LAST-CONTACT field as a two-digit day (DD), a three-letter month abbreviation (MMM), and a two-digit year (YY), separated by hyphens. PRO/DATATRIEVE does not store the hyphens in the data file.

If you add all the places reserved by the PIC and USAGE clauses together, you can see where PRO/DATATRIEVE came up with the 94 byte message.

As you gain experience with PRO/DATATRIEVE and learn more about field definition clauses, you may want to continue using ADT to create basic record definitions like CUSTOMERS-REC. Then you can edit the command file that contains the definitions to include other field definition clauses. The *PRO/DATATRIEVE Handbook* contains more information on field definition clauses.

When the command file finishes executing, your domain and record definitions are in your dictionary and your data file is ready for you to fill with customer records. If you want, you can use the SHOW command to see the definitions listed in your dictionary. If, at any time, you want to see what the CUSTOMERS-REC record definition or the CUSTOMERS domain definition looks like, type SHOW CUSTOMERS-REC or SHOW CUSTOMERS.

**GETTING DOMAINS READY**

Before you can look at or store information in a paper or card file, you have to get the file ready. If you want to add or modify records, you have to find a pencil and paper and locate the correct file. If you want to read records in a paper file, you also have to get ready, but not in the same way since you do not need the pencil and paper. In other words, you get a file ready for a specific type of *access*, depending on what you want to do with records in the file.

As you learned in Chapter 2, you get PRO/DATATRIEVE files ready with the **READY** command. In the **READY** command, you specify the domain name associated with the file and the type of access you want to the file, like this:

```
DTR> READY CUSTOMERS WRITE
```

This command tells PRO/DATATRIEVE to get the CUSTOMERS domain ready for **WRITE** access. **WRITE** access allows you all access privileges to records in the CUSTOMERS domain: read, modify, erase, and store. Other types of access, **READ** and **MODIFY**, allow you more limited privileges.

Table 3-1 lists access types and privileges.

**Table 3-1: READY Command Access Types and Privileges**

<b>Access Type:</b>	<b>Access Privileges:</b>
READ	You can only <i>read</i> records
MODIFY	You can read and <i>modify</i> records
WRITE	You can read, modify, <i>store</i> , and <i>erase</i> records

The **READY** command automatically gives you **READ** access to a domain, so you do not have to specify **READ** in the **READY** command. **READY CUSTOMERS** and **READY CUSTOMERS READ** are equivalent commands.

**TRY IT**

Type in the **READY** command and press **DO**. PRO/DATATRIEVE readies the domain and displays the **DTR>** prompt again. To verify the ready state of **CUSTOMERS**, tell PRO/DATATRIEVE to show you all the ready domains by using the **SHOW READY** command:

```
DTR> READY CUSTOMERS WRITE
DTR> SHOW READY
Ready domains:
      CUSTOMERS: RMS INDEXED, PROTECTED WRITE
DTR>
```

As you can see from the message, the CUSTOMERS domain is now ready to store your customer records.

### STORING INFORMATION

Just as you have to go through the process of writing customer records on paper and storing those records in a file, you now have to go through the process of writing all your customer records in the data file. Since you probably do not want to waste too much time typing in meaningless data, the following sections ask you to store only ten records in the CUSTOMERS domain. Feel free to store more if you want, but you only need to store ten records to do examples that use the CUSTOMERS domain.

#### Storing One Record

As you learned in Chapter 2, you tell PRO/DATATRIEVE to store records in the CUSTOMER domain with the `STORE CUSTOMERS` statement. PRO/DATATRIEVE then prompts you for a value for each field in the CUSTOMERS record. Statements are somewhat different from commands because statements do something with or to data. Commands, like `READY`, `SHOW`, `SET`, and the `DEFINE` commands, do something with or to dictionary objects and your PRO/DATATRIEVE environment. Furthermore, as discussed later in this book, you can make PRO/DATATRIEVE automatically repeat a statement or execute a group of statements as a unit.

Make it a habit to use only uppercase letters for field values. PRO/DATATRIEVE, in general, considers uppercase and lowercase letters as different letters and does not know that a company named "WIDGET" is the same as a company named "Widget" or "widget". If you get in the habit of using uppercase letters in records and in the record selection expressions you use to locate records, you save yourself problems and typing time.

#### TRY IT

Type the `STORE` statement and type in values for each field in the record in response to PRO/DATATRIEVE's prompts, like this:

```
DTR> STORE CUSTOMERS
Enter COMPANY_NAME: LOGIC SYSTEMS CO.
Enter CONTACT_PERSON: GEORGE BOOLE
Enter STREET: NOTTINGHAM CT
Enter CITY: LONDON
Enter STATE: MA
Enter ZIP_CODE: 03053
Enter PHONE_NUMBER: 617-555-9981
Enter LAST_CONTACT: 4-17-81
Enter SALES_THIS_YEAR: 0
DTR>
```

When you have entered information for all the fields in the record, the STORE statement ends and PRO/DATATRIEVE displays the DTR> prompt.

To see how PRO/DATATRIEVE protects you from making mistakes, type STORE CUSTOMERS again and try to store MASS in the STATE field. Because MASS consists of four characters and the STATE field can only contain two characters, PRO/DATATRIEVE beeps to indicate that something is wrong with your value, displays the message "Truncation during assignment", and prompts again for a STATE value. PRO/DATATRIEVE would have to drop some of the letters of the value you typed (MASS) in order to store the value in the STATE field, but it cannot make that decision for you.

You can type MA and continue storing values in the record, or you can stop the STORE statement by pressing the EXIT key. If you press EXIT, PRO/DATATRIEVE tells you that you have stopped execution and does not store any of the information you typed before pressing EXIT. Since examples in this book use specific customer records, you may not want to store this record. If you have already finished entering values, don't worry. Chapter 6 shows you how to delete records.





### Storing More Than One Record at a Time

You've already stored one customer record, but to do the examples in this book, you need to store nine more customer records in the CUSTOMERS domain. You could type STORE CUSTOMERS nine more times, but PRO/DATATRIEVE has a statement that lets you repeat another statement any number of times — the REPEAT statement. To repeat the STORE statement nine times, type REPEAT 9 STORE CUSTOMERS and press DO. You can also type REPEAT 9 and STORE CUSTOMERS as separate lines. Either way, PRO/DATATRIEVE will prompt you to enter data until nine records have been stored.

#### TRY IT

Use the REPEAT and STORE statements to store the following customer records in the CUSTOMERS domain. Some records, as you can see, do not have any value in the SALES field, while others have a zero. Also, two records do not have a ZIP code. To leave a field empty or to enter a value of zero, press the space bar before pressing DO.

If you make a mistake, use the delete key to erase it. If you have already pressed DO and notice a mistake, don't worry. Chapter 6 shows you how to change and delete records.

If you want to stop storing records part way through, press the EXIT key. The EXIT key stops the REPEAT statement and returns you to the DTR> prompt. If you press EXIT while in the middle of storing a record, PRO/DATATRIEVE does not store any portion of that record. Records completely entered before you press EXIT are stored in the database. Type new REPEAT and STORE statements to add the remaining records to the domain.

#### Customer Records for the CUSTOMERS Domain

Company:	STONE ASSOCIATES
Contact:	LAMAR PAUL
Street:	HARVARD RD
City:	CONWAY
State:	MA
ZIP code:	13425
Phone:	617-555-4423
Sales:	2091.00
Last contact:	2-22-83

Company:	PRODUCTS UNLIMITED
Contact:	DANA MCMANUS
Street:	161 WILLOW
City:	NAUSET
State:	NH
ZIP code:	
Phone:	603-555-3886
Sales:	
Last contact:	4-13-82

continued on next page

## Customer Records for the CUSTOMERS Domain (cont.)

Company: INFO ANALYSIS  
 Contact: WILMA SWAYER  
 Street: 9 REDACTOR  
 City: DUNKIRK  
 State: MA  
 ZIP code:  
 Phone: 617-555-7114  
 Sales: 309.00  
 Last contact: 3-2-83

Company: PROSE PROS  
 Contact: LEONARD CARON  
 Street: 14 HARBOR AVE  
 City: WEST ELMO  
 State: NH  
 ZIP code: 03422  
 Phone: 603-555-9449  
 Sales: 1053  
 Last contact: 2-15-83

Company: KEY SPECIALISTS  
 Contact: DANE REED  
 Street: 34 PROSPECT ST  
 City: CANDIDA  
 State: NH  
 ZIP code: 03422  
 Phone: 603-555-0570  
 Sales: 1025  
 Last contact: 12-11-82

Company: TACTICAL AID, INC.  
 Contact: RICHARD BOWDEN  
 Street: 10 CONTINENTAL  
 City: MAXWELL  
 State: MA  
 ZIP code: 34672  
 Phone: 617-555-3723  
 Sales: 0  
 Last contact: 3-2-83

Company: BASIC SYSTEMS  
 Contact: ANNE DUGGAN  
 Street: 43 HIGHLAND AVE  
 City: BOSTON  
 State: MA  
 ZIP code: 13532  
 Phone: 617-555-8495  
 Sales:  
 Last contact: 3-1-83

Company: PAPER & PEN, INC.  
 Contact: WINSLOW JOHNS  
 Street: 32 MAIN ST  
 City: NATICOOK  
 State: NH  
 ZIP code: 03412  
 Phone: 603-555-7281  
 Sales: 2031.00  
 Last contact: 1-9-83

Company: WRITE RIGHT ASSOC.  
 Contact: SONDR A CHAPLIN  
 Street: WEST LAKESIDE  
 City: NEW WILTON  
 State: NH  
 ZIP code: 03883  
 Phone: 603-555-5085  
 Sales: 0  
 Last contact: 2-13-83

## DISPLAYING INFORMATION

Now you can see the results of all your planning and typing by displaying all the customer records on your screen. As you learned in Chapter 2, you use the PRINT statement to display records.

### TRY IT

Use the SHOW READY command to verify that the CUSTOMERS domain is ready. Then, to see all the customer records, tell PRO/DATATRIEVE to display all the records in the domain by typing PRINT CUSTOMERS.

Notice that PRO/DATATRIEVE displays field names as column headers and displays all the fields of all the records. The records appear in alphabetical order according to the primary key, COMPANY. Unfortunately, though, the record does not fit on a single line, so PRO/DATATRIEVE *wraps* the record to the next line so that you can see all the fields of a record. Some of the column headers get lost in the wrapping process and the record display looks odd. Sometimes a record you think should fit on the screen doesn't fit because the column headers take up more space than the field.

What can you do to create a more orderly and useful display? Since you probably don't need all the data stored in a record at any one time, the easiest way to solve this problem is to display only the fields you want to see. To do this, tell PRO/DATATRIEVE the names of the fields you want to see and the name of the domain. To name the fields, specify the field names separated by commas, then specify the domain name by typing OF CUSTOMERS, like this:

```
DTR> PRINT COMPANY, STREET, CITY, STATE, ZIP OF CUSTOMERS
```

COMPANY NAME	STREET	CITY	STATE	ZIP CODE
BASIC SYSTEMS	43 HIGHLAND AVE	BOSTON	MA	13532
INFO ANALYSIS	9 REDACTOR	DUNKIRK	MA	
KEY SPECIALISTS	34 PROSPECT ST	CANDIDA	NH	03422
LOGIC SYSTEMS CO.	NOTTINGHAM CT	LONDON	MA	03053
PAPER & PEN, INC.	32 MAIN ST	NATICOOK	NH	03412
PRODUCTS UNLIMITED	161 WILLOW	NAUSET	NH	
PROSE PROS	14 HARBOR AVE	WEST ELMO	NH	03422
STONE ASSOCIATES	HARVARD RD	CONWAY	MA	13425
TACTICAL AID, INC.	10 CONTINENTAL	MAXWELL	MA	34672
WRITE RIGHT ASSOC.	WEST LAKESIDE	NEW WILTON	NH	03883

To see contact, phone, sales, and last contact date information for each company, use those field names in the PRINT statement:

```
DTR> PRINT COMPANY, CONTACT, SALES, LAST-C OF CUSTOMERS
```

Do not forget to include OF CUSTOMERS so PRO/DATATRIEVE knows which domain contains the fields you want to see. If you forget, PRO/DATATRIEVE displays an error message.

If you want, display various combinations of fields until you are comfortable using the PRINT statement. As you continue through this book, you learn to use many forms of the PRINT statement to display information on your screen and to create reports that you can print.

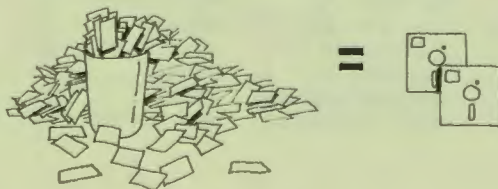
### FINISHING DOMAINS

When you finish working with the CUSTOMERS domain, use the FINISH command to end your access to the domain. Since you have only one ready domain, CUSTOMERS, type FINISH and press DO. If you had more than one domain ready, you could finish only the CUSTOMERS domain by typing FINISH CUSTOMERS. As you work through this book, you will learn more about the FINISH command.

### SUMMARY

In this chapter you learned to analyze data and use ADT to define a domain, record, and file for that data. You also learned to use the READY command to get access to a domain, the STORE and REPEAT statements to store information in a domain, the PRINT statement to display various record fields, and the FINISH command to end your access to a domain.

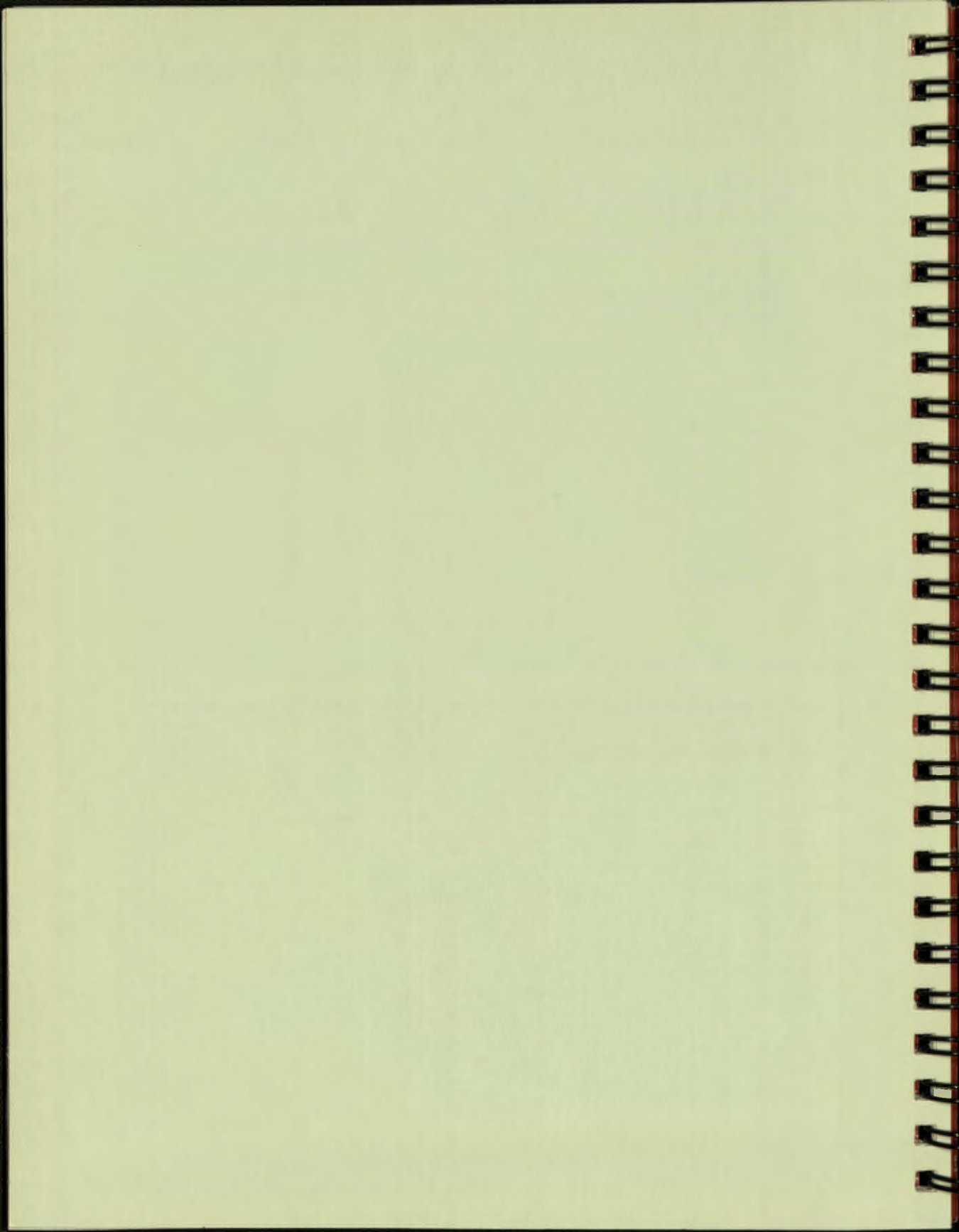
In the next chapter you find out how to get specific information from your database.



# 4

---

## Getting Information

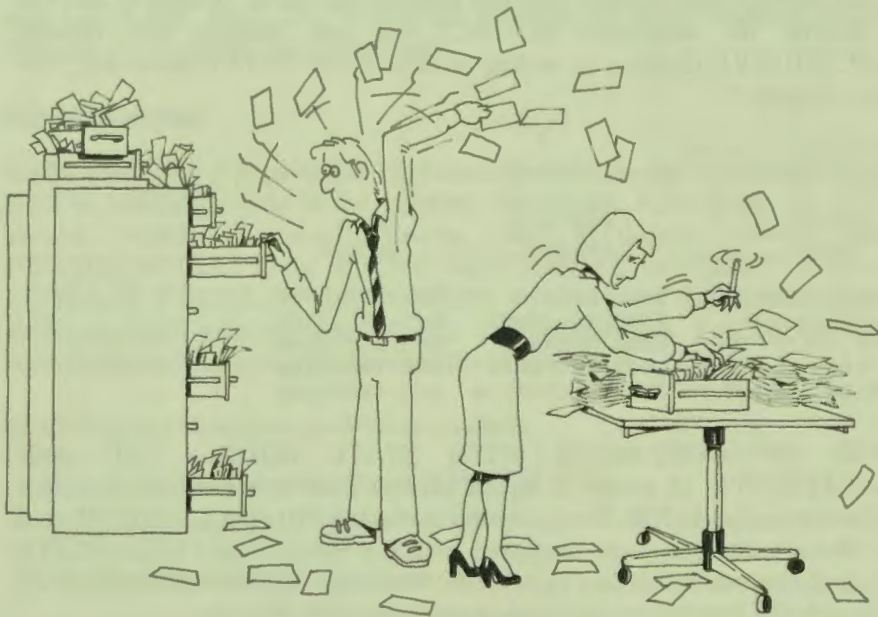


## Chapter 4

---

### Getting Information

Domains you create to manage your information will probably contain many more records than you have just stored in the CUSTOMERS domain. With only ten records, you can easily find any information you need by displaying all fields for all the records. Looking at all the records in a large database, though, is not an efficient way to get and organize information. What if you want information only for current customers? What if you want to create a mailing list of prospective customers?



You can solve those data management problems by using record selection expressions (RSEs) to form *record streams* and record collections. This chapter shows you how to form record streams to display the information you want. Unlike records stored in a data file, a record stream ceases to exist when the statement that created it finishes executing. The next chapter shows you how to form record collections.

## FORMING RECORD STREAMS

A record stream is a group of related records that you gather together temporarily. When you look through a stack of index cards and pick out only current customer records, you have formed a record stream that excludes all former and prospective customer records. If you look through that same stack of cards and pick out only current customer records in the state of NH, you have formed a different and more limited record stream.

To form a record stream of index cards, you probably think to yourself, “I want to see current customer records for companies in New Hampshire,” or “I want to see only the first five customer records in the file.” To form a record stream in PRO/DATATRIEVE, you use a record selection expression (RSE) to specify the conditions a record must meet to be included in a record stream.

Just as you identify what records you look at when you think, “I want to see customer records for companies in NH,” you can identify the records PRO/DATATRIEVE displays by adding an RSE to the PRINT statement, like you did in Chapter 2:

```
DTR> READY CUSTOMERS
DTR> PRINT COMPANY, CONTACT, PHONE OF CUSTOMERS WITH STATE EQUAL "NH"
```

COMPANY NAME	CONTACT PERSON	PHONE NUMBER
KEY SPECIALISTS	DANE REED	603-555-0570
PAPER & PEN, INC.	WINSLOW JOHNS	603-555-7281
PRODUCTS UNLIMITED	DANA MCMANUS	603-555-3886
PROSE PROS	LEONARD CARON	603-555-9449
WRITE RIGHT ASSOC.	SONDRA CHAPLIN	603-555-5085

The RSE, OF CUSTOMERS WITH STATE EQUAL “NH”, tells PRO/DATATRIEVE to create a record stream that includes only customer records for companies in NH. The quotation marks tell PRO/DATATRIEVE that NH is a literal value you want compared with the values stored in the STATE field. To limit that record stream even more, you could tell PRO/DATATRIEVE to display only the first two records for companies in NH, like this:



```
DTR> PRINT COMPANY, CONTACT, PHONE OF FIRST 2
[Looking for collection or domain name]
CON> CUSTOMERS WITH STATE EQUAL "NH"
```

COMPANY NAME	CONTACT PERSON	PHONE NUMBER
KEY SPECIALISTS	DANE REED	603-555-0570
PAPER & PEN, INC.	WINSLOW JOHNS	603-555-7281

As you can see, statements with RSEs can get to be too long to fit on one line. If you press DO at a point where PRO/DATATRIEVE can tell that the statement is incomplete, you see a message indicating what you need to supply to finish the statement and the CON> prompt. You can, for example, press DO at the indicated points in the following statement:

```
PRINT COMPANY, CONTACT, PHONE OF FIRST 2 CUSTOMERS WITH STATE EQUAL "NH"
      ↑           ↑           ↑           ↑↑           ↑           ↑           ↑
      DO          DO          DO          DO          DO          DO          DO
```

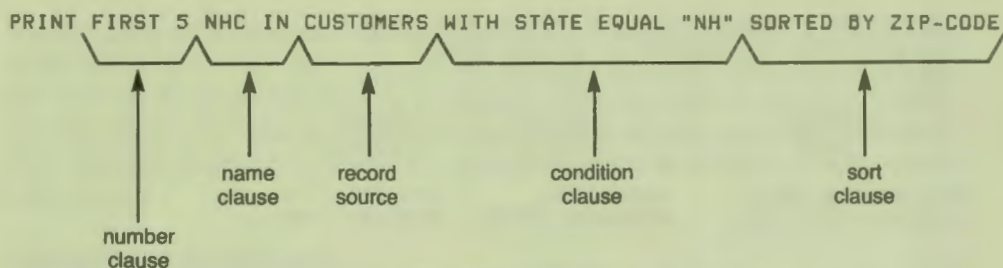
Be careful where you press DO. If you press DO at a point where PRO/DATATRIEVE considers the statement complete, say, after CUSTOMER, the statement executes and PRO/DATATRIEVE displays the DTR> prompt again. Also, if you press DO after the last field name (PHONE) and before OF, PRO/DATATRIEVE displays an error message telling you that the field names are undefined.

### FORMING AN RSE

RSEs often look complex, but if you understand the parts of an RSE and how they fit together, some of the mystery disappears. First of all, an RSE must always include a *record source*, like a domain. When you tell PRO/DATATRIEVE to PRINT CUSTOMERS or PRINT COMPANY, CONTACT, PHONE OF CUSTOMERS, you are using the simplest form of an RSE — an RSE with only one element, the record source. You will learn to use a record collection as a record source in the next chapter.

In addition to the source, an RSE can include:

- An optional *number clause* (FIRST or ALL)
- An optional *name clause*
- An optional *condition clause* (WITH)
- An optional *sort clause* (SORTED BY)

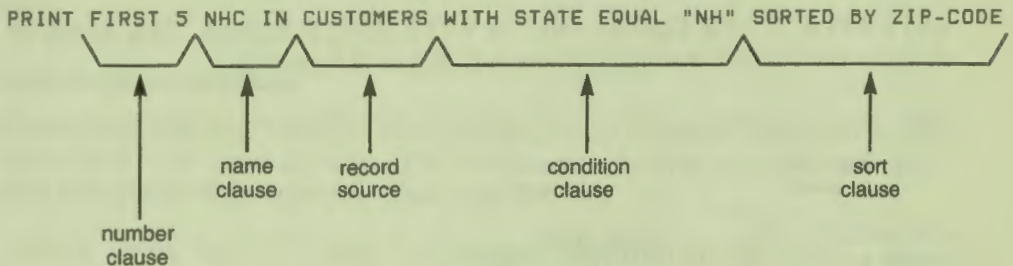


In order for PRO/DATATRIEVE to understand an RSE, you have to order the optional elements and the record source of an RSE in a particular way:

1. The number clause tells PRO/DATATRIEVE how many records, beginning with the first record in the data file, to include in a record stream. If used, this clause must precede all other RSE elements. **FIRST 5 CUSTOMERS**, for example, tells PRO/DATATRIEVE to display only the first five customer records.
2. The name clause, **NHC IN**, tells PRO/DATATRIEVE what to name a record stream. You can then refer to the record stream later by name. In the next chapter, you learn to name a group of records and to use named groups.
3. If you include a restriction clause and/or a name clause, the record source, **CUSTOMERS**, follows these clauses. In this chapter, you use domains as the record source. In the next chapter, you learn to use collections as the record source.
4. The condition clause, **WITH STATE EQUAL "NH"**, one of the most important components of an RSE, must immediately follow the record source. Once PRO/DATATRIEVE knows where to find the records, it needs to know the conditions you want records to meet. The condition clause consists of the word **WITH** and a *Boolean expression*. A Boolean expression tells PRO/DATATRIEVE how to compare records and what values to use in the comparison. A later section of this chapter describes Boolean expressions in detail. For now, all you need to know is that PRO/DATATRIEVE evaluates a Boolean expression as true or false. If the Boolean expression is true for a given record, PRO/DATATRIEVE includes the record in the record stream. **STATE EQUAL "NH"** and **STATE EQUAL "NH" AND SALES LESS-THAN 1000** are Boolean expressions. **STATE EQUAL "NH"** tells PRO/DATATRIEVE you want all New Hampshire customer records. **STATE EQUAL "NH" AND SALES LESS-THAN 1000** tells PRO/DATATRIEVE you want only records for New Hampshire customers with sales less than \$1,000.

- As the last part of an RSE, you can specify a sort clause, as in SORTED BY ZIP-CODE. The sort clause tells PRO/DATATRIEVE how to order the records in the record stream. If you do not include a sort clause, PRO/DATATRIEVE sorts records by the primary index key for the data file. The customer records you have displayed so far, for example, have been sorted by the index key, COMPANY. PRINT CUSTOMERS SORTED BY STATE tells PRO/DATATRIEVE to sort customer records by state.

That's the way, in a nutshell, components of an RSE fit together to form a record stream. Now you can see that a complex RSE like the one below simply consists of separate and identifiable components:



This RSE tells PRO/DATATRIEVE to include in the record stream the first five records with NH in the STATE field and to sort those records by ZIP code. It also tells PRO/DATATRIEVE to name the record stream COMPANIES. The following sections of this chapter show you how the number, condition, and sort clauses work together to form record streams. Chapter 5 shows how to use the name clause.

### RESTRICTING THE NUMBER OF RECORDS IN A RECORD STREAM

The number clause tells PRO/DATATRIEVE how many records, beginning with the first record, to include in the record stream. PRINT ALL CUSTOMERS tells PRO/DATATRIEVE to display all records, beginning with the first, in the CUSTOMERS domain. PRINT FIRST 2 CUSTOMERS tells PRO/DATATRIEVE to display only the first two records.

If you do not use FIRST to specify the number of records to include in the record stream, PRO/DATATRIEVE assumes you want all records that meet the conditions of the condition clause. Thus, PRINT CUSTOMERS means the same thing as PRINT ALL CUSTOMERS.

**TRY IT**

Ready the CUSTOMERS domain and display the first two records. To display only certain fields of the records, separate the fields you want to see with commas, then type OF FIRST 2 CUSTOMERS, like this:

```
DTR> PRINT COMPANY, CONTACT OF FIRST 2 CUSTOMERS
```

COMPANY NAME	CONTACT PERSON
BASIC SYSTEMS	ANNE DUGGAN
INFO ANALYSIS	WILMA SWAYER

Add a WITH STATE EQUAL "NH" or WITH STATE EQUAL "MA" condition clause to the PRINT statement to see only NH or MA records:

```
DTR> PRINT COMPANY, CONTACT OF FIRST 2 CUSTOMERS WITH STATE EQUAL "NH"
```

COMPANY NAME	CONTACT PERSON
KEY SPECIALISTS	DANE REED
PAPER & PEN, INC.	WINSLOW JOHNS

To see what happens when you specify more records than meet the conditions of the condition clause, tell PRO/DATATRIEVE to display seven MA records. Because the CUSTOMERS domain contains only four MA records, that's all PRO/DATATRIEVE can display.

**ESTABLISHING CONDITIONS FOR THE RECORD STREAM**

The condition clause may be the most important component of an RSE, because a record stream is useful only if it contains the records you want. A condition clause tells PRO/DATATRIEVE how to compare records and what values to use in the comparison. You establish conditions for a record stream by setting up Boolean expressions for PRO/DATATRIEVE to evaluate. Only records that satisfy the conditions you specify are included in the record stream.

Boolean expressions consist of field names (①), relational operators (②), values (③), and Boolean operators (④), like this:

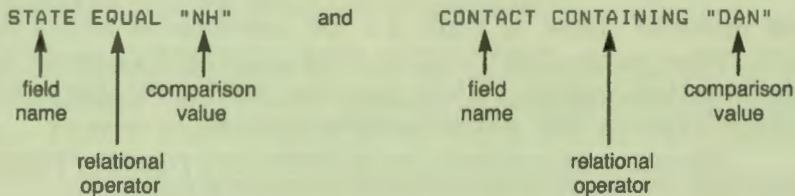
```
STATE EQUAL "NH" AND SALES LESS-THAN 1000
  ↑       ↑       ↑       ↑       ↑       ↑       ↑
  ①       ②       ③       ④       ①       ②       ③
```

A nineteenth-century English mathematician named George Boole bequeathed his name and system of logic to PRO/DATATRIEVE, although he did not live to see himself so immortalized. Although Boole did not “invent” simple expressions that compare one value with another, like STATE EQUAL “NH”, he did identify a way to join simple expressions together to form compound expressions. The keys to Boole’s logic system, as used by PRO/DATATRIEVE, are the *Boolean operators*, AND and OR.

Boolean expressions allow you to identify precisely the record or records you want to include in a record stream, thus serving as the cornerstone for PRO/DATATRIEVE’s versatility and power. In the following sections, you learn to use relational operators and Boolean operators to build Boolean expressions for RSEs.

### Using Relational Operators

A relational operator tells PRO/DATATRIEVE to compare the contents of a record field with a certain value. The field name, relational operator, and value form a simple Boolean expression that looks like this:



The Boolean expression on the left specifies that values in the STATE field must equal NH. The one on the right specifies that values in the CONTACT field must contain the characters DAN. When you use the CONTAINING expression in a PRINT statement, PRO/DATATRIEVE finds two customer records:

```
DTR> PRINT COMPANY, CONTACT OF CUSTOMERS WITH CONTACT
[Looking for relational operator (eq, gt, etc.))
CON>          CONTAINING "DAN"
```

COMPANY NAME	CONTACT PERSON
KEY SPECIALISTS	DANE REED
PRODUCTS UNLIMITED	DANA MCMANUS

PRO/DATATRIEVE recognizes many other relational operators besides EQUAL and CONTAINING. Table 4-1 lists some of the words you can use as relational operators and shows examples of their use.

**Table 4-1: PRO/DATATRIEVE Relational Operators**

<b>Operator</b>	<b>Meaning</b>	<b>Example</b>
EQUAL	Equal to	STATE EQUAL "NH"
NOT-EQUAL	Not equal to	STATE NOT-EQUAL "NH"
GREATER-THAN	Greater than	SALES GREATER-THAN 1000
GREATER-EQUAL	Greater than or equal to	SALES GREATER-EQUAL 1000
LESS-THAN	Less than	SALES LESS-THAN 1000
LESS-EQUAL	Less than or equal to	LAST-C LESS-EQUAL "3-1-83"
BETWEEN x AND y	Between and including the values x and y	SALES BETWEEN 1000 AND 2000
CONTAINING	Containing the specified value	CONTACT CONTAINING "dan"

All relational operators shown in Table 4-1 are *case-sensitive* except for CONTAINING. When you use CONTAINING, PRO/DATATRIEVE treats "X" and "x" as identical values. When you use any other operator, PRO/DATATRIEVE treats "NH" and "nh" as different values.

To save typing time, you can abbreviate relational operators as follows:

EQUAL	→ EQ or =
NOT-EQUAL	→ NE
GREATER-THAN	→ GT or >
GREATER-EQUAL	→ GE
LESS-EQUAL	→ LE
LESS-THAN	→ LT or <
BETWEEN	→ BT
CONTAINING	→ CONT

Feel free to use abbreviations and symbols to save yourself typing time even though most of the examples in this book use whole words for clarity.

**TRY IT**

Since the CUSTOMERS domain does not contain very many similar records, PRO/DATATRIEVE supplies a sample domain called EMPLOYEES for you to use while working through this chapter.

Type SHOW EMPLOYEES-REC to see the EMPLOYEES record definition. Most of the examples in this section use query names, like L-NAME, F-NAME, and STATUS for fields because they are shorter than the field names, so look carefully at the field and query names in the record. Ready the domain and type PRINT EMPLOYEES to display all the records in this sample domain.

Then, to get a feel for how relational operators work in RSEs, use the PRINT statement and different Boolean expressions to display different groups of records. PRINT EMPLOYEES WITH SALARY GT 30000, for example, displays records for employees earning more than \$30,000. To display a range of records, type something like PRINT EMPLOYEES WITH SALARY BETWEEN 10000 AND 20000. Dollar signs and commas are not stored in the SALARY field, so do not include them in the RSE. If you do, PRO/DATATRIEVE displays an error message.

As you experiment, notice that you can use different relational operators to produce the same record stream. To display all mathematicians, for example, you can type PRINT EMPLOYEES WITH DEPT EQUAL "MATHEMATICS" or PRINT EMPLOYEES WITH DEPT CONTAINING "MATH".

When you feel comfortable using relational operators in simple Boolean expressions, try forming a more complex expression with the EQUAL operator followed by a series of values, separated by commas:

```
DTR> PRINT ID, F-NAME, L-NAME, DEPT OF EMPLOYEES WITH L-NAME EQUAL
[Looking for a value expression]
CDN>      "BOOLE", "EINSTEIN", "HOWL", "STONE"
```

ID	FIRST NAME	LAST NAME	DEPT
00001	GEORGE	BOOLE	MATHEMATICS
00891	FRED	HOWL	MATHEMATICS
03991	JAMES	BOOLE	MATHEMATICS
29573	PAUL	STONE	LITERATURE
78375	ALBERT	EINSTEIN	MATHEMATICS
83771	GERALD	BOOLE	ASTRONOMY
93811	MARTHA	BOOLE	ASTRONOMY

You can specify a series of values only after the EQUAL, EQ, or = operators.

While you are experimenting with simple Boolean expressions, try breaking statements like the one in the preceding example into parts by pressing DO at points where PRO/DATATRIEVE can tell the statement is incomplete. PRO/DATATRIEVE displays helpful prompts, indicating what information you should supply next.

### Using Boolean Operators

By using Boolean operators, you can specify multiple or negative conditions for PRO/DATATRIEVE to use when selecting records for a record stream. You can, for example, display the only “real” GEORGE BOOLE record in the EMPLOYEES domain with the following Boolean expression:

```
DTR> PRINT EMPLOYEES WITH (L-NAME CONT "BOOLE") AND
[Looking for Boolean expression]
CON>      (STATUS EQUAL "EXPERIENCED") AND
[Looking for Boolean expression]
CON>      (DEPT EQUAL "MATHEMATICS")
```

The EMPLOYEES domain, as you may have noticed, contains records for four employees named BOOLE — GEORGE, MARTHA, JAMES, and GERALD. If you could use only simple Boolean expressions, like L-NAME CONT “BOOLE”, you would have a much harder time locating the “real” George Boole. Compound Boolean expressions allow you to specify precisely the record or records you want.

Compound Boolean expressions consist of simple Boolean expressions and the Boolean operators AND and OR.

PRO/DATATRIEVE evaluates all Boolean expressions as either true or false. When you use a simple expression such as STATE EQUAL “NH”, PRO/DATATRIEVE looks at the STATE value in each record. If the value is NH, PRO/DATATRIEVE evaluates the expression as true and includes the record in the record stream. PRO/DATATRIEVE determines whether compound Boolean expressions are true or false in a similar way, depending on the operator you use:



- When you link expressions with AND, the Boolean expression is true when both expressions are true and false when either expression is false. If both expressions are false, the Boolean expressions is also false. You can identify the “real” George Boole in the group of BOOLE namesakes with a Boolean expression like this:

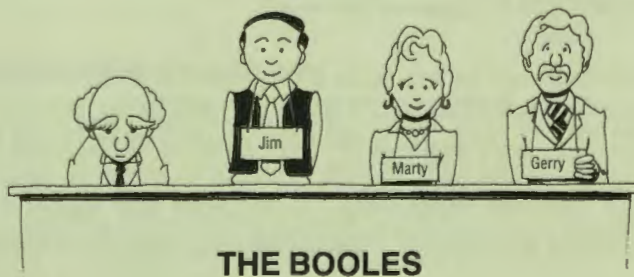
(DEPT EQUAL "MATHEMATICS") AND (STATUS EQUAL "EXPERIENCED")



MATHEMATICS (T) EXPERIENCED (T)	MATHEMATICS (T) TRAINEE (F)	ASTRONOMY (F) TRAINEE (F)	ASTRONOMY (F) EXPERIENCED (T)
GEORGE BOOLE	JAMES BOOLE	MARTHA BOOLE	GERALD BOOLE

- When you link expressions with OR, the resulting expression is true if either one or both of the expressions is true. You can find all the Boole imposters in the group of BOOLE namesakes with a Boolean expression like this:

(DEPT EQUAL "ASTRONOMY") OR (STATUS EQUAL "TRAINEE")



MATHEMATICS (F) EXPERIENCED (F)	MATHEMATICS (F) TRAINEE (T)	ASTRONOMY (T) TRAINEE (T)	ASTRONOMY (T) EXPERIENCED (F)
GEORGE BOOLE	JAMES BOOLE	MARTHA BOOLE	GERALD BOOLE

Parentheses around components of a compound Boolean expression make it easier to follow the logic of the expression. Parentheses can also ensure that your Boolean expressions work the way you want them to work. If, for example, you want to display the records of experienced employees in the math or astronomy departments, you might use a Boolean expression like this:

```
DEPT CONT "MATH" OR DEPT CONT "ASTR" AND STATUS EQUAL "EXPERIENCED"
```

Unfortunately, this Boolean expression does not do what you want it to do. PRO/DATATRIEVE first evaluates the simple expressions on either side of the OR operator and finds all math and astronomy records, both trainee and experienced. Then PRO/DATATRIEVE evaluates the simple expressions on either side of the AND operator to find astronomy department records with STATUS equal to EXPERIENCED. The record stream consequently includes experienced and trainee math records, but only experienced astronomy records.

With parentheses, you can build a Boolean expression to find the exact records you want:

```
(DEPT CONT "MATH" OR DEPT CONT "ASTR") AND (STATUS EQUAL "EXPERIENCED")
```

Because of the parentheses, this Boolean expression tells PRO/DATATRIEVE to locate the records of experienced math and experienced astronomy personnel and to ignore trainee mathematicians and astronomers. Parentheses identify the simple expressions on either side of the OR operator as a single unit linked by AND with (STATUS EQUAL "EXPERIENCED"). Thus, PRO/DATATRIEVE evaluates the OR part of the AND expression to find "true" math or astronomy records, then evaluates (STATUS EQUAL "EXPERIENCED") to identify "true" experienced records. If either part of the AND expression evaluates to false, PRO/DATATRIEVE ignores the record.

Even though parentheses do not make a difference in Boolean expressions like PRINT EMPLOYEES WITH (DEPT EQUAL "ASTRONOMY") OR (STATUS EQUAL "TRAINEE"), try to get in the habit of using them. You'll find that parentheses help you see the logic of your Boolean expressions and, as you gain experience with Boolean expressions, parentheses will make the difference between getting the right group of records and the wrong or not quite right group of records.

**TRY IT**

The best way to learn to use Boolean operators to form compound Boolean expressions is to dive right in and experiment. Ready the EMPLOYEES domain and try using Boolean expressions to print all experienced employees in the mathematics department, all trainees in the astronomy department, all trainees in the astronomy department who earn less than \$50000, all employees not in the mathematics or astronomy departments, all experienced employees not named Boole in the mathematics department, or any other categories of employees you can think of.

To display all experienced employees in the mathematics department, for example, use an RSE like this:

```
WITH (STATUS EQUAL "EXPERIENCED") AND (DEPT EQUAL "MATHEMATICS")
```

Or, you could display the same group of records with an RSE like this:

```
WITH (STATUS NOT-EQUAL "TRAINEE") AND (DEPT CONT "MATH")
```

To display all experienced mathematicians not named Boole, you could use this RSE:

```
WITH (STATUS CONT "EXP" AND DEPT CONT "MATH") AND  
(L-NAME NOT-EQUAL "BOOLE")
```

As you experiment, create expressions like the one above (with and without parentheses) to see how PRO/DATATRIEVE evaluates complex Boolean expressions. If you take the time to play with RSEs and Boolean expressions, you will be well on your way to becoming a PRO/DATATRIEVE wizard.

**SORTING RECORDS IN A RECORD STREAM**

By adding a SORTED BY clause to an RSE, you tell PRO/DATATRIEVE to sort records in a particular way. You tell PRO/DATATRIEVE how to sort the records by naming a field in a record as a *sort key*. The normal sort key for the EMPLOYEES domain is the primary index key field, ID. To display employees in the math department alphabetically by the LAST-NAME field (query name L-NAME), include a SORTED BY clause:

DTR> PRINT EMPLOYEES WITH DEPT CONT "MATH" SORTED BY L-NAME

ID	STATUS	FIRST NAME	LAST NAME	DEPT	SALARY
00001	EXPERIENCED	GEORGE	BOOLE	MATHEMATICS	\$2,000
03991	TRAINEE	JAMES	BOOLE	MATHEMATICS	\$14,000
73713	TRAINEE	REED	DION	MATHEMATICS	\$23,000
78375	EXPERIENCED	ALBERT	EINSTEIN	MATHEMATICS	\$40,095
00891	EXPERIENCED	FRED	HOWL	MATHEMATICS	\$59,594
83764	EXPERIENCED	JIM	HEADER	MATHEMATICS	\$41,029

PRO/DATATRIEVE chooses the mathematics records to include in the record stream and then sorts the records by L-NAME. If you include a restriction clause, PRO/DATATRIEVE selects the number of records you specify after sorting the records in the record stream.

By separating sort keys with commas, you can sort records by more than one field. PRO/DATATRIEVE sorts records by the first sort key, then by the second sort key, and so on. To display the first five employee records sorted by last name and first name, for example, specify L-NAME, then F-NAME as sort keys:

DTR> PRINT FIRST 5 EMPLOYEES SORTED BY L-NAME, F-NAME

ID	STATUS	FIRST NAME	LAST NAME	DEPT	SALARY
00001	EXPERIENCED	GEORGE	BOOLE	MATHEMATICS	\$2,000
83771	EXPERIENCED	GERALD	BOOLE	ASTRONOMY	\$21,000
03991	TRAINEE	JAMES	BOOLE	MATHEMATICS	\$14,000
93811	TRAINEE	MARTHA	BOOLE	ASTRONOMY	\$20,000
83991	TRAINEE	LESLIE	CARON	GEOGRAPHY	\$21,000

Unless you specify otherwise, PRO/DATATRIEVE sorts records by the ascending value of the sort key, putting the record with the smallest value (A or 0) first and the record with the largest value (Z or the highest number) last. To sort records in descending order, from largest to smallest, specify DESCENDING (or DESC) in the SORTED BY clause, like this:

SORTED BY DESCENDING DEPT, L-NAME

When you change the sort order (ASC or DESC) for one key, PRO/DATATRIEVE uses that order to sort any following keys until you specify otherwise. To sort by descending department and ascending last name, specify ASCENDING (or ASC) before L-NAME, like this:

SORTED BY DESCENDING DEPT, ASC L-NAME

**TRY IT**

Try various combinations of sort keys and sort orders to see how you can make PRO/DATATRIEVE sort records in the EMPLOYEES domain in different ways. You might want to sort records by descending L-NAME, by ascending DEPT and descending L-NAME, by ascending STATUS to see all the experienced records first, or by ascending or descending salary. Remember, once you change the sort order with DESCENDING or ASCENDING, PRO/DATATRIEVE uses that order to sort all keys that follow until you specifically change the order again. Remember, also, that PRO/DATATRIEVE sorts records before limiting the record stream. Thus, the record streams formed with FIRST 5 EMPLOYEES and FIRST 5 EMPLOYEES SORTED BY L-NAME do not look alike.

As you experiment with the the SORTED BY clause and the EMPLOYEES domain, there may be times when you do not want to wait to see all 32 records in the domain displayed. To stop the display and return to the DTR> prompt, press INTERRUPT and DO. EXIT or CTRL/Z returns you to the Main Menu.

When you feel that you have had enough practice forming RSEs, use the FINISH command to end your control over the EMPLOYEES domain and the CUSTOMERS domain.

**SUMMARY**

In this chapter you learned the fundamentals of forming RSEs and creating record streams. You learned how to:

- Restrict the number of records included in a record stream with the FIRST or ALL clause
- Specify conditions for the record stream with Boolean expressions
- Sort record streams with the SORTED BY clause

The remaining chapters of this book show you how to use RSEs and record streams to manipulate information and make data meaningful.

Dear Mr. [Name],

I have received your letter of the 15th and am glad to hear from you. The information you provided is being reviewed and I will get back to you as soon as possible. Thank you for your patience.

I am sorry that I cannot give you a more definite answer at this time, but the process is still ongoing.

I will contact you again once a final decision has been reached.

Very truly yours,

[Signature]

[Title]

[Address]

[City]

[State]

[Zip]

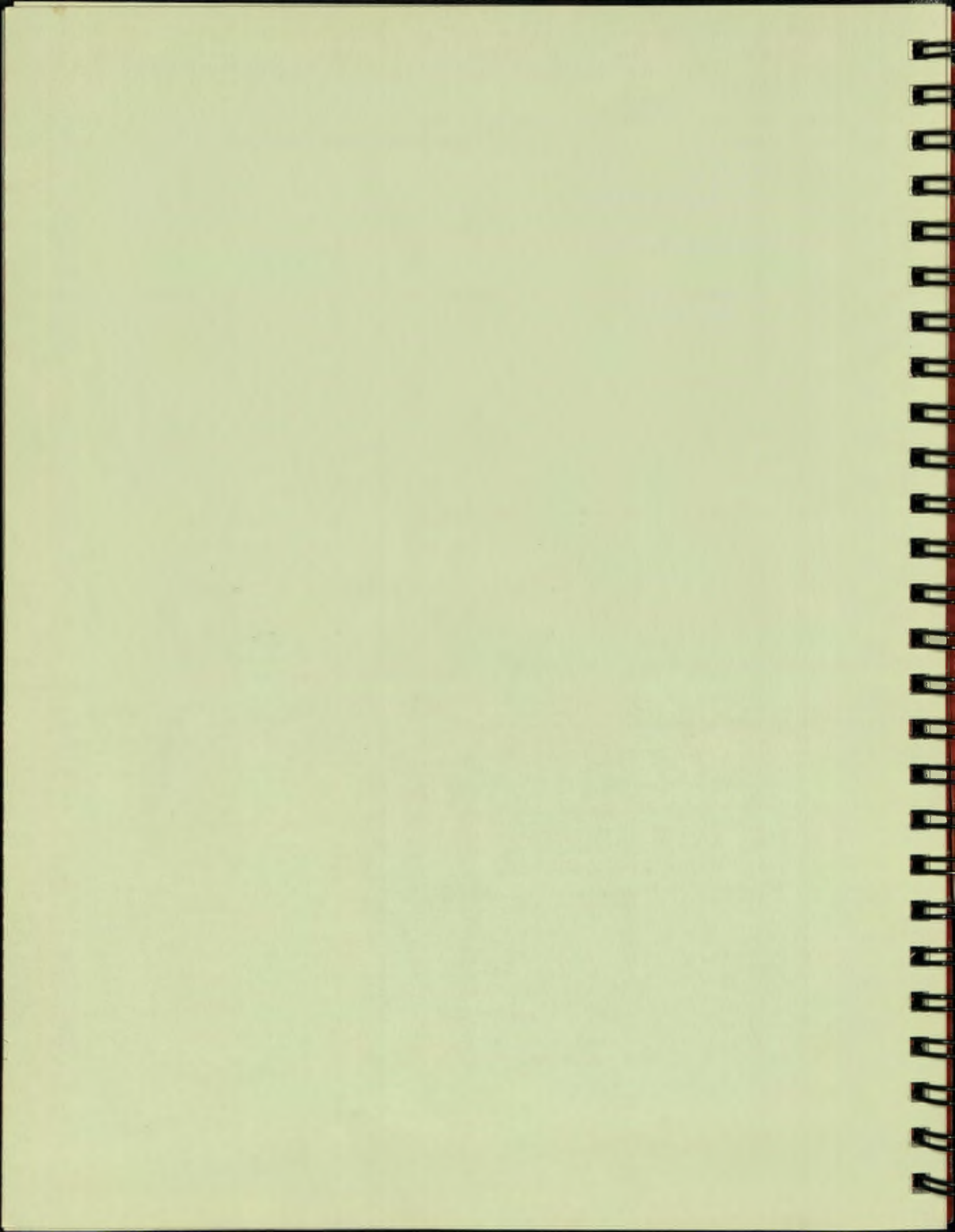
[Phone]

[Fax]

# 5

---

Creating Record  
Collections





## Chapter 5

### Creating Record Collections

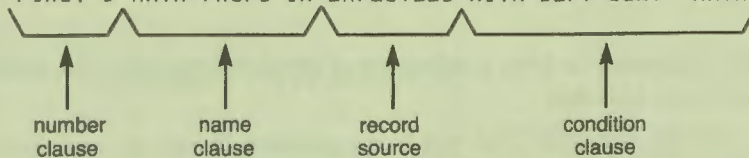
So far, you have worked with all the records in a domain, forming temporary record streams to display on your screen. Unfortunately, each time you want to see a group of records or sort a group of records in a different way, you have to retype the entire PRINT statement.

By using the FIND statement, you can form a special type of record stream called a collection. You can keep a collection of records around to work with until you are finished with it or until you exit from PRO/DATATRIEVE.

#### FORMING AND NAMING COLLECTIONS

To form a collection, type FIND and an RSE. In the RSE, you tell PRO/DATATRIEVE which records to include in the collection and, if you want, how many records to include and what to name the collection. A FIND statement that creates a collection named MATH-PROFS looks something like this:

```
FIND FIRST 5 MATH-PROFS IN EMPLOYEES WITH DEPT CONT "MATH"
```



Each time PRO/DATATRIEVE forms a collection, it assigns the name CURRENT to the collection. If you specify a name, like MATH-PROFS, PRO/DATATRIEVE locates the records you want, names the collection MATH-

PROFS *and* CURRENT, displays a message telling you how many records it found, and displays the DTR> prompt. If you omit the name clause in the RSE, a collection has only one name: CURRENT. If you include the name clause, a collection has two names.

By naming collections, you can keep old collections around until you no longer need them. A collection with only the name CURRENT is not as permanent as a collection with a name you specify, because PRO/DATATRIEVE never keeps more than one collection named CURRENT around to clutter up your computer workspace. If a collection's only name is CURRENT, the collection disappears when you use a FIND statement to form a new collection. If a collection has another name, in addition to CURRENT, it loses the name CURRENT when you form a new collection, but it does not disappear.

In the last chapter you learned to form record streams from domains. You can also form record streams from collections by using the collection name in the RSE, like this:

```
DTR> FIND EMPLOYEES WITH (STATUS EQUAL "EXPERIENCED")
[20 records found]
DTR> FIND CURRENT WITH (DEPT CONT "MATH")
[4 records found]
DTR> PRINT CURRENT
```

ID	STATUS	FIRST NAME	LAST NAME	DEPT	SALARY
00001	EXPERIENCED	GEORGE	BOOLE	MATHEMATICS	\$2,000
00891	EXPERIENCED	FRED	HOWL	MATHEMATICS	\$59,594
78375	EXPERIENCED	ALBERT	EINSTEIN	MATHEMATICS	\$40,095
83764	EXPERIENCED	JIM	MEADER	MATHEMATICS	\$41,029

By using a series of FIND statements, you can refine a collection until it contains just the records you want.

### **TRY IT**

Use the FIND statement to form a collection of employee records that contains only mathematicians, like this:

```
DTR> FIND EMPLOYEES WITH (DEPT CONT "MATH")
[6 records found]
```

Because you did not specify a name for the collection, PRO/DATATRIEVE gives it only the name CURRENT. To see that the collection exists for you to work with and is named CURRENT, type SHOW COLLECTIONS:

```
DTR> SHOW COLLECTIONS
Collections:
    CURRENT
```

To display the collection, type PRINT ALL, PRINT CURRENT, or PRINT ALL CURRENT. If you just type PRINT and press DO PRO/DATATRIEVE displays the message "No record selected, printing whole collection" and displays all records in the CURRENT collection.

When you type PRINT and then press DO, PRO/DATATRIEVE first looks to see if you have chosen a particular record earlier with the SELECT statement. You will learn how to use the SELECT statement in the next chapter. For now, all you need to know is that the PRINT statement displays a selected record if one exists and the entire CURRENT collection if no selected record exists.

You can get information about a collection by typing SHOW and the collection name. In this case, you need to type SHOW CURRENT. PRO/DATATRIEVE displays the name of the domain from which the collection was formed, the number of records in the collection, and tells you there is no selected record.

Now form another collection containing only astronomy records and name the collection ASTRONOMERS, like this:

```
DTR> FIND ASTRONOMERS IN EMPLOYEES WITH DEPT CONT "ASTR"
[5 records found]
```

To see that you have formed a collection with two names, ASTRONOMERS and CURRENT, and that the old CURRENT collection has disappeared, type SHOW COLLECTIONS again:

```
DTR> SHOW COLLECTIONS
Collections:
    ASTRONOMERS (also CURRENT)
```

If you want to see information about the ASTRONOMERS collection, you can type either SHOW ASTRONOMERS or SHOW CURRENT. Likewise, you can refer to the collection by either name in the PRINT statement.

To see what happens when you form a new collection, form another collection of mathematics department records and name it MATH-PROFS. Then type SHOW COLLECTIONS again:

```
DTR> FIND MATH-PROFS IN EMPLOYEES WITH DEPT CONT "MATH"
[8 records found]
DTR> SHOW COLLECTIONS
Collections:
    MATH-PROFS (also CURRENT)
    ASTRONOMERS
```

Note that the MATH-PROFS collection appears above the ASTRONOMERS collection, as the CURRENT collection. The only way you can now refer to the ASTRONOMERS collection is by using the name ASTRONOMERS.

Use PRINT statements to display records in your collections. When you have more than one collection, PRINT ALL without a collection name always displays the records in the CURRENT collection. To display all the records in a named collection, specify the collection name in the PRINT statement.

Experiment with forming new collections from existing collections by using an existing collection name as the record source in the RSE. You might, for example, form a collection of experienced astronomy records from the ASTRONOMERS collection, like this:

```
DTR> FIND EXP-ASTR IN ASTRONOMERS WITH STATUS CONT "EXP"
[3 records found]
```

You can replace a named collection anytime by specifying the old collection name as the new collection name. If you had specified ASTRONOMERS instead of EXP-ASTR in the previous example, PRO/DATATRIEVE would have created a new ASTRONOMERS collection from the old ASTRONOMERS collection.

Use the FIND statement to form a collection of employee records with SALARY greater than \$30,000. Then use a series of FIND statements to form new, smaller CURRENT collections from previous CURRENT collections by using new limiting values for SALARY. To form a collection from a CURRENT collection, substitute the name CURRENT for the domain name in the FIND statement, like this:

```
DTR> FIND EMPLOYEES WITH SALARY GT 30000
[19 records found]
DTR> FIND CURRENT WITH SALARY GT 40000
[12 records found]
DTR> FIND CURRENT WITH SALARY GT 50000
[8 records found]
```

Form several more named and unnamed collections, stopping now and then to display information about your collections by typing `SHOW COLLECTIONS` and information about specific collections by typing `SHOW` and the collection name. Notice that the `CURRENT` collection is always the most recently formed collection.

Note also that a named collection does not disappear when you form a new `CURRENT` collection, but that an unnamed `CURRENT` collection disappears when you form a new collection. By naming collections, you can ensure that they do not disappear when you form a new collection. Collections do, though, disappear when you exit from `PRO/DATATRIEVE`.

### DELETING COLLECTIONS WITH THE RELEASE COMMAND

Collections occupy space in memory, so it is a good idea to get rid of a collection when you no longer need it. As you have seen, the `CURRENT` collection disappears every time you form a new collection. You can get rid of named collections with the `RELEASE` command. If you just type `RELEASE` and press `DO`, `PRO/DATATRIEVE` prompts you for a collection name. To release more than one collection, separate collection names with commas. If the collection you release is also the `CURRENT` collection, `PRO/DATATRIEVE` assigns the name `CURRENT` to the collection that was the previous `CURRENT` collection.

### TRY IT

Type `SHOW COLLECTIONS` to see what collections you have already established and which collection is the `CURRENT` collection. If the `CURRENT` collection is also a named collection, you can release it by typing either `RELEASE CURRENT` or `RELEASE` and the collection name.

After you type a `RELEASE` command, use the `SHOW COLLECTIONS` command to see how `PRO/DATATRIEVE` reassigns the name `CURRENT` to the previous `CURRENT` collection. Such a sequence of statements and commands might look like this:

```
DTR> FIND MATH IN EMPLOYEES WITH (DEPT CONT "MATH")
[8 records found]
DTR> FIND LIT IN EMPLOYEES WITH (DEPT CONT "LIT")
[7 records found]
DTR> FIND BIOL IN EMPLOYEES WITH (DEPT CONT "BIOL")
[4 records found]
DTR> FIND ASTRON IN EMPLOYEES WITH (DEPT CONT "ASTR")
[5 records found]
```

(continued on next page)

```

DTR> SHOW COLLECTIONS
Collections:
    ASTRON (also CURRENT)
    BIOL
    LIT
    MATH
DTR> RELEASE ASTRON
DTR> SHOW COLLECTIONS
Collections:
    BIOL (also CURRENT)
    LIT
    MATH
DTR> RELEASE BIOL, LIT
DTR> SHOW COLLECTIONS
Collections:
    MATH (also CURRENT)
DTR> RELEASE CURRENT
DTR> SHOW COLLECTIONS
No established collections

```

## SORTING COLLECTIONS

You can sort a collection when you form it by including a SORTED BY clause in the FIND statement. PRO/DATATRIEVE locates the records you want in the collection, then arranges those records in the order you specified in the sort key. If you do not use a SORTED BY clause when you form a collection, PRO/DATATRIEVE arranges records by the index key.

You can also sort an existing collection by using the collection name as the record source and including a SORTED BY clause in an RSE:

```

DTR> FIND EMPLOYEES WITH (DEPT CONT "BIOL")
[4 records found]
DTR> PRINT CURRENT SORTED BY L-NAME

```

ID	STATUS	FIRST NAME	LAST NAME	DEPT	SALARY
90342	EXPERIENCED	BRUNO	DONCHIKOV	BIOLOGY	\$35,952
87465	EXPERIENCED	ANTHONY	IACOBONE	BIOLOGY	\$58,462
99029	EXPERIENCED	RANDY	PODERESIAN	BIOLOGY	\$33,738
49001	EXPERIENCED	DAN	ROBERTS	BIOLOGY	\$41,395

As with temporary record streams, PRO/DATATRIEVE sorts records in the collection only temporarily, to display them. The order of records in the collection does not change.

To rearrange records in the collection itself, you need to use the SORT statement. This statement works only for collections. That is, you cannot use the SORT statement to rearrange the order of records in a data file. To sort the CURRENT collection, you tell PRO/DATATRIEVE to SORT CURRENT BY L-NAME, or whatever sort key you want. To sort a named collection, you must include the collection name in the SORT statement, like this:

```
DTR> SORT MATH-PROFS BY L-NAME
```

As with the SORTED BY clause, you can specify more than one field as a sort key, up to a maximum of nine fields.

### TRY IT

Form two named collections, ASTRONOMERS and BIOLOGISTS, from the EMPLOYEES domain. Form the BIOLOGISTS collection after the ASTRONOMERS so that it is also the CURRENT collection. Display the ASTRONOMERS collection sorted by L-NAME, then type PRINT ASTRONOMERS to see that the original order of the collection has not changed.

Now type SORT BIOLOGISTS BY L-NAME to change the order of the BIOLOGISTS collection (also the CURRENT collection). If you then type PRINT BIOLOGISTS or PRINT CURRENT, you see that the record order has changed.

If you type SHOW ASTRONOMERS and SHOW BIOLOGISTS to see information about the collections, you see that PRO/DATATRIEVE provides you with sort information for collections that you have sorted:

```
DTR> SORT BIOLOGISTS BY L-NAME
DTR> SHOW ASTRONOMERS
Collection ASTRONOMERS
  Domain:                EMPLOYEES
  Number of records: 10
  No selected record
DTR> SHOW BIOLOGISTS
Collection BIOLOGISTS
  Domain:                EMPLOYEES
  Number of records: 10
  No selected record
  Sort order: LAST_NAME
```

Practice using the SORT statement and the SORTED BY clause with the CURRENT collection and with named collections.

**DELETING COLLECTIONS WITH THE FINISH COMMAND**

In Chapter 3, you learned to use the FINISH command to end your access to domains. You cannot specify a collection in the FINISH command. However, when you use a FINISH command to end access to a domain, PRO/DATATRIEVE releases all collections formed from that domain.

If you type just FINISH, PRO/DATATRIEVE finishes all ready domains and collections. If you specify a domain name in the FINISH command, PRO/DATATRIEVE finishes that domain and the collections formed from it. Other domains and collections remain ready.

**TRY IT**

To see how FINISH works, ready the CUSTOMERS domain and create a collection from it. Make sure you still have one or two collections from the EMPLOYEES domain around and that the EMPLOYEES domain is still ready by using the SHOW READY and SHOW COLLECTIONS commands. Then finish the CUSTOMERS domain by typing FINISH CUSTOMERS and use SHOW commands again to see that the FINISH command did not affect the EMPLOYEES domain or collections. Your sequence of commands should look something like this:

```
DTR> SHOW READY
Ready domains:
    CUSTOMERS: RMS INDEXED, PROTECTED READ
    EMPLOYEES: RMS INDEXED, PROTECTED READ
DTR> SHOW COLLECTIONS
Collections:
    CUSTOMER-INFO (also CURRENT)
    EMPLOYEE-INFO
DTR> FINISH CUSTOMERS
DTR> SHOW READY
Ready domains:
    EMPLOYEES: RMS INDEXED, PROTECTED READ
DTR> SHOW COLLECTIONS
Collections:
    EMPLOYEE-INFO
```



Notice that PRO/DATATRIEVE does not reassign the name CURRENT to the remaining collection, making FINISH different from RELEASE:

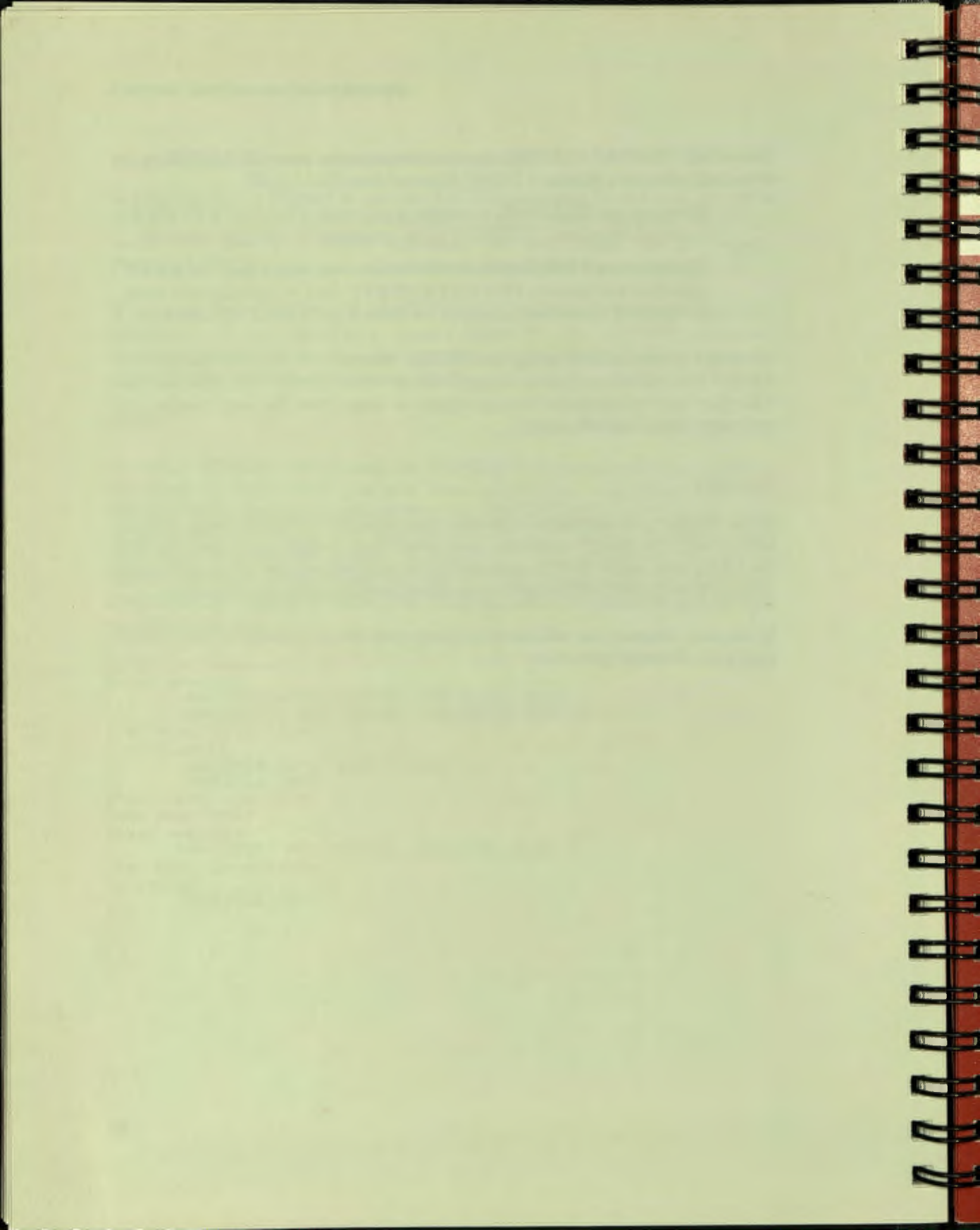
- When you use RELEASE to release a collection, PRO/DATATRIEVE reassigns the name CURRENT to the previous CURRENT collection.
- When you use FINISH to finish the domain from which the CURRENT collection was formed, PRO/DATATRIEVE does not reassign the name CURRENT to another collection, so there is no CURRENT collection.

Try to get in the habit of using the FINISH command to end your control over domains and collections you no longer need or want to work with. Domains and collections use up computer storage space, so learn from the start to use your computer wisely and efficiently.

### SUMMARY

In this chapter you learned to form collections with the FIND statement, sort collections with the SORT statement, and to end your control over collections with the RELEASE and FINISH commands. You also learned how to use new forms of the SHOW COMMAND to get information about domains and collections.

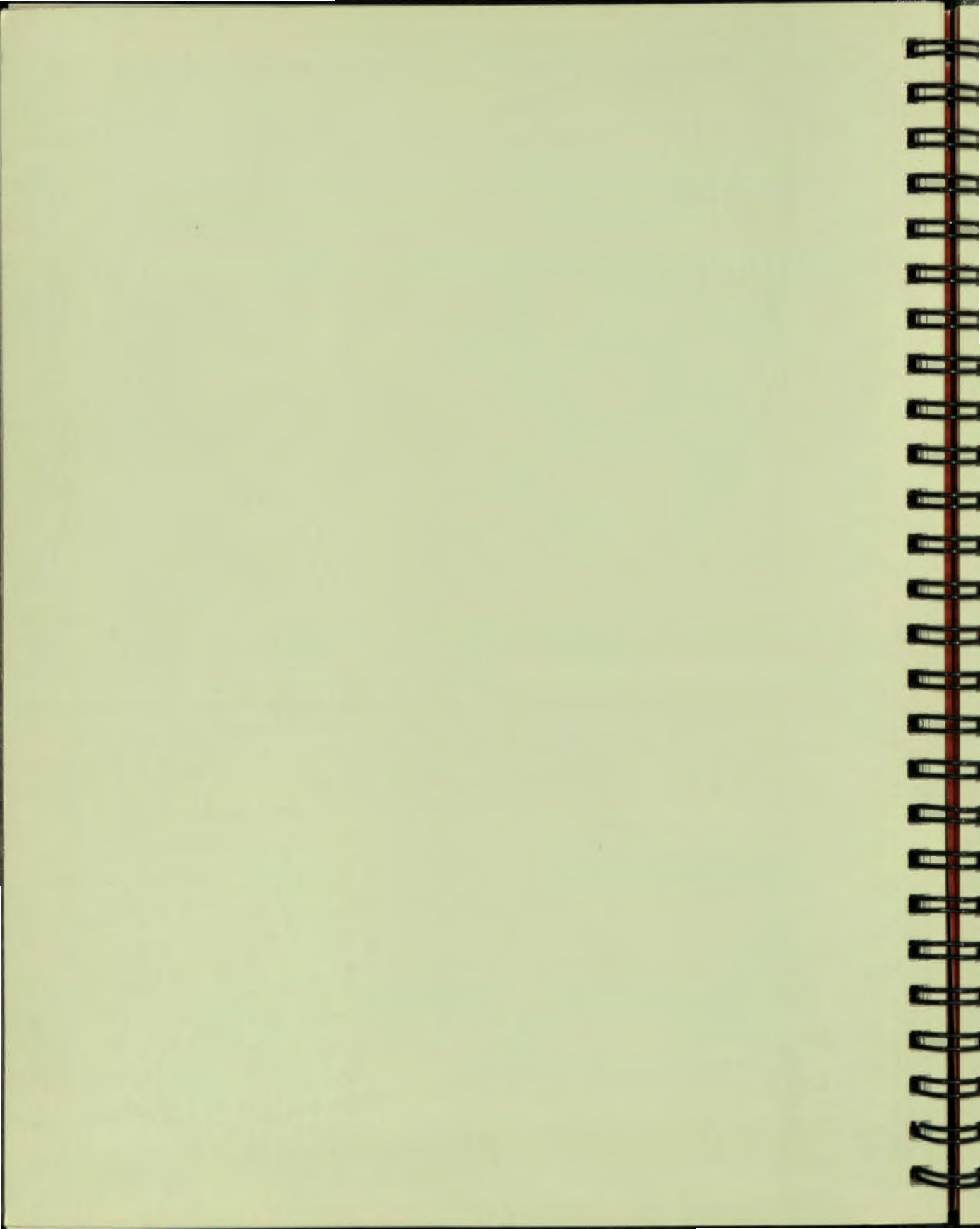
In the next chapter you will learn to change and delete records so that you can keep your database up to date.



# 6

---

## Changing and Deleting Information



## Chapter 6

---

### Changing and Deleting Information

Information changes all the time. When information changes, you need to modify records to keep your database up-to-date and useful to you. If you accidentally store incorrect information, you also need to modify records. Sometimes information changes so dramatically that you need to delete or erase records.

You change or delete PRO/DATATRIEVE records by using the MODIFY or ERASE statement. You can use the SELECT statement to identify a record in a collection as the target for a MODIFY or ERASE statement, or you can identify a group of records with an RSE.

This chapter shows you how to use the SELECT, MODIFY, and ERASE statements to change or delete records. When you modify or erase records in a collection, you change those records in the data file associated with the domain from which the collection was formed. In this respect, MODIFY and ERASE work differently than statements such as PRINT that only change the appearance.

Remember that you can erase records only from indexed files. Because PRO/DATATRIEVE finds records in a sequential file as they physically occur in the file, you cannot physically delete records. You can use the MODIFY statement to delete all data from a record in a sequential file, but you cannot delete the physical record. Because empty records continue to take up space on your hard disk or diskette, you should use indexed files.

## SELECTING RECORDS FROM A COLLECTION

One of the easiest ways to modify or erase one or a few records is to form a collection that contains the record or records you want to change. Then you can use the **SELECT** statement to choose a record as the target for a **MODIFY** or **ERASE** statement.

You can select records only from a collection.

Once you have formed a collection, it's a good idea to display the records in the collection with the **PRINT ALL** statement so that you can identify the record you want to select. When you have not yet selected a record from the collection, typing **SELECT** and pressing **DO** tells **PRO/DATATRIEVE** to select the first record in the collection. To choose the next record in the collection, type **SELECT** or **SELECT NEXT**. Each time you type **SELECT** again or **SELECT NEXT**, **PRO/DATATRIEVE** marks the next record in the collection as the selected record.

As shown in Table 6-1, you can choose any single record in the target collection by using the different forms of the **SELECT** statement.

**Table 6-1: SELECT Statement Forms and Effects**

<i>SELECT Statement</i>	<i>Effect</i>
<b>SELECT number</b>	Selects the record you identify by its place in the collection. <b>SELECT 5</b> , for example, selects the fifth record in the collection, not five records. If you specify a number greater than the number of records in the collection, <b>PRO/DATATRIEVE</b> displays the message "Record number out of range for collection."
<b>SELECT FIRST</b>	Selects the first record in the collection.
<b>SELECT LAST</b>	Selects the last record in the collection.
<b>SELECT</b> or <b>SELECT NEXT</b>	Selects the next record in the collection. If you have not yet selected a record in the collection, <b>SELECT</b> picks the first record. If the last record in the collection is the current selected record, <b>PRO/DATATRIEVE</b> displays the message "Record number out of range for collection" when you type <b>SELECT</b> . To go back to a previous record, type <b>SELECT number</b> .

If you know that you want to choose the third or fourth record, you can type `SELECT 3` or `SELECT 4`. To select a record that precedes the currently selected record, type `SELECT` and the number of the record. To select the first record at any time, type `SELECT FIRST`, and to choose the last record in the collection, type `SELECT LAST`.

To see the selected record, type `PRINT`. In the previous chapter you learned to use `PRINT` to display all records in the `CURRENT` collection. If you have a selected record, `PRINT` displays the selected record instead of the entire collection.

The `SELECT` statement always chooses a record from the `CURRENT` collection unless you specify a collection name. To tell `PRO/DATATRIEVE` to select a record in a collection other than the `CURRENT` collection, specify the collection name in the `SELECT` statement, like this:

- `SELECT FIRST MATH-PROFS` selects the first record in the `MATH-PROFS` collection.
- `SELECT 5 BIO-PROFS` selects the fifth record in the `BIO-PROFS` collection.
- `SELECT LAST MATH-PROFS` selects the last record in the `MATH-PROFS` collection.

Each collection you have available can have a selected record. When you type `PRINT`, `MODIFY`, or `ERASE` and press `DO`, though, `PRO/DATATRIEVE` operates on the selected record in the `CURRENT` collection. To make sure that the selected record is the one you want to modify or erase, always type `PRINT` and press `DO` before typing a `MODIFY` or `ERASE` statement.

### **TRY IT**

Ready the `CUSTOMERS` domain and form a collection named `NH` that contains records for customers with `STATE` equal to `NH`. Then use a `SELECT` statement to select the first record:

```
DTR> READY CUSTOMERS
DTR> FIND NH IN CUSTOMERS WITH STATE EQUAL "NH"
[5 records found]
DTR> SELECT
DTR>
```

You can now type `PRINT` and press `DO` to see the selected record or you can type `PRINT` and the names of the fields you want to see. If you type `SHOW NH`, you see that `PRO/DATATRIEVE` tells you the first record in the collection is the selected record.

Type `SELECT` to select the second record in the collection. If you type `SHOW NH` again, you see that the selected record is now the second record in the collection. Experiment with `SELECT` statements, including `SELECT LAST`, to see how `PRO/DATATRIEVE` selects records. Type `SELECT` after `SELECT LAST` to see the message `PRO/DATATRIEVE` displays.

Now form a collection named `MA` of customers with `STATE` equal to `MA`. `MA` is now the `CURRENT` collection. If you type `SELECT`, `PRO/DATATRIEVE` selects the first record in `MA`. When you type `PRINT`, you see the selected record from this collection. The last record you selected from the `NH` collection remains the selected record for that collection, but the `PRINT` statement displays the selected record from the `CURRENT` collection.

You can select another record in the `NH` collection by specifying the collection name in the `SELECT` statement. Then, to make that record the target of a `MODIFY` or `ERASE` statement, release the `CURRENT` collection with the `RELEASE` command, then type `PRINT` to see that the selected record is now from the `NH` collection.

If you want, form another collection of customer records and experiment with `SELECT` and `PRINT` statements to see how `PRO/DATATRIEVE` selects records from the `CURRENT` collection and from other collections. Use the `SHOW` command to keep track of the selected record in each collection.

If you want to see all the records in the `CURRENT` collection, type `PRINT ALL`. To see all the records in any other collection, type `PRINT` or `PRINT ALL` and the collection name.

When you finish experimenting with the `SELECT` statement, type `RELEASE` to release all collections.

### **USING COLLECTIONS TO MODIFY AND ERASE RECORDS**

The following sections show how to use the `MODIFY` statement to change records in a collection and the `ERASE` statement to delete records from a collection. Remember, when you modify or erase records in a collection, `PRO/DATATRIEVE` modifies or erases the record in the data file associated with



the domain, not just in the collection. If you make a mistake, you have to make the change again or store an erased record again. The modified record replaces the original record in the data file and an erased record is lost forever. So, use **MODIFY** with care and **ERASE** with special care.

You can use **MODIFY** to change all fields or specific fields of a selected record or collection. You can use **ERASE** to delete a selected record or all records in a collection. Remember that you can only erase records in an indexed data file.

### **Modifying the Selected Record**

To change information in a selected record, you need to take the following steps:

1. Get the domain ready for **WRITE** or **MODIFY** access.
2. Form a collection containing the records you want to modify.
3. Use the **SELECT** statement to identify the record you want to modify.
4. Display the selected record with the **PRINT** statement to make sure it is the one you want to modify.
5. Use the **MODIFY** statement to change the selected record.
6. Display the changed record with the **PRINT** statement.

To modify all the fields of a selected record, type **MODIFY** and press **DO**. **PRO/DATATRIEVE** then prompts you for all fields, just as it does when you use the **STORE** statement.

Remember that you cannot change the index key field. To change the company name of a record, you have to erase the entire record and store it again as a new record with the changed name.

In the **CUSTOMERS** domain, for example, you cannot change the **COMPANY-NAME** field. You could type in the same company name in response to the prompt, but to save typing time, you can press the **TAB** key and press **DO**. The **TAB** key tells **PRO/DATATRIEVE** not to change the value currently stored in the field. You can use the **TAB** key to retain the value of any field in the record.

If you want to leave a field blank, press the space bar, just as you did when you stored records in the **CUSTOMERS** domain. If you do not provide a new value of space and do not press the **TAB** key before pressing **DO**, **PRO/DATATRIEVE** prompts you again.

If you decide you do not want to modify the record after you typed **MODIFY**, you press the **EXIT** key in response to any prompt. Pressing **EXIT** returns you to the **DTR>** prompt.

### **TRY IT**

When you stored records in the **CUSTOMERS** domain, you left the **ZIP-CODE** field blank in two records. To get ready to store ZIP codes in these empty fields, ready the **CUSTOMERS** domain for **WRITE** or **MODIFY** access and form a collection of records with no ZIP code, like this:

```
DTR> READY CUSTOMERS MODIFY
DTR> FIND NO-ZIPS IN CUSTOMERS WITH (ZIP EQUAL " ")
[2 records found]
```

The space between the quotation marks tells **PRO/DATATRIEVE** to include only records with no ZIP codes in the collection. If **ZIP-CODE** were a numeric field, you could type zero instead of empty quotation marks.

Now use the **SELECT** statement to select the first record, display the record using the **PRINT** statement, and type **MODIFY**. If you forgot to ready the domain for **MODIFY** access, **PRO/DATATRIEVE** displays an error message. Otherwise, **PRO/DATATRIEVE** prompts you for new information for each field. Use the **TAB** key to retain the values currently stored in each field until **PRO/DATATRIEVE** prompts you for the ZIP code. Type in a 5-digit number and press **DO**. If you want, type in a new phone number and change the **SALES-THIS-YEAR** and **LAST-CONTACT** fields in response to the next two prompts. After you respond to the last prompt, type **PRINT** to display the modified record.

The modified record is still the selected record, so if you want to change it again, just type **MODIFY** again. To store a value in the ZIP code field for the second record in the collection, type **SELECT** or **SELECT NEXT** to select the next record.

You could go through the **PRINT**, **MODIFY**, and **PRINT** process again for each record you want to change. Since you want to modify only one field, though, you can save time by specifying the name of the field you want to change.

**MODIFY ZIP**, for example, specifies that you want to modify only the **ZIP-CODE** field of the selected record, so **PRO/DATATRIEVE** prompts only for that field. This saves you from having to use the **TAB** key to pass through fields you do

not want to change. If you want to modify more than one field, separate field names with commas. `MODIFY ZIP, PHONE`, for example, tells `PRO/DATATRIEVE` to prompt for `ZIP-CODE` and `PHONE`.

Form a collection of current customer records, select a record, display it, then type `MODIFY COMPANY, SALES` to change the company name and sales value. `PRO/DATATRIEVE` prompts you for new values for the two fields, then displays a message telling you that you cannot modify the primary key field.

If you want, use the `STORE` statement to store new records in the `EMPLOYEES` or `CUSTOMERS` domain, then use the `MODIFY` statement to change the new information. Note that if you specified `MODIFY` access to the `EMPLOYEES` domain, `PRO/DATATRIEVE` will not let you store new records and tells you that the domain is not properly readied for storing records. To store new records, you must ready a domain for `WRITE` access. `MODIFY` access gives you the right to change existing records but not to store new records. `WRITE` access gives you the the right to change existing records, store new records, and erase records.

#### **Erasing the Selected Record**

To erase a selected record from a collection, you need to take the following steps:

1. Ready the domain for `WRITE` access.
2. Form a collection containing the record you want to erase.
3. Use the `SELECT` statement to select the record.
4. Type `ERASE` and press `DO`.

If your collection contains only one record and you erase that record, the record disappears but the collection does not. `PRO/DATATRIEVE` erases the record from the collection and the data file, but the record's slot in the collection remains. The empty slot disappears when you use the `RELEASE` command to release the collection or the `FINISH` command to finish the domain from which you formed the collection.

#### **TRY IT**

Ready the `CUSTOMERS` domain for `WRITE` access and use the `REPEAT` and `STORE` statements to store two records. Remember that you have to specify the number of times you want the `REPEAT` statement to execute, so type in `REPEAT 2 STORE CUSTOMERS`.

Then, store two records, form a collection that contains these records, select a record, and erase it. Select the next record and erase it. Then type `PRINT CURRENT` to see that the collection remains even though you have deleted the records.

If you stored an incorrect record in the `CUSTOMERS` domain earlier, isolate the record in a collection, select the record, and erase it.

### **Modifying All Records in a Collection**

You can modify all records in a collection with the `MODIFY ALL` statement. With `MODIFY ALL`, you can, for example, change the `DEPT` value for all literature department records from `LITERATURE` to `AMERICAN LIT`. To do this, gather all the literature department records into a collection, display the collection with the `PRINT CURRENT` statement to make sure it contains the records you want to change, then type `MODIFY ALL DEPT. PRO/DATATRIEVE` prompts for the new `DEPT` value and stores that value in the `DEPT` field of all records in the collection.

Be sure to use the `PRINT CURRENT` statement to look at all the records in a collection before using the `MODIFY ALL` statement. If the collection contains records you do not want to change, you may have to spend a lot of time replacing data you changed by mistake. Do not be afraid of using `MODIFY ALL` where appropriate, but do use it carefully.

### **TRY IT**

Ready the `EMPLOYEES` domain for `WRITE` or `MODIFY` access and form a collection of literature department records. Display the collection to make sure you have the right records, then use the `MODIFY ALL` statement to change the `DEPT` field to `AMERICAN LIT`.

Display the records again to see that the change has been made, then type another `MODIFY ALL` statement to change the `DEPT` field back to `LITERATURE`.

### **Erasing All Records in a Collection**

To erase all records in a collection, establish the collection as the `CURRENT` collection and type `ERASE ALL`. You cannot recover deleted records, so type `PRINT ALL` to see the records `PRO/DATATRIEVE` will delete before you type `ERASE ALL`.

**TRY IT**

Use REPEAT and STORE to store two fictitious records in the CUSTOMERS domain, then form a collection that contains these records. Use the ERASE ALL statement to erase both records in the collection and the PRINT statement to verify that PRO/DATATRIEVE has deleted both records. If you like, try to print the erased records from the domain to see that PRO/DATATRIEVE has erased them from the data file associated with the domain.

**USING RECORD STREAMS TO MODIFY AND ERASE RECORDS**

You can modify and erase records without forming a collection and selecting individual records by using an RSE to create a record stream containing the records you want to change or erase and by putting the keywords ALL OF in front of the record source, like this:

```
DTR> ERASE ALL OF EMPLOYEES WITH (ID CONT 9999)
```

This statement erases all records as specified in the RSE. PRO/DATATRIEVE requires the keywords ALL OF in an ERASE statement that deletes a record stream. As with collections, pay careful attention to the content of the record stream when you use the ERASE ALL OF statement.

Likewise, use the MODIFY ALL OF statement carefully. Even though PRO/DATATRIEVE does not require the keywords ALL OF in the MODIFY statement, when you type MODIFY and an RSE, you modify an entire record stream, changing the field values of every record in the stream to the same values.

If you do not specify a field or fields in the MODIFY statement, PRO/DATATRIEVE assumes you want to modify all fields. A statement like MODIFY EMPLOYEES WITH SALARY GT 40000 tells PRO/DATATRIEVE to prompt once for new values for each field in the records in the record stream. If you provide new values for all the fields, all records in the stream will be identical.

To modify one field of a record stream, specify the field name first, then the RSE that identifies the records, like this:

```
DTR> MODIFY DEPT OF EMPLOYEES WITH (DEPT CONT "LIT")
```

PRO/DATATRIEVE prompts you for a new DEPT value and stores that value in all records in the record stream. If you want to modify two or more fields of a record stream, separate the field names with commas.

The preceding example does the same thing you did earlier when you used MODIFY ALL to change the DEPT value for all records in a collection. As with MODIFY ALL, be careful when modifying records in a record stream. By forming limited record streams and specifying only those fields you want to modify, you can save yourself many problems.

Always display the records you intend to modify or erase before you use the MODIFY or ERASE statement. If the displayed records are the ones you want, use the same RSE in the MODIFY or ERASE statement. For example:

```
DTR> PRINT FIRST 2 EMPLOYEES WITH (DEPT CONT "MATH")
```

ID	STATUS	FIRST NAME	LAST NAME	DEPT	SALARY
00001	EXPERIENCED	GEORGE	BOOLE	MATHEMATICS	\$2,000
00891	EXPERIENCED	FRED	HOWL	MATHEMATICS	\$59,594

```
DTR> MODIFY DEPT OF FIRST 2 EMPLOYEES WITH (DEPT CONT "MATH")
```

```
Enter DEPT: LOGIC
```

```
DTR> PRINT EMPLOYEES WITH (DEPT EQUAL "LOGIC")
```

ID	STATUS	FIRST NAME	LAST NAME	DEPT	SALARY
00001	EXPERIENCED	GEORGE	BOOLE	LOGIC	\$2,000
00891	EXPERIENCED	FRED	HOWL	LOGIC	\$59,594

If you display a record stream and decide it contains more records than you want to change or erase, you can limit the RSE in the MODIFY or ERASE statement. You could, for example, display the first two mathematicians as above, and decide that you want to change only the second record. To do this, you could use an RSE like WITH (ID EQUAL 891) in the MODIFY statement.

### TRY IT

Ready the EMPLOYEES domain for WRITE or MODIFY access and use the MODIFY statement to change the DEPT field in the literature department records. First use limited RSEs to change records for CASSIDY, SWAY, and WINSLOW to AMERICAN LIT, records for STONE and FREIBURG to COMPARATIVE LIT, and records for TERRY and TASHKENT to WORLD LIT.

To change the CASSIDY record, you might use a MODIFY statement like this:

```
DTR> MODIFY DEPT OF EMPLOYEES WITH (DEPT CONT "LIT") AND
[Looking for Boolean expression]
CON> (L-NAME EQUAL "CASSIDY")
Enter DEPT: AMERICAN LIT
```

Or, you can change all three records (CASSIDY, SWAY, and WINSLOW) with one MODIFY statement, like this:

```
DTR> MODIFY DEPT OF EMPLOYEES WITH (DEPT CONT "LIT") AND
[Looking for Boolean expression]
CON> (L-NAME EQUAL "CASSIDY", "SWAY", "WINSLOW")
Enter DEPT: AMERICAN LIT
```

You can change the TERRY and TASHKENT records this way, too, or you can use the FIRST 2 restriction clause in the RSE. The important thing to remember is to use an explicit RSE to identify the record or records you want to change.

When you finish modifying the literature department records, use one MODIFY statement to change the DEPT of all the records back to LITERATURE. When you finish, the seven literature department records should look like they did when you started this section.

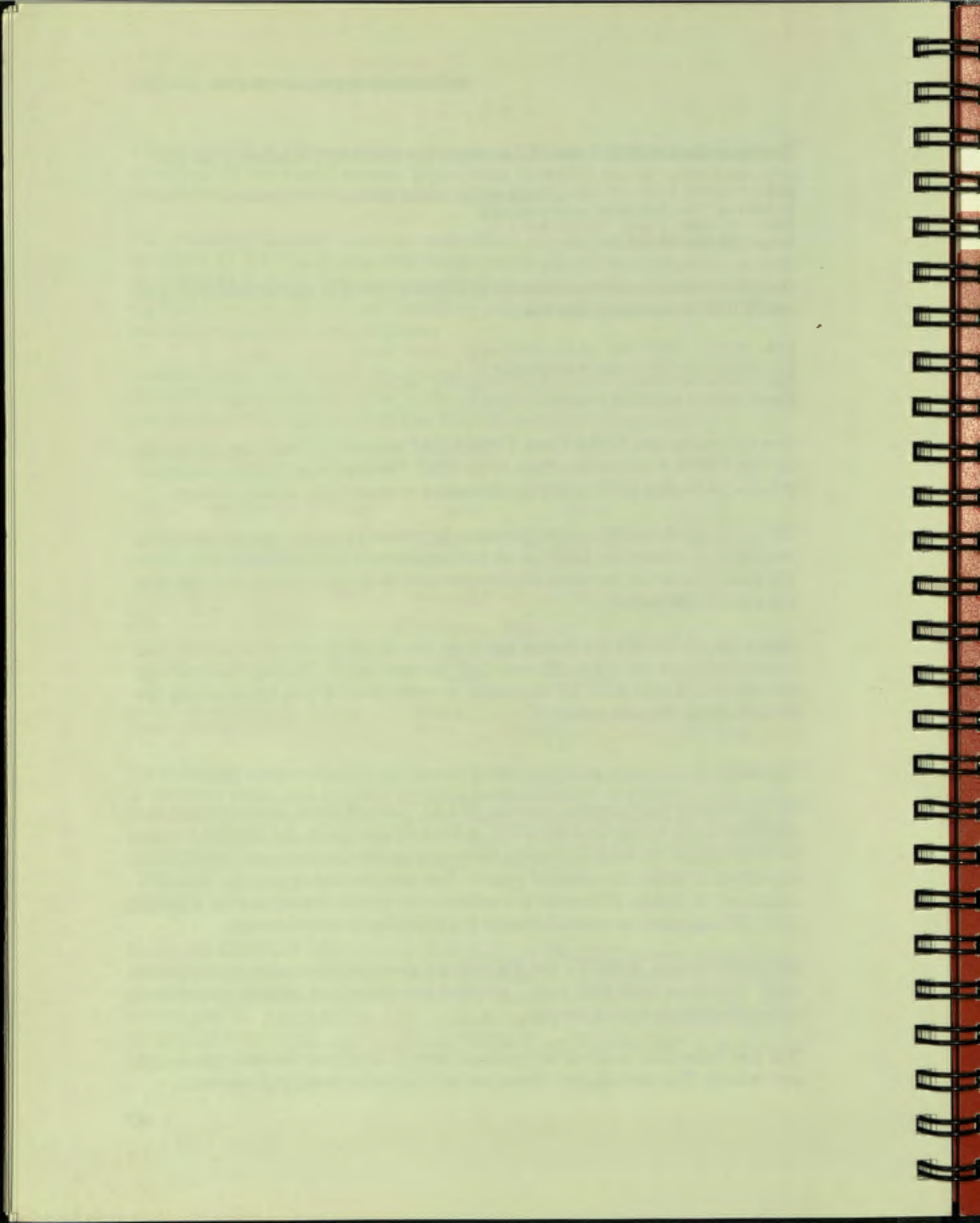
Ready the CUSTOMERS domain and store two fictitious records using IDs that contain 9999 and one other different digit for each record. Display these records and use an ERASE ALL OF statement to erase them. If you try to display the records again, they are gone.

## SUMMARY

In this chapter you learned to use the SELECT statement to select a record in a collection as the target for a MODIFY or ERASE statement, the MODIFY statement to change all fields or specific fields in a selected record, and the ERASE statement to erase the selected record. You also learned to use the MODIFY statement to change all records in a collection or record stream and the ERASE ALL OF statement to erase all records in a collection or record stream.

As you have seen, MODIFY and ERASE are essential information management tools. Use these tools with care — modified records replace original records and erased records disappear forever.

You now know how to use all the tools you need to keep your database up-to-date and orderly. The next chapter shows you how to create meaningful reports.

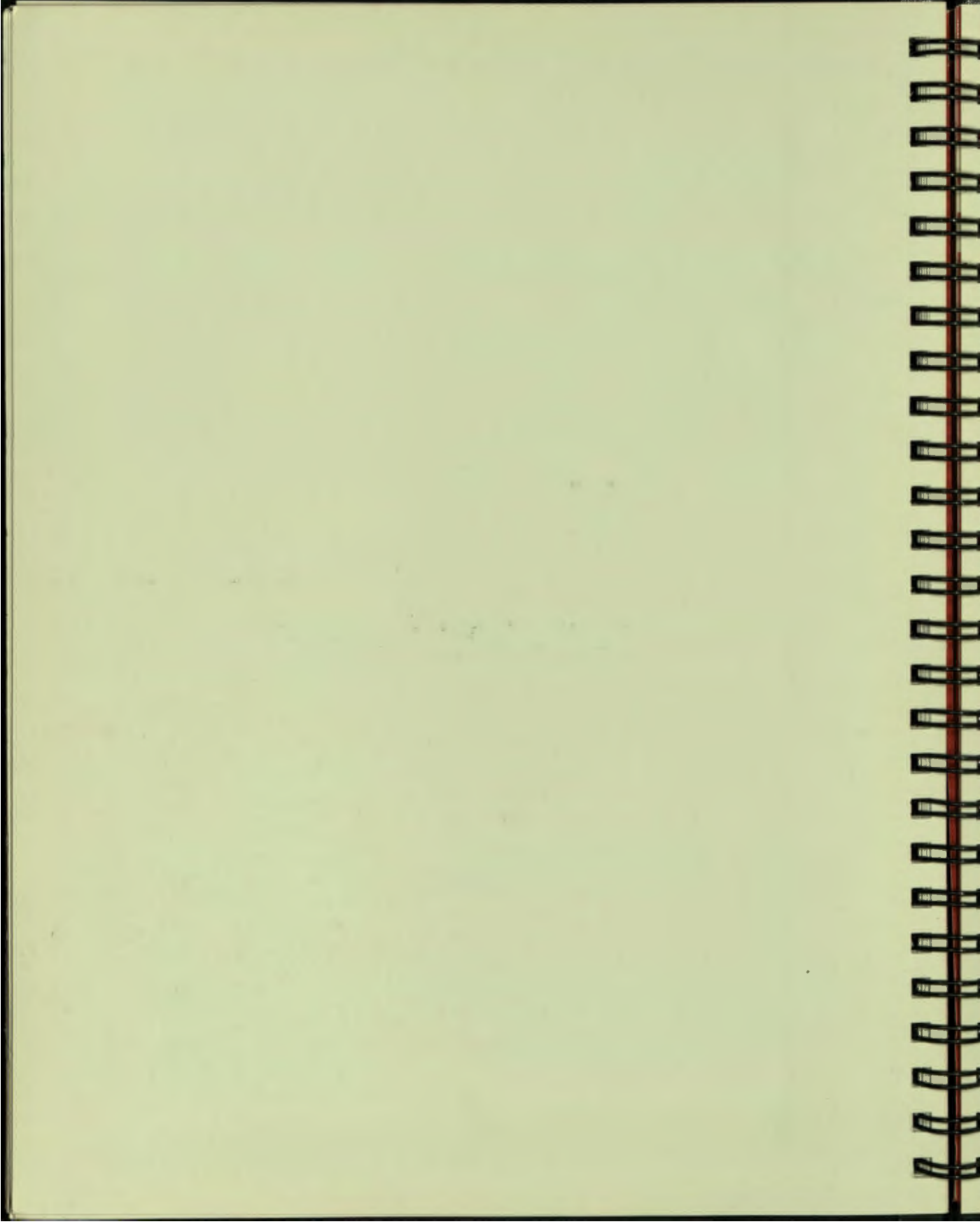




# 7

---

## Creating Reports

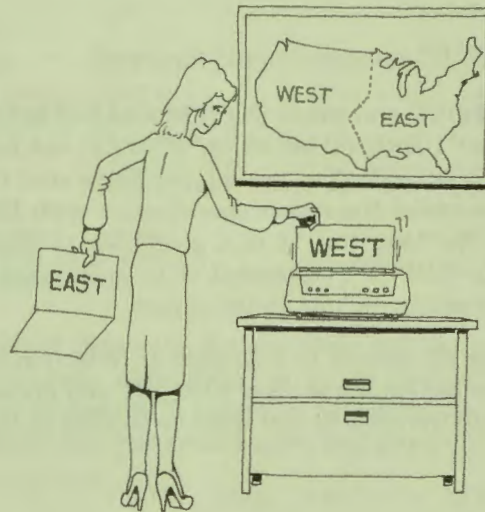


## Chapter 7

---

### Creating Reports

Information stored in a database is useful to you only if you can organize it into a meaningful form such as a report. Sure, you can use the PRINT statement to find the phone number for XYZ company if you are sitting at your Professional, but what if you are somewhere else and need the number? What if you want to print a mailing list? What if you need a sales-to-date report for current customers to refer to in a meeting? Enter the PRO/DATATRIEVE Report Writer.



With the Report Writer, you can organize and summarize information and create clear, easy-to-read reports. You can display reports on your Professional, write a report to a file that you can print on a printer, and edit a report with PROSE to make it part of a larger document.

If you want the report to have a title, you can give it one. If you want to number the pages and print the current date, you can do that too. You can even include summary information in the report. A sales-to-date report, for example, can include lines showing the total sales-to-date and the average sales-to-date.

You can control the format of the report or you can allow the Report Writer to do some or all of the formatting automatically. This chapter introduces you to the basics of using the Report Writer.

#### FORMING A SIMPLE REPORT SPECIFICATION

You create a PRO/DATATRIEVE report with a series of Report Writer statements, called a *report specification*. The simplest form of a report specification includes:

- A REPORT statement that tells PRO/DATATRIEVE you want to create a report and what records to include in the report
- A PRINT statement that tells PRO/DATATRIEVE the fields or records to print in the report
- An END-REPORT statement to end the report

If you only type REPORT and press DO, PRO/DATATRIEVE assumes you want to report on the CURRENT collection. If you do not have a CURRENT collection, PRO/DATATRIEVE does not let you know that it cannot create a report until you have ended the report specification with END-REPORT. To save typing time in the long run, it is a good idea to specify a domain or collection name in the REPORT statement or to include an RSE that identifies the records you want to include in the report.

If you want to include all records in a domain or collection in the report, just type the domain or collection name after REPORT and press DO. Or, you can use an RSE to identify the records you want to include in the report:

```
DTR> REPORT EMPLOYEES WITH (DEPT CONT "MATH")  
RW>
```

After you type a REPORT statement, PRO/DATATRIEVE displays the RW> prompt. This prompt tells you that you can use only Report Writer statements. PRO/DATATRIEVE continues to use the RW> prompt until you type the END-REPORT statement.

The next step in creating a simple report is to type a PRINT statement. The Report Writer PRINT statement tells PRO/DATATRIEVE what fields of the records identified in the REPORT statement to print in the report. Note that a report specification can contain only one PRINT statement, but that you can continue PRINT statements over many lines.

Unlike the PRINT statement you have been using, the Report Writer PRINT statement cannot contain an RSE. It cannot even contain a domain or collection name. To print all the fields in the chosen records, you have to tell the Report Writer to print the top-level field for the records in the report.

A *top-level* field is the first field in a record definition and has a *field level number* of 01. All records have a top-level field that includes all other fields in the record. When you defined your CUSTOMERS domain and record with ADT, PRO/DATATRIEVE added "-REC" to the domain name to create the record name, then used that name as the top-level field name (CUSTOMERS-REC). If you ready the CUSTOMERS domain and type SHOW FIELDS, you see this display:

```
DTR> SHOW FIELDS
CUSTOMERS
  CUSTOMERS_REC
    COMPANY_NAME (COMPANY)      [Character string, indexed key]
    CONTACT_PERSON (CONTACT)   [Character string]
    STREET                      [Character string]
    CITY                        [Character string]
    STATE                       [Character string]
    ZIP_CODE (ZIP)             [Character string]
    PHONE_NUMBER (PHONE)       [Character string]
    SALES_THIS_YEAR (SALES)     [Number]
    LAST_CONTACT (LAST_C)      [Date]
```

The first line displayed shows the domain name and the second line shows the name of the top-level field. ADT always gives the top-level field in a record the same name as the record so that it is easy to remember the name of the top-level field. When you define your own records, you may give the top-level field another name if you like.

To print all fields of the specified records in the CUSTOMERS report, you can simply type PRINT CUSTOMERS-REC in response to the RW> prompt.

If you do not want to print all the fields in the report, you can specify which fields to include by listing the field names, separated by commas. The order in which you list field names determines how the Report Writer displays them on a line.

After you type a PRINT statement and press DO, PRO/DATATRIEVE displays the RW> prompt again, indicating that you need to type another Report Writer statement. You can type END-REPORT to end the report specification, and PRO/DATATRIEVE then displays the report on your terminal.

### TRY IT

Ready the EMPLOYEES domain and type SHOW FIELDS to see the name of the top-level field of EMPLOYEES-REC. Use the REPORT, END-REPORT and PRINT statements to create a report for all records in the EMPLOYEES domain. The report specification should look like this:

```
DTR> REPORT EMPLOYEES
RW> PRINT PERSON
RW> END-REPORT
```

PRO/DATATRIEVE displays the report on your screen, then returns you the DTR> prompt. Notice that the upper right-hand corner of the report shows the current date and a page number. Notice too, that the fields are evenly spaced across the screen. Type PRINT FIRST 2 EMPLOYEES to see the difference between the normal PRINT statement and the Report Writer PRINT statement.

If you want, ready the CUSTOMERS domain and display a report on your screen that includes the company name, contact, and phone fields for companies with state equal to NH. You can form a collection of records and use the collection name in the REPORT statement, or you can use an RSE in the REPORT specification:

```
DTR> REPORT CUSTOMERS WITH (STATE EQUAL "NH")
RW>
```

Practice creating simple reports using RSEs and PRINT statements that list fields to include in the report.

### Titling a Report

If you want to add a title to a report, you can include the SET REPORT-NAME statement in the report specification, like this:

```
RW> SET REPORT-NAME = "CUSTOMERS"
```

You must use quotation marks around the title or PRO/DATATRIEVE displays an error message and you have to start the report specification again.

It doesn't matter whether the SET REPORT-NAME statement comes before or after the PRINT statement, but it is more logical to put it before the PRINT statement. PRO/DATATRIEVE prints the title in the center of the first line of the report before printing any records.

You can give a report a two-line title by separating the text for each line with a slash (/), like this:

```
RW> SET REPORT-NAME = "CONTACTS AND ADDRESSES"/
[Looking for header segment]
RW> "FOR NEW HAMPSHIRE COMPANIES"
```

Note that you must enclose both portions of the title in quotation marks and that you can press DO after the slash. If you press DO after CONTACTS or AND, PRO/DATATRIEVE displays an error message and you have to type the whole report specification again. The beginning and ending quotation marks for each segment of the title have to be on the same line.

### **TRY IT**

Form a collection of EMPLOYEES records and create a report with a two-line title. PRO/DATATRIEVE centers the title on the first two lines of the report with the first portion on the same line as the date and the second portion on the same line as the page number.

Create a report with a three-line title to see where PRO/DATATRIEVE puts the first line of the title. The last line of a title is always on the same line as the page number.

### **Writing Reports to a File**

You can save reports by including the keyword ON and a file name at the end of the REPORT statement, like this:

```
DTR> REPORT EMPLOYEES ON EMPREP
```

This statement tells PRO/DATATRIEVE to open a file named EMPREP and to write the report to that file. When you write a report to a file, you do not see the report on your screen.

PRO/DATATRIEVE stores the file in your current directory and assigns it a .LST file type. If you want to store the file on a diskette or in another directory, or if you want the file to have a different file type, you have to specify an extended file name, like this:

```
CUSTOMERS:[CUSTINFOJEMPREP.REP
```

For information on using extended file names, see your Professional 350 *Hard Disk User's Guide*.

For now, specify only the file name or the file name and a .TXT file type for your reports. Your P/OS operating system converts a .TXT file type to "Text" when you display a listing of the directory that contains the file.

### **TRY IT**

Now you can create a phone list from your CUSTOMERS domain and write it to a file. Create a report specification to write CUSTOMER contact and phone information (COMPANY, CONTACT, PHONE) to a file named PHONE.TXT. Your report specification should look something like this:

```
DTR> REPORT CUSTOMERS ON PHONE.TXT
RW> SET REPORT-NAME = "CUSTOMER CONTACTS"/"AND PHONE NUMBERS"
RW> PRINT COMPANY, CONTACT, PHONE
RW> END-REPORT
DTR>
```

If you like, exit from PRO/DATATRIEVE and use "Print services" to print a copy of your report. You can also edit the file with PROSE if you want to make some changes and additions.

Now try creating a report that contains mailing list information. Determine which fields you want to include in the report, then list those fields in the PRINT statement. This report may not be as useful as you would like because the PRINT statement prints all the fields on the same line. Later in this chapter, you learn how to use the PRINT statement to print information in a record on more than one line.

### **Setting Report Width**

With the Report Writer's SET COLUMNS-PAGE statement, you can control the width of a report. When you do not explicitly set the width of a report, the Report Writer uses an 80-character line.



This statement is particularly useful when you write a report to a file. Customer records, for example, exceed 80 characters and an entire record wraps when you write a report. If you set the column width to 132 characters with SET COLUMNS-PAGE = 132, PRO/DATATRIEVE uses a 132-character line for the report. Customer records still wrap when you display a report on your screen, but when you write a customer report to a file, a customer record fits on one line. When you print the report on a printer, the records are printed on one line. If you press the SETUP key and set your screen to 132 columns, you can use “File services” to display the file in small characters with each record on one line.

### TRY IT

Type in the following report specification to create a 132-column report containing customer records and display the report on your screen:

```
DTR> READY CUSTOMERS
DTR> REPORT CUSTOMERS
RW> SET COLUMNS-PAGE = 132
RW> PRINT CUSTOMERS-REC
RW> END-REPORT
```

Because the screen can contain only 80 characters per line, the records wrap. Use the PRINT statement to display customer records on your screen. Because the Report Writer spaces fields evenly across a line, the records wrap differently.

Now write the same report to a file. Exit from PRO/DATATRIEVE, use the SETUP key to set your screen to 132 columns, and use “File services” to display the report file. P/OS uses small characters and displays your customer report with each record on one line. If you like, print a copy of the report on your printer.

You can also create reports with widths smaller than 80 characters. Create a 50-column report containing the COMPANY and CONTACT fields to see how the Report Writer formats the report. Then create the same report without the SET COLUMNS-PAGE statement to see the difference between a 50-column report and an 80-column report.

### FORMATTING REPORT LINES

With the Report Writer, you can format *detail lines* and *summary lines* of your reports. Detail lines contain information from individual records in the report. Summary lines summarize information included in the report. The Report Writer

can, for example, count the number of records in a report and print that count in the report. The Report Writer can also calculate totals and averages for numeric values in a column.

### Formatting Detail Lines

To format detail lines, you include *print list items* in the PRINT statement. Print list items are PRO/DATATRIEVE keywords, symbols, and field names that tell the Report Writer where, what, and how to print record fields.

Useful print list items include:

- SKIP number – If you do not specify a number, the Report Writer skips to the next line. If you specify a number, the Report Writer skips that many lines. SKIP 2, for example, tells the Report Writer to skip two lines before printing the next field.
- COL number – Tells the Report Writer to print the next field in the column specified by the number. COL 10, for example, tells the Report Writer to print the next field in the tenth column.
- SPACE number – Tells the Report Writer to add one or more spaces between fields. If you do not specify a number, the Report Writer adds one space to the end of the field. If you specify a number, the Report Writer adds that many spaces. SPACE 2, for example, tells the Report Writer to skip two spaces between fields.
- (-) – Tells the Report Writer to not print column headings for the immediately preceding field. COMPANY(-), for example, tells the Report Writer to print COMPANY fields without a column heading.

The way you combine print list options determines how the Report Writer formats each record in the report. To print a blank line between each record and suppress column headings, for example, use a PRINT statement like this:

```
RW> PRINT COMPANY(-), CONTACT(-), PHONE(-), SKIP
```

To print record fields in vertical order, use the COL option:

```
RW> PRINT COL 5, COMPANY(-), COL 5, CONTACT(-),  
[Looking for next element in list]  
RW> COL 5, PHONE(-), SKIP
```

To print two fields separated by spaces, use the SPACE option:

```
RW> PRINT COL 5, COMPANY(-), SPACE, CONTACT(-)
```

The space option appears to leave more than one space between fields when the field that precedes the SPACE keyword does not contain as many characters as the field can contain. A report specification that includes the preceding PRINT statement, for example, formats the first two customer records like this:

```
DTR> REPORT FIRST 2 CUSTOMERS WITH ZIP = 0
RW> SET COLUMNS-PAGE = 50
RW> PRINT COL 5, COMPANY(-), SPACE, CONTACT(-)
RW> END-REPORT
```

1-Jun-83  
Page 1

```
INFO ANALYSIS      WILMA SWAYER
PRODUCTS UNLIMITED DANA MCMANUS
```

DTR>

The COMPANY field, as defined in the CUSTOMERS-REC record definition, can contain as many as 20 characters, so the Report Writer prints the company name as a 20-character field, then skips one space before printing the contact name.

Note that you must separate multiple print list options with commas, or PRO/DATATRIEVE displays an error message and does not produce the report. If you need to continue the print list on a new line, be sure to include a comma at the end of the first line as in the previous examples. Then the Report Writer prompts for you to continue the list with "Looking for next element in list."

### TRY IT

By using print list options, you can now create a useful mailing list. Ready the CUSTOMERS domain and type in the following report specification:

```
DTR> REPORT FIRST 2 CUSTOMERS
RW> SET REPORT-NAME = "MAILING LIST TEST"
RW> SET COLUMNS-PAGE = 40
RW> PRINT COL 5, COMPANY(-),
[Looking for next element in list]
RW> COL 5, CONTACT(-),
[Looking for next element in list]
RW> COL 5, STREET(-),
[Looking for next element in list]
RW> COL 5, CITY(-), SPACE, STATE(-), SPACE 3, ZIP(-), SKIP
RW> END-REPORT
```

(continued on next page)

MAILING LIST TEST    12-May-83  
Page 1

BASIC SYSTEMS  
ANNE DUGGAN  
43 HIGHLAND AVE  
BOSTON        MA    13532

INFO ANALYSIS  
WILMA SWAYER  
9 REDACTOR  
DUNKIRK       MA

DTR>

Do not forget to include commas after every print list element before pressing DO to continue the list on a new line. If you forget a comma, the Report Writer thinks you are finished with the list and looks for a new Report Writer statement and when it encounters the next print list element, displays an error message.

In the above example, each line of the report specification corresponds to a line of the report. Breaking a long PRINT statement in this way helps show the logic of the statement. Just remember to include a comma at the end of the line so that PRO/DATATRIEVE does not think you are finished with the PRINT statement.

Try formatting the details of your report in other ways, using SKIP, COL, and SPACE in various places in the PRINT statement. Create a mailing list with a larger column setting to see how it looks.

To get rid of the extra spaces between the city and state and add a comma to make the record look like a real address, do this:

```
RW> PRINT COL 5, COMPANY(-),  
[Looking for next element in list]  
RW> COL 5, CONTACT(-),  
[Looking for next element in list]  
RW> COL 5, STREET(-),  
[Looking for next element in list]  
RW> COL 5, CITY!!", ":STATE(-), SPACE 3, ZIP(-), SKIP
```

Two vertical bars (| |) tell the Report Writer to discard blank characters at the end of the immediately preceding field and to print the characters in quotation marks (a comma and one space) exactly as they appear. The single vertical bar (|) before STATE tells the Report Writer to print the STATE field immediately after the space. The single and double vertical bars are called *concatenation sym-*

*bols* because they tell the Report Writer to *concatenate*, or join, fields and literal values together. Literal values are those enclosed in quotation marks.

If you want to print only one space between the STATE field and ZIP field, change the line that formats CITY, STATE, and ZIP to this line:

```
RW> COL 5, CITY!!!, " !STATE!!" !ZIP, SKIP
```

The two vertical bars after STATE tell the Report Writer to ignore any trailing blanks, just as the bars after CITY do, and to treat the quoted string as it appears. The single vertical bar before ZIP tells the Report Writer to print the ZIP code field immediately after the quoted string.

Create a report specification that uses COL, SPACE, and concatenation symbols to make customer records look like real addresses. If you like, play with formatting detail lines using concatenation symbols and quoted literals. Remember that two vertical bars tell the Report Writer to discard trailing blanks, while one vertical bar tells the Report Writer to print the field immediately after the last item, without skipping any space.

### Formatting Summary Lines

By using the Report Writer AT BOTTOM OF REPORT PRINT statement, you can create and display lines that summarize data in your report. The keywords OF REPORT in the statement tell the Report Writer to print the summary lines at the end of the report and the keyword PRINT tells the Report Writer what to include in the summary line. Only one AT BOTTOM OF REPORT PRINT statement can be used in a report specification.

To specify what data to include in the summary line, you include statistical keywords like COUNT, AVERAGE, and TOTAL in the AT BOTTOM statement:

- COUNT - Tells the Report Writer to count the number of records in the report and to print that number at the bottom of the report.
- AVERAGE field name - Tells the Report Writer to calculate the average value for the specified numeric field and print it at the bottom of the field column.
- TOTAL field name - Tells the Report Writer to total the values in the field and print the total at the bottom of the field column. If you want the field printed in money format, with a leading dollar sign, commas, and a decimal point, you need to include an edit string. Just as ADT included an EDIT-STRING clause and the \$\$\$,\$\$\$.\$\$ edit string in the SALES field

definition, you can include an edit string in the field definition for the TOTAL field. Just type the keyword USING and a money edit string like \$\$\$, \$\$\$, \$\$\$, \$\$\$. Since a total can contain more digits than the values being totalled, the edit string should allow for more digits. A \$\$\$, \$\$\$, \$\$\$, \$\$\$, \$\$ edit string tells PRO/DATATRIEVE that the total value can have as many as eight digits and two commas to the left of the decimal point and two digits to the right.

To identify what the values displayed at the bottom of a report represent, you can include quoted literals like "TOTAL SALARY:" and "AVERAGE SALARY:" in the AT BOTTOM statement, like this:

```
RW> AT BOTTOM OF REPORT PRINT SKIP,
[Looking for next element in list]
RW> "TOTAL SALARY:", TOTAL SALARY USING $$$, $$$, $$$, $$ SKIP,
[Looking for next element in list]
RW> "AVERAGE SALARY:", AVERAGE SALARY
```

If you do not want zero values to appear as blanks, you can specify an edit string like \$\$\$, \$\$\$, \$\$\$9.99. PRO/DATATRIEVE then displays a zero value as \$0.00.

To display the number of records in a report, you might include the COUNT statistical function in the AT BOTTOM statement:

```
RW> AT BOTTOM OF REPORT PRINT SKIP,
[Looking for next element in list]
RW> COL 1, "NUMBER OF RECORDS: " !COUNT
```

### TRY IT

The best way to learn to create summary lines in a report is to experiment with the AT BOTTOM OF REPORT statement. To get you started, ready the EMPLOYEES domain and type in this report specification:

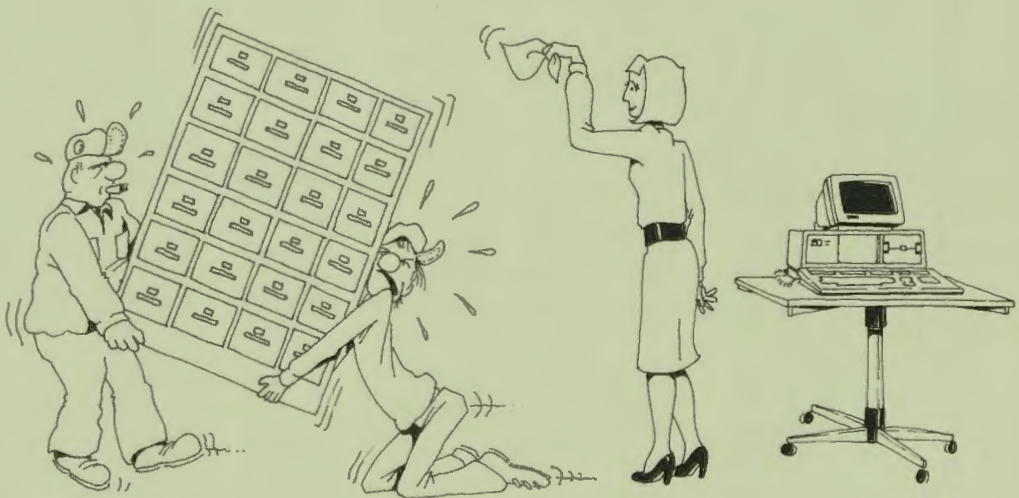
```
DTR> REPORT FIRST 5 EMPLOYEES
RW> PRINT PERSON
RW> AT BOTTOM OF REPORT PRINT SKIP,
[Looking for next element in list]
RW> COL 1, "TOTAL SALARY:", TOTAL SALARY USING $$$, $$$, $$$, $$,
[Looking for next element in list]
RW> COL 1, "AVERAGE SALARY:", AVERAGE SALARY,
[Looking for next element in list]
COL 1, "NUMBER OF RECORDS: " !COUNT
END-REPORT
```

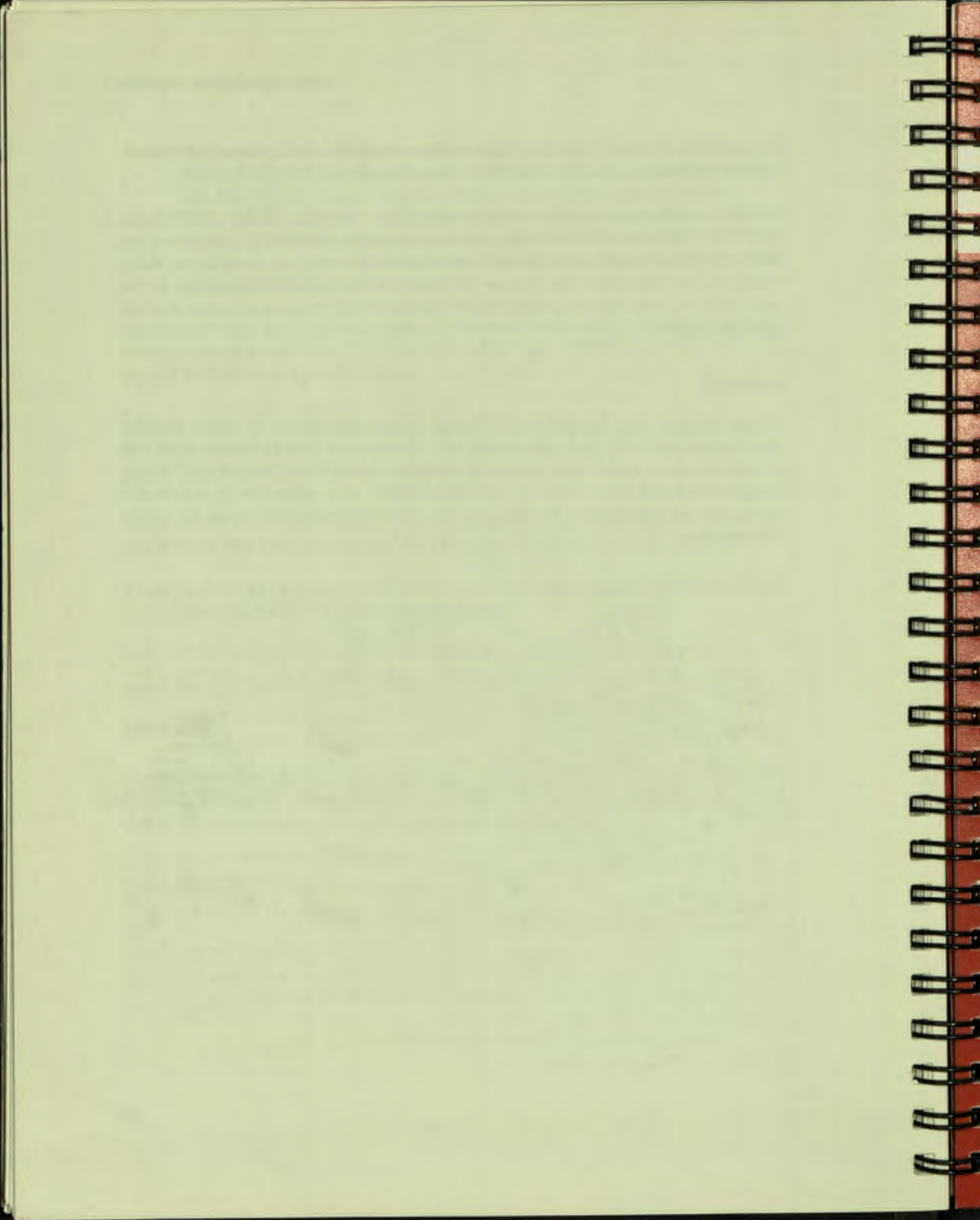
To keep from having to watch all the employee records scroll across your screen, practice formatting detail and summary lines for only the first few records.

Practice creating some simple reports using COL, SPACE, SKIP, AVERAGE, COUNT, TOTAL, quoted strings, and concatenation symbols. Do not spend too much time creating detailed reports, as you can become frustrated if you make typing errors that force you to type the entire report specification again. In the next chapter, you learn to create report specifications that you can save and use again and again.

### SUMMARY

In this chapter, you learned to use Report Writer statements to create and format reports and to write a report to a file. Because the Report Writer gives you so many ways to control the format of a report, report specifications can be very complex and tedious to type. In the next chapter, you learn how to create and save a report specification so that you can use it over and over again by typing only one line.



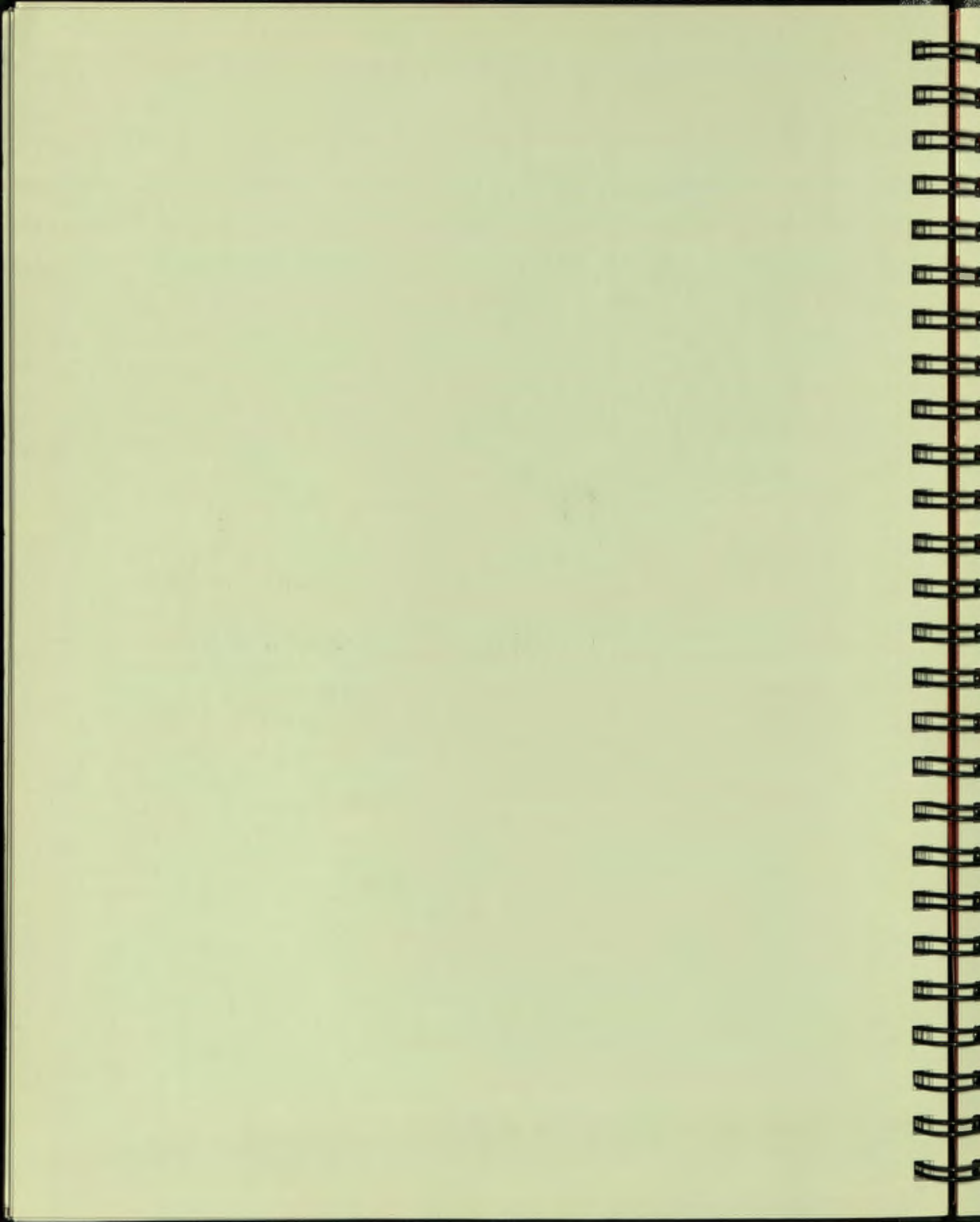




# 8

---

## Saving Typing Time



## Chapter 8

---

### Saving Typing Time

PRO/DATATRIEVE provides you with several techniques to let you spend more time planning your database and report specifications and less time typing. You've already learned to save time by using ADT, the Application Design Tool, to create domains and record definitions. Other ways to save time include using Guide Mode, defining *procedures* that PRO/DATATRIEVE stores in your dictionary for you to use again and again, and using command files.

#### USING GUIDE MODE

Guide Mode helps you work step-by-step through a PRO/DATATRIEVE session and saves you typing time by providing additional prompts and information as you type in commands and statements. To use Guide Mode, type SET GUIDE in response to the DTR> prompt.

After you type SET GUIDE, you see the following line:

```
Enter command, type ? for help
```

You can type a question mark or press the HELP key to see that your options at that point are READY, SHOW, and LEAVE. Guide Mode helps you make entries by letting you know what your options are. It can also complete your entries so you do not have to type the entire command or statement. All you have to do is type enough letters to uniquely identify the word you are typing, like R, S, or L, and press the space bar. Guide Mode then completes typing the word for you. Your possible options change as you go along, depending on where you are in creating a command or statement. Note, though, that not all PRO/DATATRIEVE commands and statements are available in Guide Mode.

Throughout, Guide Mode prompts you for the next part of your entry, including spaces, quotation marks, and returns. You can press DO or RETURN wherever Guide Mode tells you you can press "<return>".

By pressing the HELP key or typing a question mark, you can find out which commands, statements, keywords, or symbols you can use at that time. PRO/DATATRIEVE immediately displays all the acceptable possibilities. If you use a word or symbol that Guide Mode does not recognize, PRO/DATATRIEVE displays an error message.

You can exit from Guide Mode anytime you see the "Ready for next command" line by typing L (for LEAVE) or by pressing the EXIT or MAIN SCREEN key. After you exit from Guide Mode, PRO/DATATRIEVE displays the message "You're on your own now. Good Luck!" and displays the DTR> prompt. If you want to return directly from Guide Mode to the Main Menu, press the INTERRUPT key and then the DO key.

### **TRY IT**

The best way to learn about Guide Mode is to try it. Type SET GUIDE and experiment by typing a question mark to see your options at different points and trying different options. When you finish experimenting, type L to return to PRO/DATATRIEVE command level.

## **CREATING AND USING PROCEDURES**

Another way to save typing time is to define frequently-used sequences of commands and statements as procedures. A procedure is a fixed sequence of commands and statements that PRO/DATATRIEVE stores in your current dictionary with the name you specify in the DEFINE PROCEDURE command. When you type a colon (:) and the procedure name, PRO/DATATRIEVE executes the procedure as though you had just typed the commands and statements it contains.

As you saw in the last chapter, report specifications can be complex to create and type. By putting Report Writer statements in a procedure, you can save report specifications and use them again.

To define a procedure, type DEFINE PROCEDURE and a procedure name. A procedure name can consist of from 1 to 31 characters, digits, hyphens, and underscores, but it must begin and end with a character or digit.

For example:

```
DTR> DEFINE PROCEDURE SALARY-REPORT
DFN>
```

The DFN> prompt indicates that PRO/DATATRIEVE expects a command or statement.

Type each command and statement you want to include in the procedure just as you would in response to the DTR> prompt. Note, though, that when you span long statements over more than one line, PRO/DATATRIEVE does not display the CON> prompt. Be particularly careful where you break a statement. PRO/DATATRIEVE does not detect mistakes until you execute the procedure with the colon.

When you finish typing the commands and statements you want to store in the procedure, use END-PROCEDURE to end the procedure definition. PRO/DATATRIEVE then stores the procedure in your dictionary.

To see the names of procedures stored in your dictionary, use the SHOW PROCEDURES command. To see the contents of a particular procedure, type SHOW and the procedure name.

If you no longer need a particular procedure, you can delete it with the DELETE command, like this:

```
DTR> DELETE OLD-SALARY-REPORT;
```

Note that you must end the DELETE command with a semicolon. If you do not, PRO/DATATRIEVE displays an error message.

You can also delete domain and record definitions with the DELETE command. Just use the domain or record name in the DELETE command and end the command with a semicolon. Deleting domain and record definitions you no longer need or use helps keep your data dictionary organized and up-to-date.

### **TRY IT**

Define a procedure that creates a salary report for the EMPLOYEES domain. You can even include a READY command in the procedure so that you do not have to ready the domain before executing the procedure. Be sure to begin the procedure definition with the DEFINE PROCEDURE command and to end it with the END-PROCEDURE command.

After you have defined a procedure, use SHOW commands to see that the procedure is stored in your dictionary and to see the contents of the procedure. Then execute the procedure by typing a colon and the procedure name.

Don't worry if PRO/DATATRIEVE detects an error and does not do what you expect. In the next section, you learn to edit procedures and other dictionary objects to correct mistakes.

For practice, type the following procedure exactly as it appears here:

```
DTR> DEFINE PROCEDURE SALARY-REPORT
DFN> | By preceding a line with an exclamation point, you
DFN> | can include comments that tell what the procedure
DFN> | does. You can also use exclamation points to include
DFN> | blank lines to make a procedure easier to read.
DFN> |
DFN> | This procedure prints records from the EMPLOYEES domain, then
DFN> | prints the number of records, the average salary, and the total
DFN> | salary at the bottom of a report.
DFN> |
DFN> PRINT ID, STATUS, FIRST_NAME||" "||LAST_NAME, DEPT, SALARY
DFN> AT BOTTOM OF REPORT PRINT SKIP,
DFN> "NUMBER OF RECORDS: "||COUNT,
DFN> COL 40, "AVERAGE SALARY:", AVERAGE SALARY, SKIP,
DFN> COL 40, "TOTAL SALARY:", TOTAL SALARY USING $$$, $$$, $$$
DFN> END_REPORT
DFN> END_PROCEDURE
```

Notice the exclamation points and comments in this procedure. An exclamation point tells PRO/DATATRIEVE to ignore the rest of the line. You can make procedures easier to read and use by including comments.

Notice, too, that this procedure definition does not include a REPORT statement. How, then, can you use this procedure to produce a report? You can execute the procedure after typing the READY command and REPORT statement:

```
DTR> READY EMPLOYEES
DTR> REPORT EMPLOYEES WITH (DEPT CONT "MATH") AND
[Looking for Boolean expression]
RW> (STATUS EQUAL "EXPERIENCED")
RW> :SALARY-REPORT
```

1-Jun-83  
Page 1

ID	STATUS		DEPT	SALARY
00001	EXPERIENCED	GEORGE BOOLE	MATHEMATICS	\$2,000
00891	EXPERIENCED	FRED HOWL	MATHEMATICS	\$59,594
78375	EXPERIENCED	ALBERT EINSTEIN	MATHEMATICS	\$40,095
83764	EXPERIENCED	JIM MEADER	MATHEMATICS	\$41,029

NUMBER OF RECORDS:	4	AVERAGE SALARY:	\$35,679
		TOTAL SALARY:	\$142,718

DTR&gt;

By not including the READY command and REPORT statement in the procedure, you can use the procedure to create reports for different groups of records instead of for all the records in the EMPLOYEES domain or a single group of employees specified within the procedure. When you specify the records to include in a report in a report procedure, you limit the versatility of your procedure.

To help you work as efficiently as possible, you can execute procedures after commands and statements and even from within statements. If you do not want to think about edit strings, for example, you could create a little procedure that defines a money edit string the same way all the time. You could then use this procedure whenever you wanted to specify a money edit string. For example:

```
DTR> DEFINE PROCEDURE MONEY-EDIT
DFN> USING $$,$$$,$$$
DFN> END-PROCEDURE
DTR> REPORT EMPLOYEES WITH DEPT CONT "BIOL"
RW> SET REPORT-NAME = "BIOLOGY SALARIES"
RW> SET COLUMNS-PAGE = 40
RW> AT BOTTOM OF REPORT PRINT COL 5, "TOTAL SALARY:",
[Looking for next element in list]
RW> TOTAL SALARY :MONEY-EDIT
RW> END-REPORT
```

BIOLOGY SALARIES 12-May-83  
Page 1

	SALARY
TOTAL SALARY:	\$169,547

DTR&gt;

Once you have defined the MONEY-EDIT procedure, you can use it over and over again without having to think about how to format money values. You can even execute the MONEY-EDIT procedure from another procedure, such as a report procedure, like this:

```
DTR> DEFINE PROCEDURE SUM-REPORT
DFN> SET COLUMNS-PAGE = 50
DFN> AT BOTTOM OF REPORT PRINT SKIP,
DFN> COL 5, "TOTAL SALARY:", TOTAL SALARY :MONEY-EDIT, SKIP,
DFN> COL 5, "AVERAGE SALARY:", AVERAGE SALARY :MONEY-EDIT
DFN> END-REPORT
DFN> END-PROCEDURE
DTR> REPORT EMPLOYEES WITH DEPT CONT "MATH"
RW> SET REPORT-NAME = "MATH SALARIES"
RW> PRINT SALARY
RW> :SUM-REPORT
```

MATH SALARIES

12-May-83

Page 1

SALARY

```
$2,000
$59,594
$14,000
$20,000
$20,000
$23,000
$40,095
$41,029
```

```
TOTAL SALARY:    $219,718
AVERAGE SALARY:  $29,714
```

DTR&gt;

Experiment with defining and executing procedures. After a little practice, you will find that procedures save you thinking time as well as typing time. You have to think about where to put a COL or SKIP or SPACE in a report specification only when you create the procedure. Once you finish, PRO/DATATRIEVE does the work for you.



## EDITING ITEMS IN YOUR DICTIONARY

When you execute a procedure, PRO/DATATRIEVE checks each line of the procedure for errors. If the procedure contains an error, PRO/DATATRIEVE displays a message. You could type the entire procedure over again, but it is easier to copy a long procedure to a command file. You can then exit from PRO/DATATRIEVE and edit the command file with your PROSE editor. When you finish, select PRO/DATATRIEVE from the Main Menu and execute the command file by typing @ and the file name when you see the DTR> prompt.

PRO/DATATRIEVE executes the commands and statements in the command file as though you typed them in response to the DTR> prompt, just as it executed the command file created by ADT that contained domain, record, and file definition commands.

To copy a dictionary definition to a command file, use the EXTRACT command, like this:

```
DTR> EXTRACT ON REPORTS SUM-REPORT
DTR>
```

This EXTRACT command tells PRO/DATATRIEVE to create a command file named REPORTS and to copy the SUM-REPORT procedure to the file. If you want, you can specify REPORTS.CMD, but it isn't necessary. As with other files on your Professional, a command file name can contain from one to nine letters (A-Z) and/or numbers (0-9). PRO/DATATRIEVE assumes that the word following ON names the command file you want to create and assigns a .CMD file type to the file.

You can then exit from PRO/DATATRIEVE and use PROSE to edit REPORTS.CMD. PRO/DATATRIEVE even includes a DELETE command in the file so that when you later execute the file to store the modified procedure definition, you do not get an error message. Remember that a domain, record, or procedure name cannot duplicate an existing dictionary name. If PRO/DATATRIEVE did not include the DELETE command in the command file, you would have to type it yourself to delete the old definition before you could execute the command file containing the new definition.

When you execute a command file to store a procedure definition, PRO/DATATRIEVE stores the definition, but does not execute the procedure. To execute the procedure, type a colon (:) and the procedure name.

As you experiment more with PRO/DATATRIEVE, you will find command files very useful. You can create command files that contain more than one existing dictionary definition by separating the names with commas, like this:

```
DTR> EXTRACT ON REPORTS SUM-REPORT, SALARY-REPORT, MONEY-EDIT
DTR>
```

PRO/DATATRIEVE puts all the specified procedures in REPORTS.CMD and includes DELETE commands for each one. You can then edit all the procedures at one time by using PROSE.

### **TRY IT**

Use the EXTRACT command to copy all your existing procedures to a command file named PROCEDURE. Then exit from PRO/DATATRIEVE and select PROSE from the Main Menu. Modify any or all of the procedures to make them work just as you want, then exit from PROSE and choose PRO/DATATRIEVE from the Main Menu.

Type @PROCEDURE to execute the command file and watch your screen. PRO/DATRIEVE displays each statement and command on your screen as it executes, just as though you typed them in response to the DTR> prompt. When the command file finishes executing, you see the DTR> prompt again and can continue doing what you like.

If you like, use the SHOW command to look at your modified procedures. The command file deleted the old procedure definitions and created new ones for you.

As you use PRO/DATATRIEVE more and more, you can save a lot of typing time by creating command files with PROSE. If PRO/DATATRIEVE detects any errors, you can then edit the command file with PROSE and do not have to type a definition from scratch.

You can also create command files with your PROSE editor to store new definitions and procedures in your dictionary. When you exit from PROSE, press the ADDTNL OPTIONS key and choose "Use extended file name" to give the file a .CMD file type. Then, when you execute the command file, you do not need to specify the file type as PRO/DATATRIEVE always searches for a file with a .CMD file type.

Note that when you create or edit a command file with PROSE, you *must not* save the file with word wrapping turned off. If you do, PRO/DATATRIEVE cannot execute the command file and displays an error message.

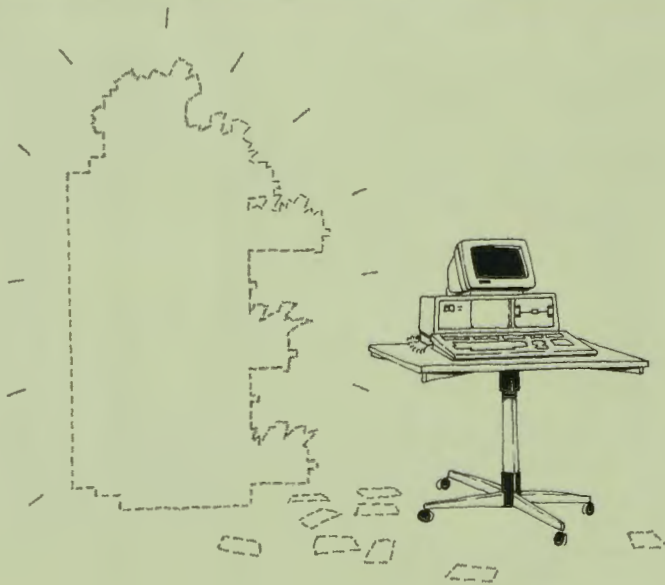
PRO/DATATRIEVE executes commands and statements in a command file just as though you typed them in response to the DTR> prompt. If you have made an error in a command or statement, PRO/DATATRIEVE displays an error message and stops executing the command file. When this happens, exit from PRO/DATATRIEVE and use PROSE to edit the command file. After you correct the error, exit from PROSE and choose PRO/DATATRIEVE again from the Main Menu. Then execute the modified command file.

### SUMMARY

In this chapter you learned to use Guide Mode, to define and execute procedures, and to create and execute command files. As you continue to use PRO/DATATRIEVE, remember to use procedures whenever possible to save yourself time and frustration. Once a procedure works the way you want it, you can use it forever.

### CONCLUSION

This book has introduced you to the basics of using PRO/DATATRIEVE as an information management tool. You may want to go back over some sections to reinforce what you have learned. Use the glossary if you have any questions about the terms and concepts introduced in this book.

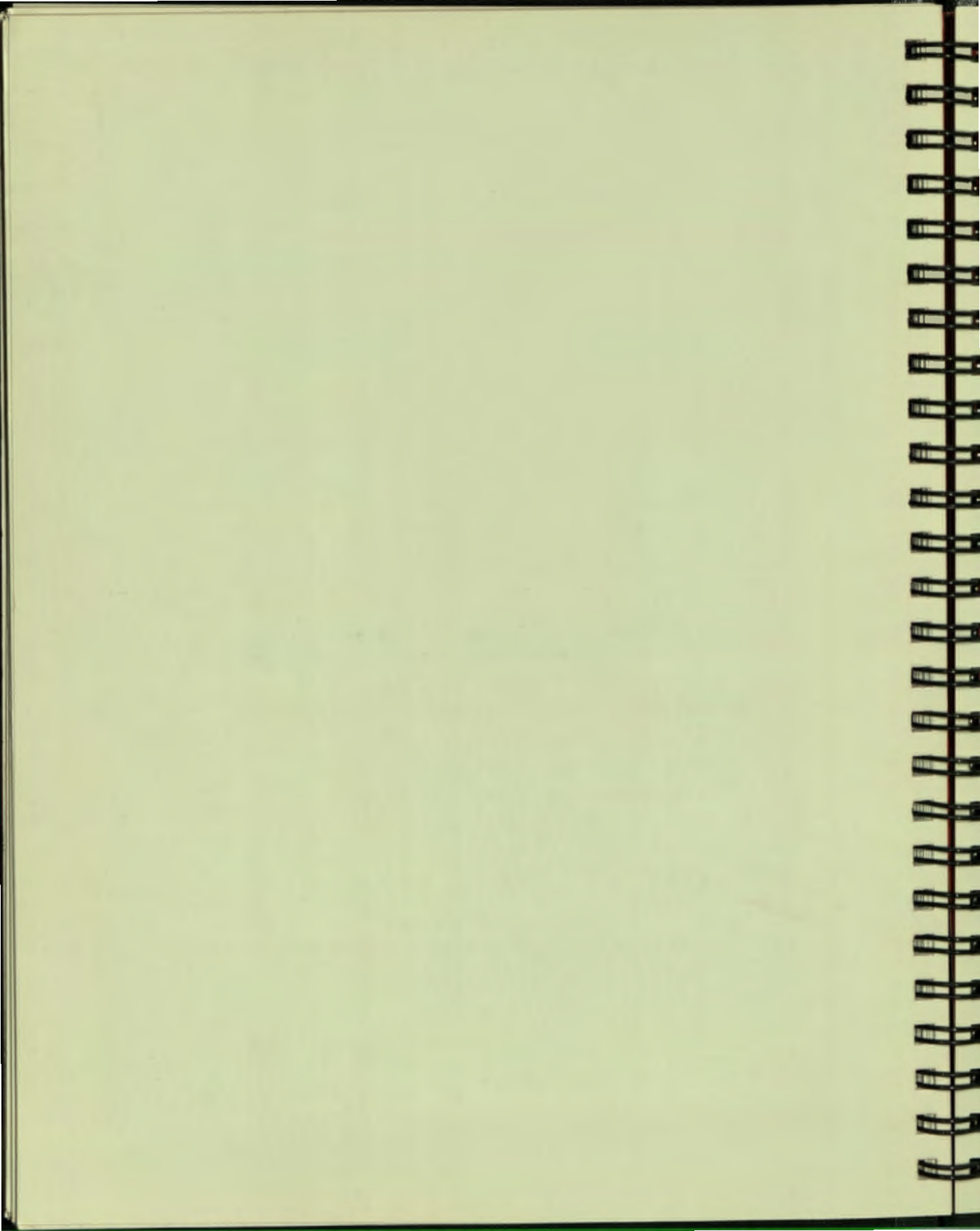


By using the PRO/DATATRIEVE features discussed in this book, you can perform a variety of information management tasks, but PRO/DATATRIEVE has more tricks and features than this book covers. You can, for example, edit dictionary definitions without leaving PRO/DATATRIEVE by using the PRO/DATATRIEVE Editor as explained in Chapter 17 in the *PRO/DATATRIEVE Handbook*.

When you want to learn more, read the *PRO/DATATRIEVE Handbook*. If you just have a question about performing a particular task, refer to the reference section of the *PRO/DATATRIEVE Handbook* to see what commands, statements, or keywords to use.

---

## Glossary and Index



# Glossary

---

## **ADT**

A command that lets you use the Application Design Tool.

## **Application Design Tool**

An interactive feature of PRO/DATATRIEVE that helps you define domains, records, and data files by asking questions and prompting for all the information needed.

## **Arithmetic Operator**

A symbol used to perform arithmetic operations. PRO/DATATRIEVE operators include + (addition), - (subtraction), \* (multiplication), and / (division).

## **AT BOTTOM OF REPORT PRINT**

A Report Writer statement that creates and prints summary lines at the end of the report. A report specification can contain only one AT BOTTOM OF REPORT PRINT statement.

## **AVERAGE**

A statistical function that calculates the average of a group of values.

## **Boolean Expression**

An expression that tells PRO/DATATRIEVE how to compare records and what values to use in the comparison. Boolean expressions may contain field names, relational operators, values, and Boolean operators. PRO/DATATRIEVE evaluates Boolean expressions to true or false.

## GLOSSARY

### **Boolean Operator**

A keyword that joins Boolean expressions together into one expression. Boolean operators include AND and OR.

### **Collection**

A group of records gathered together from a larger group of records with the FIND statement. You can name a collection and can have several named collections available at once. To refer to an established collection, use the collection's name.

### **Column Header**

Headings displayed over columns of data displayed on your screen or printed in a report. PRO/DATATRIEVE uses field names as column headers unless a field definition contains a QUERY-HEADER clause that specifies a query header.

### **Command**

One or more keywords, like DEFINE, SET, SHOW, FINISH, RELEASE, and READY, that you use to define data domains, records, procedures, and tables and to control PRO/DATATRIEVE.

### **Command File**

A file with a .CMD file type that contains PRO/DATATRIEVE commands and statements. You tell PRO/DATATRIEVE to execute the command file by typing an at sign (@) and the file name. Then PRO/DATATRIEVE displays and executes each statement and command in the file as though you just typed them.

### **Condition Clause**

A component of an RSE that tells PRO/DATATRIEVE the characteristics of records you want in the record stream. It consists of the word WITH and a Boolean expression.

### **CURRENT**

The name PRO/DATATRIEVE gives to a collection formed with the FIND statement. When you form a new collection with another FIND statement, the new collection replaces the old one as the CURRENT collection.

### **Data**

Information. Data can range from the smallest meaningful piece of information, such as a word or number, to a huge database consisting of many



files. PRO/DATATRIEVE stores data in fields, records, and files.

**Data Dictionary**

A file that contains information about your data and its organization. PRO/DATATRIEVE automatically connects you to a dictionary every time you choose PRO/DATATRIEVE from the Main Menu. To create a dictionary, use the DEFINE DICTIONARY command. To change your current dictionary, use the SET DICTIONARY command.

**Data File**

A collection of related records stored as a unit on your Professional 350 disk or on a diskette. To create a data file, use the DEFINE FILE command.

**Database**

Information maintained on a computer and organized to facilitate immediate and accurate inquiry and update.

**DEFINE**

A command that lets you define PRO/DATATRIEVE dictionaries (DEFINE DICTIONARY), domains (DEFINE DOMAIN), records (DEFINE RECORD), data files (DEFINE FILE), or procedures (DEFINE PROCEDURE).

**DELETE**

A command that deletes a domain, record, table, or procedure definition from your current data directory.

**Detail Lines**

Lines in a report that contain information from individual records in the report.

**Dictionary Object**

A data structure definition (domain, record, or table) or command sequence (procedure) stored in a dictionary file. Dictionary objects tell PRO/DATATRIEVE how to interpret data and what to do with data.

**Domain**

A named data structure that associates a data file on your disk or diskette with a record definition. To create a domain, use the DEFINE DOMAIN command. When you use a domain name, you tell PRO/DATATRIEVE

## GLOSSARY

to use a particular record definition to interpret information stored in a specific file. A domain itself does not contain data.

### **EDIT-STRING**

A field definition clause that tells PRO/DATATRIEVE how to display and print values stored in a field. An EDIT-STRING clause overrides a field's PIC or USAGE clause.

### **END-PROCEDURE**

A statement that signals the end of a procedure and returns you to the DTR> prompt.

### **END-REPORT**

A statement that signals the end of a report specification and returns you to the DTR> prompt.

### **Environment**

Your PRO/DATATRIEVE surroundings — ready domains, established collections, the current dictionary, available domains, procedures, record definitions, and so on. Use the SHOW command to get information on your PRO/DATATRIEVE environment.

### **ERASE**

A statement that deletes a record or records from an indexed file. You cannot erase records from a sequential file.

### **EXIT**

A command that returns you to the Main Menu when typed in response to the DTR> prompt and to the DTR> prompt when typed in response to any other prompt.

### **Field**

A named segment of a record that contains data. A field can consist of a very simple piece of data like a two-character state abbreviation or a combination of data items like a street number and name. Related fields make up a record.

### **Field Definition Clause**

A phrase that describes a characteristic of a field. ADT, for example, includes PIC, USAGE IS, EDIT-STRING, and QUERY-NAME clauses in record definitions to describe the field characteristics you provide in answer to its questions.

**FIND**

A statement that forms a special type of record stream called a collection and gives the collection the name `CURRENT`, in addition to any name you may specify. Each `FIND` statement replaces the old `CURRENT` collection with the new `CURRENT` collection it forms.

**FINISH**

A command that ends your access to a ready domain and releases any collections formed from the finished domain.

**Function Key**

A key that, when pressed, instructs `PRO/DATATRIEVE` to do something specific. `PRO/DATATRIEVE` function keys include `CANCEL`, `DO`, `EXIT`, `MAIN SCREEN`, and the `INTERRUPT/DO` key sequence. Other function keys beep when pressed.

**Guide Mode**

An interactive tutorial tool. Guide Mode helps you work step-by-step through a `PRO/DATATRIEVE` session and saves you typing time by providing additional prompts and information as you type in commands and statements. To use Guide Mode, type `SET GUIDE`.

**Index Key**

A field that `PRO/DATATRIEVE` uses to locate records in an indexed file. You cannot change the primary index key that identifies records in the file, but you can change alternate index keys.

**Indexed File**

A data file that contains records identified by an index key. You can delete records from an indexed file, but you cannot change the primary index key.

**Information Management**

A process you use to organize and manage information. `PRO/DATATRIEVE` is an information management tool that helps you organize information accurately and efficiently.

**Interactive language**

The way you and `PRO/DATATRIEVE` communicate. You tell `PRO/DATATRIEVE` what you want to do and `PRO/DATATRIEVE` does it. If you do not tell `PRO/DATATRIEVE` all it needs to know, `PRO/DATATRIEVE` asks you for more information.

## GLOSSARY

### **Keyword**

A word that has specific meaning for PRO/DATATRIEVE. Commands and statements consist of keywords like DEFINE, FIND, SET, and REPORT. In the text of this book, keywords are printed in uppercase letters. You cannot use PRO/DATATRIEVE keywords as the names of domains, records, procedures, tables, or fields.

### **MODIFY**

A statement that lets you change values stored in record fields. You cannot modify the primary index key field of records in an indexed file.

### **Name Clause**

A component of an RSE that tells PRO/DATATRIEVE what to name a record stream.

### **PIC**

A field definition clause that tells PRO/DATATRIEVE the type of data the field contains (numeric or character), how long the field can be, and how to format field values. PICTURE is a synonym for PIC.

### **Procedure**

A named sequence of commands and statements stored in your data dictionary. To create a procedure, use the DEFINE PROCEDURE command. To execute a procedure, type a colon (:) and the procedure name. PRO/DATATRIEVE executes the procedure as though you had just typed the commands and statements it contains.

### **PRINT**

A statement that displays records on your Professional screen or writes records and other values to a report file.

### **Print List Items**

Keywords, symbols, and field names that tell PRO/DATATRIEVE or the Report Writer where, what, and how to print record fields. Useful Report Writer print list items include SKIP, COL, SPACE, and (-). The way you combine print list options determines how the Report Writer formats detail and summary lines in the report.

### **Prompt**

A character or phrase that PRO/DATATRIEVE displays on your screen to let you know it is waiting to do your bidding. The DTR> command prompt indicates that you should type in commands and statements that

tell PRO/DATATRIEVE what to do next. Other types of prompts, like CON>, RW>, and Enter, let you know what PRO/DATATRIEVE expects from you.

**QUERY-HEADER**

A field definition clause that tells PRO/DATATRIEVE to use a substitute column header instead of the field name when displaying field values on your terminal or printing field values to a report. You cannot use query headers as query names.

**QUERY-NAME**

A field definition clause that defines a synonym for a field name, usually a shorter word. You can use query names anywhere you can use field names.

**READY**

A command that gets a PRO/DATATRIEVE domain ready for you to use. Depending on the access you specify (READ, WRITE, or MODIFY), you can read, modify, store, and erase records from a file associated with the domain.

**Record**

A group of related fields that PRO/DATATRIEVE treats as a unit and stores in a file. To create a record, use the DEFINE RECORD command.

**Record Definition**

A named structure that lists and names all the fields in a record and describes the type of data (number, letters, money, a percentage, and so on) that the fields contain. A record definition tells PRO/DATATRIEVE what pieces of data a record can contain.

**Record Selection Expression (RSE)**

A phrase that identifies the records you want to include in a record stream. An RSE tells PRO/DATATRIEVE where to find the records and may specify how many records (number clause), a name for the record stream (name clause), conditional characteristics for the records (condition clause), and a sort order (sort clause).

## GLOSSARY

### **Record Source**

A component of an RSE that tells PRO/DATATRIEVE the name of the domain or collection that contains the records you want to include in a record stream.

### **Record Stream**

A temporary group of related records formed by a record selection expression.

### **Relational Operator**

A symbol that compares the contents of a record field with a certain value. A relational operator states a condition that may be either true or false. PRO/DATATRIEVE relational operators include EQUAL (=), NOT-EQUAL, GREATER-THAN (>), LESS-THAN (<), LESS-EQUAL, BETWEEN, and CONTAINING.

### **RELEASE**

A command that releases a collection. A released collection no longer exists.

### **REPEAT**

A statement that lets you repeat another statement a specified number of times.

### **REPORT**

A statement that calls the PRO/DATATRIEVE Report Writer. PRO/DATATRIEVE prompts you for Report Writer statements with the RW> prompt. Use the END-REPORT statement to end the report.

### **Report Specification**

A series of Report Writer statements that tells PRO/DATATRIEVE what to include in the report and how to format the report. The simplest type of report specification includes a REPORT statement, a PRINT statement, and an END-REPORT statement.

### **Report Writer**

A component of PRO/DATATRIEVE that lets you create and print reports in a variety of formats. You can tell the Report Writer what data and records to include in the report, what to title the report, what headings to print at the top of the report, and what summary lines to include

in the report. You can also tell the Report Writer to perform mathematical operations and can include the results of these operations in the report.

**Restriction Clause**

A component of an RSE that tells PRO/DATATRIEVE how many records to include in a record stream.

**RSE**

See Record Selection Expression.

**SELECT**

A statement that identifies a record in a collection as the target of a subsequent MODIFY or ERASE statement. You can select a record only from a collection.

**Sequential File**

A data file that contains records stored in sequential order, next to each other. You cannot delete records from a sequential file, but you can modify or erase all record fields. An empty record in a sequential file continues to take up space on your disk or diskette.

**SET**

A command that lets you control your PRO/DATATRIEVE environment. SET DICTIONARY establishes a specified data dictionary as your current dictionary. SET GUIDE starts Guide Mode.

**SET COLUMNS-PAGE**

A Report Writer statement that allows you to control the width of a report. When you do not explicitly set the width of a report, the Report Writer uses an 80-character line.

**SET REPORT-NAME**

A Report Writer statement that specifies a title for a report.

**SHOW**

A command that tells PRO/DATATRIEVE to display information about your current data dictionary and PRO/DATATRIEVE environment. SHOW ALL displays all available information. To display specific information, specify what you want to display, as in SHOW RECORDS, SHOW DICTIONARY, and SHOW domain-name.

## GLOSSARY

### **SORT**

A statement that changes the order of records in a collection according to the sort key you specify.

### **SORTED BY**

A component of an RSE that tells PRO/DATATRIEVE how to order records in the record stream. If you do not include the SORTED BY clause in an RSE, PRO/DATATRIEVE sorts records in ascending order, by the primary index key field.

### **Sort Key**

A field used to sort a group of records. You specify a sort key in an RSE sort clause or in a SORT statement.

### **Statement**

One or more PRO/DATATRIEVE keywords, like FIND, MODIFY, STORE, and REPORT, for example, that you use to do something with or to data. You can automatically repeat a statement and you can execute a group of statements as a unit.

### **Statistical Functions**

Keywords, like COUNT, AVERAGE, and TOTAL, that tell the Report Writer or PRO/DATATRIEVE to perform a particular calculation.

### **STORE**

A statement that writes a record to a data file. You must ready the file's domain for WRITE access before you use the STORE statement.

### **Summary Lines**

Lines in a report that summarize information included in the report.

### **Top-level Field**

The first field in a record definition. All records have a top-level field that includes all other fields in the record. The top-level field always has the lowest field level number (usually 01) and a name.

### **TOTAL**

A statistical function that totals a group of values.

### **USAGE**

A field definition clause, like USAGE IS DATE, that identifies a field's data type and tells PRO/DATATRIEVE how to store values in the field.



# Index

---

- !, in procedures, 94
- \$, in EDIT-STRING clause, 31
- /, in SET REPORT-NAME statement, 81
- 9, in PIC clause, 30
- : command, 92
- <, *see LESS-THAN relational operator*
- =, *see EQUAL relational operator*
- >, *see GREATER-THAN relational operator*
- @ command, 29, 97
- | and ||, to concatenate strings, 86
  
- Abbreviations
  - in Guide Mode, 91
  - relational operators, 46
  - sort order keywords, 52
- Accessing domains, 32, 71
- ADT command, 27
- ADT, *see Application Design Tool*
- ALL
  - in an RSE, 43
  - in ERASE statement, 72
  - in MODIFY statement, 72
  - OF, 73
- AND Boolean operator, 45
  - example of, 49
- Application Design Tool, 6, 21, 91
  - getting help, 27
- ASC, 52
- ASCENDING sort order, 52
  
- AT BOTTOM OF REPORT statement, 87
- AVERAGE statistical function, 87
  
- BETWEEN relational operator, 46
- Boolean expressions
  - components of, 44
  - compound, 48
  - evaluating, 48
  - field names in, 44
  - in RSEs, 42
  - parentheses in, 50
  - simple, 45
- Boolean operators, 44, 48
- BT, 46
  
- CANCEL key, 16
- Changing
  - dictionaries, 15
  - index keys, 26
  - one field, 70, 73
  - records, 65
    - in a record stream, 73
    - in collections, 68, 72
  - sort order, 52
  - the selected record, 69
- COL formatting keyword, 84
- Collections, 7
  - creating, 55
  - CURRENT, 55
  - deleting all records in, 72
  - displaying

- information about, 57
  - records in, 57
- finishing, 62
- forming record streams from, 56
- including in reports, 78
- modifying all records in, 72
- naming, 55, 57
- releasing from memory, 59
- selecting records in, 66, 67
- sorting
  - with SORT, 61
  - with SORTED BY, 60
- Column headings
  - displaying, 37
  - suppressing in reports, 84
- Command files, 91
  - creating
    - with ADT, 29
    - with EXTRACT, 97
    - with PROSE, 98
  - executing, 29, 97
- Commands, 10
- Commas
  - in EDIT-STRING clause, 31
  - in USING clause, 88
- Comments, in procedures, 94
- CON prompt, 10, 41
- Concatenation symbols (| and ||), 86
- CONT, 46
- CONTAINING relational operator, 46
  - case with, 46
- Copying dictionary definitions, 97
- COUNT statistical function, 87
- Creating
  - a database with ADT, 21
  - command files
    - with ADT, 29
    - with EXTRACT, 97
    - with PROSE, 98
  - dictionaries, 15
  - named collections, 57
  - procedures, 92
  - report
    - detail lines, 84
    - files, 81
    - summary lines, 87
    - titles, 80
    - reports, 78
    - unnamed collections, 56
- CTRL/C, *see* INTERRUPT key
- CTRL/Z, *see* EXIT key
- CURRENT collection, 55
- Data definition commands, 29
- Date fields, 31
  - formatting, 24
- Decimal point
  - in EDIT-STRING clause, 31
  - in USING clause, 88
- DEFINE DICTIONARY command, 15
- DEFINE DOMAIN command, 29
- DEFINE FILE command, 29
- DEFINE PROCEDURE command, 92
- DEFINE RECORD command, 29
- Defining
  - a database with ADT, 21
  - dictionaries, 15
  - procedures, 92
- DELETE command, 93
- Deleting
  - all records in collections, 72
  - domain definitions, 93
  - procedure definitions, 93
  - record definitions, 93
  - records, 65
    - in a record stream, 73
    - in collections, 68
    - selected records, 71
- DESC, 52
- DESCENDING sort order, 52
- Detail lines, in reports, 83
- DFN prompt, 93
- Dictionaries, 11
  - changing, 15
  - creating, 15
  - displaying
    - current name, 14
    - object names, 11
  - editing definitions in, 97
  - naming, 15
- Displaying
  - collection information, 57
  - CURRENT collection, 57
  - current dictionary name, 14
  - dictionary information, 11
  - domain
    - definitions, 12
    - names, 12
  - fields, 37
  - information, 12, 37
  - procedure
    - definitions, 93
    - names, 93
  - readied domain names, 32
  - record
    - definitions, 12
    - fields, 79
    - names, 12

- records in a collection, 57
  - selected
    - record information, 68
    - records, 67
- DO key, 11, 16
- Dollar signs (\$)
  - in EDIT-STRING clause, 31
  - in USING clause, 88
- Domain definitions
  - creating with ADT, 21
  - deleting, 93
  - displaying, 12
- Domains, 5
  - accessing, 32, 71
  - defining with ADT, 21
  - displaying names of, 12
  - finishing, 38
  - naming, 25
  - readying, 32
- DTR prompt, 10
- Edit strings
  - in EDIT-STRING clause, 31
  - in USING clause, 87
- EDIT-STRING clause, 30, 31
- Editing dictionary definitions, 97
- END-PROCEDURE command, 93
- END-REPORT statement, 78
- Environment, 11
- EQ, 46
- EQUAL relational operator, 46
  - list with, 47
- ERASE statement, 65, 68
  - ALL in, 72
  - ALL OF in, 73
  - specifying an RSE in, 73
- Executing
  - command files, 29, 97
  - procedures, 92, 95
- EXIT command, 10
- EXIT key, 18
  - in Guide Mode, 92
  - to exit PRO/DATATRIEVE, 10
  - to leave ADT, 29
  - with MODIFY, 70
  - with REPEAT statement, 35
  - with STORE statement, 34
- Exiting
  - from ADT, 29
  - from Guide Mode, 91, 92
  - from PRO/DATATRIEVE, 10
- EXTRACT command, 97
- Field definition clauses, 30
- Field names
  - in Boolean expressions, 44
  - rules for, 23
- Fields, 4
  - creating blank, 35, 69
  - date, 24
  - describing characteristics, 30
  - determining characteristics, 23
  - displaying, 37, 79
  - identifying blank, 70
  - modifying selected, 70
  - naming, 23
  - ordering, 25
  - providing query names, 24
  - specifying
    - data type, 24
    - length, 30
    - size, 24
    - top-level, 79
- Files, 4
  - creating report, 81
  - defining with ADT, 21
  - indexed, 26
  - naming, 26
  - sequential, 26
- FIND statement, 55
- FINISH command, 38
  - with collections, 62, 63
- FIRST clause
  - in an RSE, 43
  - in SELECT statement, 66
- Formatting
  - date fields, 24, 31
  - keywords, 84
  - money fields, 31, 88
  - report
    - detail lines, 84
    - summary lines, 87
- Function keys, 16, 19
  - CANCEL, 16
  - DO, 11, 16
  - EXIT, 10, 18, 29, 34, 35, 70, 92
  - HELP, 17, 27, 91
  - INTERRUPT, 18, 92
  - MAIN SCREEN, 16, 92
- GE, 46
- GREATER-EQUAL relational operator, 46
- GREATER-THAN relational operator, 46
- GT, 46
- Guide Mode, 91
  - exiting, 18, 92

- getting help, 91
- HELP command, 17
- HELP key, 17
  - in Guide Mode, 91
  - with ADT, 27
- Hyphens
  - in domain names, 25
  - in field names, 23
  - in Report Writer statements, 84
- Index keys, 26
  - allowing duplicates, 26
  - alternate, 26
  - changing, 26
  - primary, 26
- Indexed files, 26
  - erasing records from, 65
- Installing PRO/DATATRIEVE, 9
- INTERRUPT key, 18
  - in Guide Mode, 92
- Keys
  - index, 26
  - sort, 51, 61
- Keywords, 10
  - formatting, 84
  - statistical function, 87
- LAST clause, in SELECT statement, 66
- LE, 46
- LESS-EQUAL relational operator, 46
- LESS-THAN relational operator, 46
- Letters
  - case with CONTAINING, 46
  - lowercase, 33
  - uppercase, 33
- Literals, 87
- LT, 46
- MAIN SCREEN key, 16
  - in Guide Mode, 92
- MODIFY access, 32, 71
- MODIFY statement, 65, 68
  - ALL in, 72
  - ALL OF in, 73
  - specifying an RSE in, 73
- Money fields
  - formatting with EDIT-STRING, 31
  - formatting with USING, 88
- Multiline
  - record selection expressions, 41
  - report titles, 81
  - statements, 41
- Naming
  - collections, 55, 57
  - command files, 97
  - dictionaries, 15
  - domains, 25
  - fields, 23
  - files, 26
  - procedures, 92
  - records, 25
- NE, 46
- NEXT clause, in SELECT statement, 66
- NOT-EQUAL relational operator, 46
- Operators
  - Boolean, 44, 48
  - relational, 44, 45
- OR Boolean operator, 45
  - example of, 49
- Parentheses
  - around hyphens, 84
  - in expressions, 50
- PIC clause, 30
- PRINT statement, 12, 37
  - ALL in, 57
  - CURRENT in, 57
  - Report Writer, 78, 84
- Printing information
  - in a report, 78
  - on the screen, 12, 37
- Procedure definitions, 91
  - comments in, 94
  - creating, 92
  - deleting, 93
  - displaying, 93
  - ending, 93
  - executing, 92
  - naming, 92
- Prompts, 10
  - CON, 10, 41
  - DFN, 93
  - DTR, 10
  - RW, 10, 79
- Query names, 24, 30
- QUERY-NAME clause, 30
- Quotation marks
  - around literals, 87
  - in RSEs, 40
  - in SET REPORT-NAME statement, 81
- READ access, 32
- READY command, 12, 32

- READY Command Access Types and Privileges, 32t
- Record definitions, 5
  - creating with ADT, 21
  - deleting, 93
  - displaying, 12
- Record selection expressions
  - ALL clause in, 43
  - Boolean expression in, 42
  - components of, 41
  - creating, 39
  - defined, 6
  - FIRST clause in, 43
  - forming, 41
  - in ERASE statement, 73
  - in MODIFY statement, 73
  - in REPORT statement, 78
  - multiline, 41
  - record source, 41
  - SORTED BY clause in, 51
  - WITH clause in, 44
- Record streams, 40
  - deleting all records in, 73
  - forming
    - from collections, 56
    - from domains, 41
  - including in reports, 78
  - modifying all records in, 73
  - specifying source of, 42
- Records, 4
  - creating collections of, 55
  - defining with ADT, 21
  - deleting, 65
    - in collections, 68, 72
    - in record streams, 73
  - selected, 71
  - displaying, 12, 37
    - names of, 12
    - selected, 57
  - modifying, 65
    - in collections, 68, 72
    - in record streams, 73
  - selected, 69
  - naming with ADT, 25
  - selecting, 66
  - sorting
    - with SORT, 61
    - with SORTED BY, 51, 60
  - storing, 13, 33, 35
- Relational Operators, 46t
- Relational operators, 44, 45
  - abbreviations, 46
  - symbols, 46
- RELEASE command, 59, 63
- REPEAT statement, 35
- REPORT statement, 78
  - ON in, 81
  - record selection expressions in, 78
- Report Writer, 7, 78
  - AT BOTTOM OF REPORT statement, 87
  - END-REPORT statement, 78
  - PRINT statement, 78, 84
  - prompt, 10
  - REPORT statement, 78
  - SET COLUMNS-PAGE statement, 82
  - SET REPORT-NAME statement, 80
- Reports
  - averaging values in, 87
  - counting records in, 87
  - creating, 78
  - formatting
    - detail lines, 84
    - page width, 82
    - summary lines, 87
  - titling, 80
  - totaling values in, 87
  - writing to a file, 81
- RETURN key, 11, 16
- RSE, *see Record selection expressions*
- RW prompt, 10, 79
- S, in PIC clause, 30
- SELECT statement, 57, 65, 66
- SELECT Statement Forms and Effects, 66t
- Selected records
  - deleting, 71
  - displaying, 67
    - information about, 68
  - modifying, 69
- Sequential files, 26
  - deleting data from, 65
- SET COLUMNS-PAGE statement, 82
- SET DICTIONARY command, 15
- SET GUIDE command, 91
- SET REPORT-NAME statement, 80
- SHOW COLLECTIONS command, 57
- SHOW command, 11, 57
- SHOW CURRENT command, 57
- SHOW DICTIONARY command, 14
- SHOW DOMAINS command, 12
- SHOW FIELDS command, 79
- SHOW PROCEDURES command, 93
- SHOW READY command, 32
- SHOW RECORDS command, 12
- SKIP formatting keyword, 84
- Sort keys

- in SORT statement, 61
- in SORTED BY clause, 51
- specifying multiple, 52
- Sort order
  - ASCENDING, 52
  - changing, 52
  - DESCENDING, 52
- SORT statement, 61
- SORTED BY clause, 51
  - sort order, 52
  - with collections, 60
- Sorting records
  - by more than one key, 52
  - by one key, 51
  - in collections, 60, 61
- SPACE formatting keyword, 84
- Statements
  - defined, 33
  - function of, 10
  - multiline, 41
  - repeating, 35
- Statistical functions, 87
- STORE statement, 13, 33
- Summary lines, in reports, 83, 87
- Suppressing column headings, 84
- TAB key, with MODIFY statement, 70
- Titling reports, 80
- TOTAL statistical function, 87
- Underscores, *see Hyphens*
- USAGE clause, 30, 31
- USING clause, 87
- V, in PIC clause, 30
- Values in Boolean expressions, 44
- WITH clause, 44
- WRITE access, 32, 71
- X, in PIC clause, 30

DECLIT AA PRO V439A

Professional 300 series  
DATATRIEVE for beginners

DECLIT AA PRO V439A

Professional 300 series  
DATATRIEVE for beginners




**SHREWSBURY LIBRARY**

**Digital Equipment Corporation**

**333 South Street SHR1-3/G18**

**Shrewsbury, MA 01545**

**(DTN) 237-3271**

digital



\*098859\*

Printed in U.S.A.  
AA-V439A-TH



# Professional™

PROSE PLUS  
User's Guide

AA  
PRO  
R755A

digital

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The specifications and drawings, herein, are the property of Digital Equipment Corporation and shall not be reproduced or copied or used in whole or in part as the basis for the manufacture or sale of items without written permission.

Copyright © 1984 by Digital Equipment Corporation  
All Rights Reserved

The following are trademarks of Digital Equipment Corporation:

CTI BUS	MASSBUS	PROSE PLUS
DEC	PDP	Rainbow
DECmate	P/OS	RSTS
DECsystem-10	PRO/BASIC	RSX
DECSYSTEM-20	PRO/Communications	Tool Kit
DECUS	Professional	UNIBUS
DECwriter	PRO/FMS	VAX
DIBOL	PRO/RMS	VMS
<b>digital</b>	PRO/Videotex	VT
	PROSE	Work Processor

94-003/049/28

C in cat

## Before You Start

---

### Who Should Read This Book

This book tells you how to use PROSE PLUS, a combination text and graphics editor product that allows you to create documents on your Professional with text enhancements and detailed custom graphics. The graphics produced using this application can be stored and recalled at any time for editing or insertion into other documents for creating charts and diagrams for presentation. PROSE PLUS uses an enhanced version of DIGITAL's popular PROSE text editor. While you need no experience in programming to use PROSE PLUS, you should be familiar with the general operation of your Professional, including the keyboard, function keys, and menus. These topics are covered in your *Professional 300 Series User's Guide for Hard Disk System*. This book covers operating instructions for the Professional only.

### Information in This Book

This guide is written for PROSE PLUS users of Professional 350 hard disk systems, and is intended to supplement the information contained in other Professional 300 Series user's guides that are mentioned elsewhere in this book. PROSE PLUS requires the installation of P/OS version 1.7, or later. A summary of each chapter follows.

- Chapter 1 introduces PROSE PLUS, tells how to set up your hardware and install PROSE PLUS, and provides an overview of the user interface.
- Chapter 2 discusses PROSE PLUS concepts, keyboard definitions, types of displays, and terms that apply throughout the manual and during the use of the product.

- Chapter 3 explains how to use the PROSE PLUS text editor for creating documents.
- Chapter 4 describes the PROSE PLUS graphics editor and shows how to create and edit graphics frames for inclusion in your documents.
- Appendix A is a glossary of terms common to the Professional 300 series personal computers, and the PROSE PLUS application.
- Appendix B lists the differences between using PROSE PLUS on version 1.7 and 2.0 of P/OS.

# Contents

---

## **CHAPTER 1      GETTING STARTED WITH PROSE PLUS**

---

Required Hardware and Software .....	1
How to Install and Run PROSE PLUS.....	2
Running PROSE PLUS.....	2

## **CHAPTER 2      PROSE PLUS CONCEPTS**

---

PROSE PLUS Menus .....	5
The PROSE PLUS Main Menu .....	6
Creating a New Document.....	7
Editing an Existing Document .....	7
Printing a document .....	8
Viewing a document .....	8
The Status Line .....	9
Text Editor Status .....	9
Graphics Editor Status .....	10

## **CHAPTER 3      THE PROSE PLUS TEXT EDITOR**

---

Elementary PROSE PLUS .....	13
Moving the cursor .....	14
Typing in Overtyping Mode .....	15
Deleting characters with the delete key .....	15
Inserting text with INSERT HERE.....	16
Deleting segments of text using REMOVE .....	17

The EXIT Key.....	17
Save.....	18
Quit.....	18
Save without formatting.....	19
The MAIN SCREEN Key.....	19
More Advanced PROSE PLUS.....	20
Locating Text Segments with FIND.....	20
PROSE PLUS Additional Options Menus.....	20
PROSE PLUS Attributes Menu.....	21
Bold.....	22
Underline.....	22
Clear attributes.....	23
Uppercase.....	23
Lowercase.....	23
PROSE PLUS Edit Menu.....	24
Relocate Text.....	24
Pictures.....	25
Top of document.....	26
Bottom of document.....	26
Get document.....	26
Create work area.....	26
Switch work areas.....	27
Headers.....	27
PROSE PLUS Personalize Menu.....	28
Columns 80/132.....	28
Define a Key.....	29
List defined keys.....	31
Set tab stops.....	31
Status Line ON/OFF.....	32
US Units/Metric.....	32
PROSE PLUS Format Menu.....	33
Center Line.....	33
Fill Text.....	33
Margin Change/Word Wrap.....	34
New page.....	35
Paginate document.....	37
Layout page.....	38

## CHAPTER 4 THE PROSE PLUS GRAPHICS EDITOR

---

Getting Started with the Graphics Editor .....	42
Help .....	43
The Graphics Editing Screen .....	44
Moving the Cursor .....	45
Find.....	45
Drag .....	47
Creating New Pictures .....	48
Drawing Lines .....	50
Drawing Rectangles .....	50
Entering Text.....	51
Drawing Arcs or Circles .....	52
Changing an Existing Object .....	52
Moving an Existing Object.....	52
Changing the Characteristics of an Existing Object .....	53
Exiting Back to the Text Editor .....	53
Additional Functions.....	54
DO .....	55
CANCEL.....	55
SELECT .....	55
REMOVE.....	56
Additional Options Menu.....	56
Refresh screen .....	56
Erase picture.....	56
Undo last remove .....	57
Save.....	57
Overlay.....	58
Quit .....	58
Toggle page scroll.....	59
Editing Old Pictures .....	59
Pictures Created Using PROSE PLUS .....	60
Picture Created by Another Application and Printed Using PROSE PLUS	63

**APPENDIX A      DIFFERENCES BETWEEN P/OS V1.7 AND V2.0**

---

Appendix A ..... 67

**GLOSSARY**

---

Glossary ..... 71

**INDEX**

---

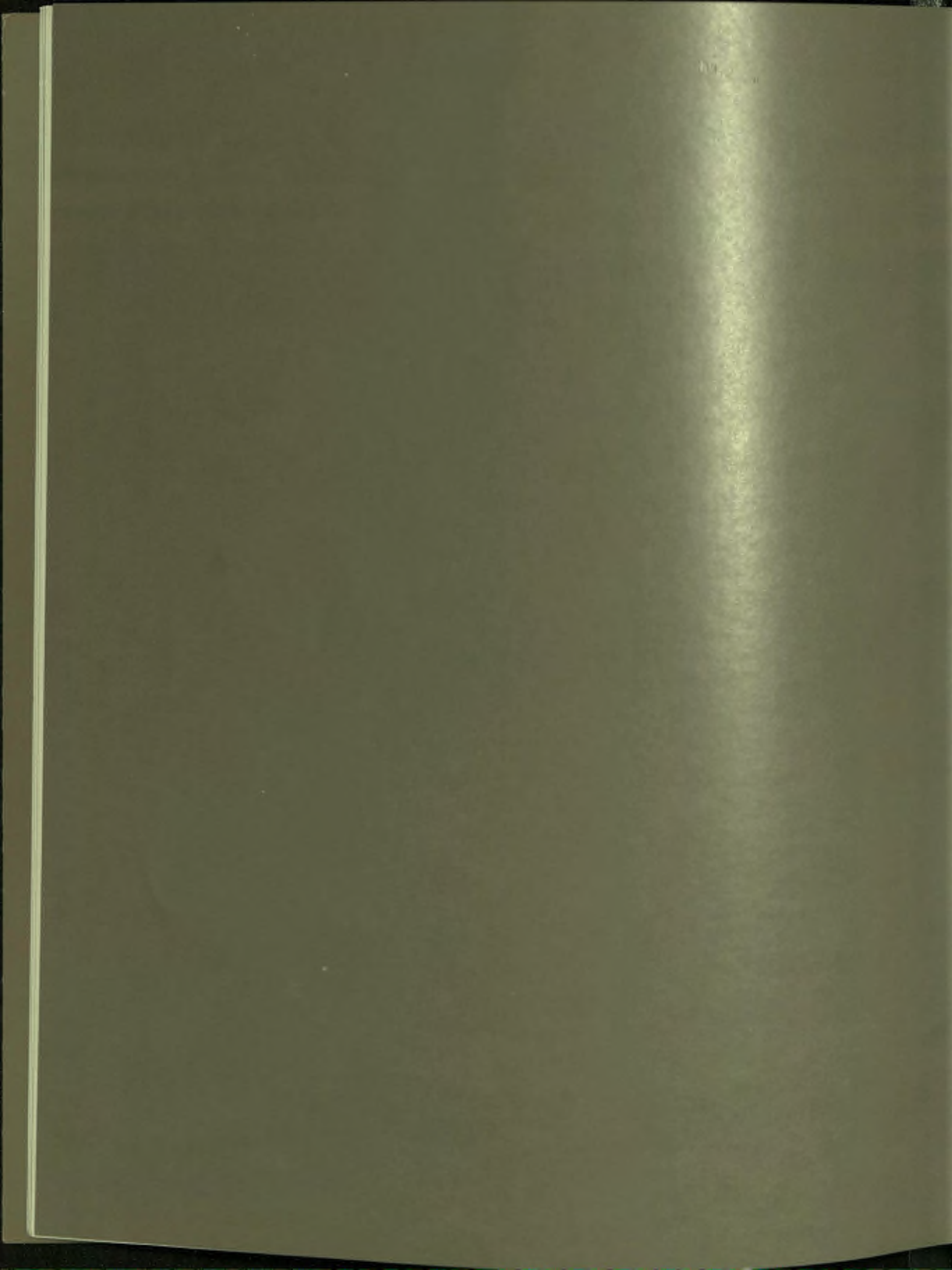
Index ..... 77



# 1

---

Getting Started with  
PROSE PLUS



## Chapter 1

---

### Getting Started with PROSE PLUS

PROSE PLUS is a software application designed to take advantage of the powerful capabilities of the Professional 300 series personal mini-computers. Using your Professional with PROSE PLUS you can easily create a document containing text and highly detailed graphics normally associated with more elaborate and complex graphics editing systems. In conjunction with a DIGITAL dot matrix printer (such as the LA50 or LA100), your Professional can function as a multi-purpose text processor, not only capable of standalone text editing, but also graphics production for reports, letters, and charts.

Using the PROSE PLUS application you can:

- Perform standalone document processing utilizing many features that you may already be familiar with
- Create detailed pictures using the graphics features
- Maintain a directory of graphics files for use in other PROSE PLUS documents

#### **REQUIRED HARDWARE AND SOFTWARE**

Before installing PROSE PLUS, make sure your Professional is equipped with the following DIGITAL hardware and software:

- P/OS version 1.7 or 2.0
- DIGITAL LA50 or LA100 printer for text and graphics hardcopy, or and an LQP02 for letter quality text hardcopy only

## HOW TO INSTALL AND RUN PROSE PLUS

The first step toward using PROSE PLUS is to install it. If you have installed other applications on your system, then you are familiar with the installation procedures described in your *Professional 300 Series User's Guide for Hard Disk System*. Follow those procedures to install PROSE PLUS.

### Running PROSE PLUS

Once you have successfully installed PROSE PLUS, you can run it by selecting it from the P/OS application menu on which it was installed.<sup>1</sup> Move the pointer to the PROSE PLUS application name listed on the menu and press DO.

---

<sup>1</sup>Since *application* names can be changed at install time, your PROSE PLUS may be identified by a different name.

# 2

---

PROSE PLUS Concepts



## Chapter 2

---

### PROSE PLUS Concepts

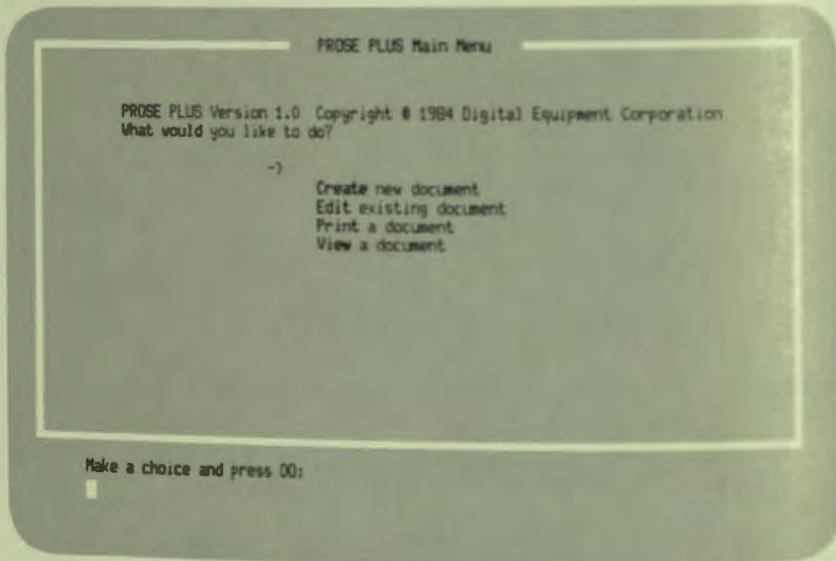
This chapter discusses the basic concepts of PROSE PLUS and introduces you to the menu structure and displays used when creating documents. It is recommended that you read this chapter and understand it before proceeding to the chapters on Text Editing and Graphics Editing.

#### **PROSE PLUS MENUS**

PROSE PLUS uses two different types of menus. The first is typical of Professional 300 Series applications where you are given several options to choose from on a full screen menu. A menu item is chosen either by moving the pointer alongside the item using the UP and DOWN arrow keys and pressing DO, or by typing in the bolded option keyword and pressing DO. The second type of menu appears while a text or graphics editing screen is displayed. The menu appears as a small box that “pops” onto the screen. Choosing an item from this type of menu is similar to the previous method. The only difference is that the pointer is replaced by *reverse video* to highlight the selected menu option.

### The PROSE PLUS Main Menu

The main menu for PROSE PLUS gives you the option of creating a new document or editing an existing one. In addition, you can select an existing document to view or print. The PROSE PLUS Menu looks like this:

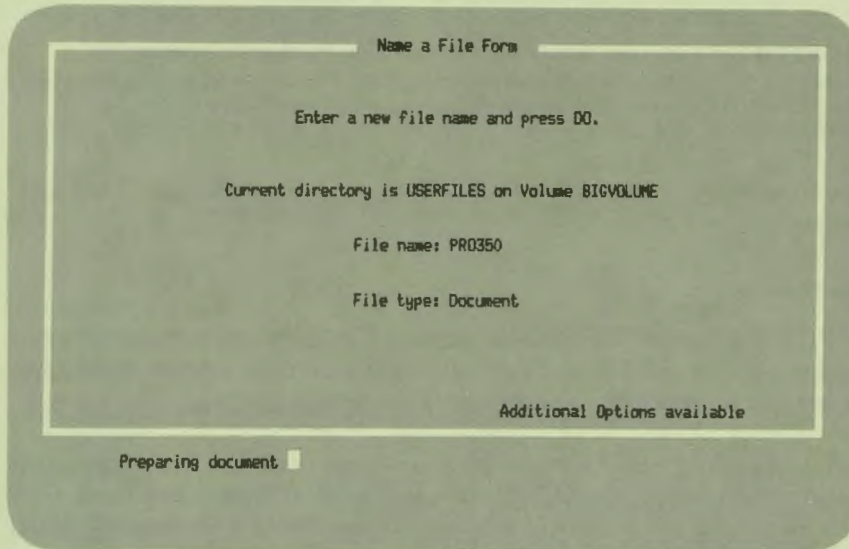


If you want to create a memo or other new text, choose "Create new document."  
If you want to alter an existing document, choose "Edit existing document."



### Creating a New Document

If you choose “Create new document” from the PROSE PLUS Menu, the screen looks like this:



The Name a File Form is for you to fill out with a name for the document you are creating. Since a document always consists of a .DOC type file, and can also have one or more picture files, it is necessary to ensure that all files related to a particular document are retrievable from the **same volume** and *directory*. Additional information on file specifications is contained in the *Professional 300 Series User's Guide for Hard Disk System*.

### Editing an Existing Document

If you choose “Edit existing document” from the PROSE PLUS Menu, a list of the document files in your current volume and directory appears. Pressing the NEXT SCREEN key reveals any additional document names that could not fit on the first screen. If the document you wish to edit is listed, move the pointer down next to it and press DO.

To choose a document from a different directory, press **ADDTNL OPTIONS**. Choose the option "Choose a different directory/volume" and press **DO**. When the Directory Selection Menu appears, choose the directory containing the document you want and press **DO**. When the File Selection Menu appears, choose the document you want to edit and press **DO**.

**WARNING:** If a document references any pictures, the document and picture files must be contained together in the same directory. This directory must be your current directory. The current directory can be changed using P/OS File Services.

When the document is displayed on your screen, you can add or modify the text and graphics!

### **Printing a document**

Choose "Print a document" to print an existing document on your local printer. You must have either a DIGITAL LA50 or LA100 dot matrix printer attached to your system box to print out any documents containing pictures.

A DIGITAL LQP02 can be used if your requirements are limited to letter-quality text **only**. An LQP02 will not print pictures contained in a document. Wait until the printer is not busy to use "Print a document." When printing begins, the PROSE PLUS Main Menu reappears. At this point, you can select another operation to perform while the document is printing.

### **Viewing a document**

Choose "View a document" to display an entire document on your screen. This feature allows you to scan a document's text and graphics for any errors before printing it out. Pictures greater than 24 lines in height are automatically scaled down to fit on the screen for full viewing. However, they will print out at full size.

---

<sup>1</sup>You may see some blinking characters in the text display. These represent control characters. Refer to documents describing the *system* or application that created the document for more information about control characters.

## THE STATUS LINE

The status line appears in reverse video at the bottom of the text and graphics editing screens. It contains information about the current editing operation, and any additional options that are available at the time. The status line does **not** appear in the printed document.

### Text Editor Status

The status line that appears while you are in the text editor looks like this:

This is the PROSE PLUS text editor Status Line.

```
Now doing: Overtyp e                               Margins:  1, 72 Not wrapped
┌─── Attributes ──┐ ┌─── Edit ───┐ ┌─── Personalize ─┐ ┌─── Format ───┐
```

The information contained in it includes the following:

- |                              |  |
|------------------------------|--|
| <b>Now doing:</b>            | Tells you what you are doing at the moment. If you are entering text, the word "overtyp e" appears, to show that you are in <i>overtyp e mode</i> . (You are in overtype mode unless you press <b>INSERT HERE</b> .) If you perform a function, such as find a word or insert text, the function name appears instead of the word "overtyp e." When using some additional options, other messages appear in this area to assist you.   |
| <b>Margins:</b>              | Identifies the left and right margin settings by their column number for the paragraph that the cursor is in. These values are determined by the default settings, or by the settings you specify when embedding rulers in the text. The values change whenever an embedded ruler is encountered by the cursor. In the example above, the left margin is at column 8 and the right margin at column 72. In addition, the type of line format is identified as <i>word wrapped</i> , <i>not wrapped</i> , or <i>justified</i> . |
| <b>Active Function Keys:</b> | The bottom line identifies function keys F11 through F14 that are currently active, and the options menus or functions that they represent.  |

**Graphics Editor Status**

The status line that appears while you are in the graphics editor looks like this:

This is the PROSE PLUS *graphics editor* Status Line.

File: TR.GID

DRAG: Off PAGE: Not Scrolled

└─ Addnl Options ─┘

The information contained in it varies with the operation being performed. This information includes the following:

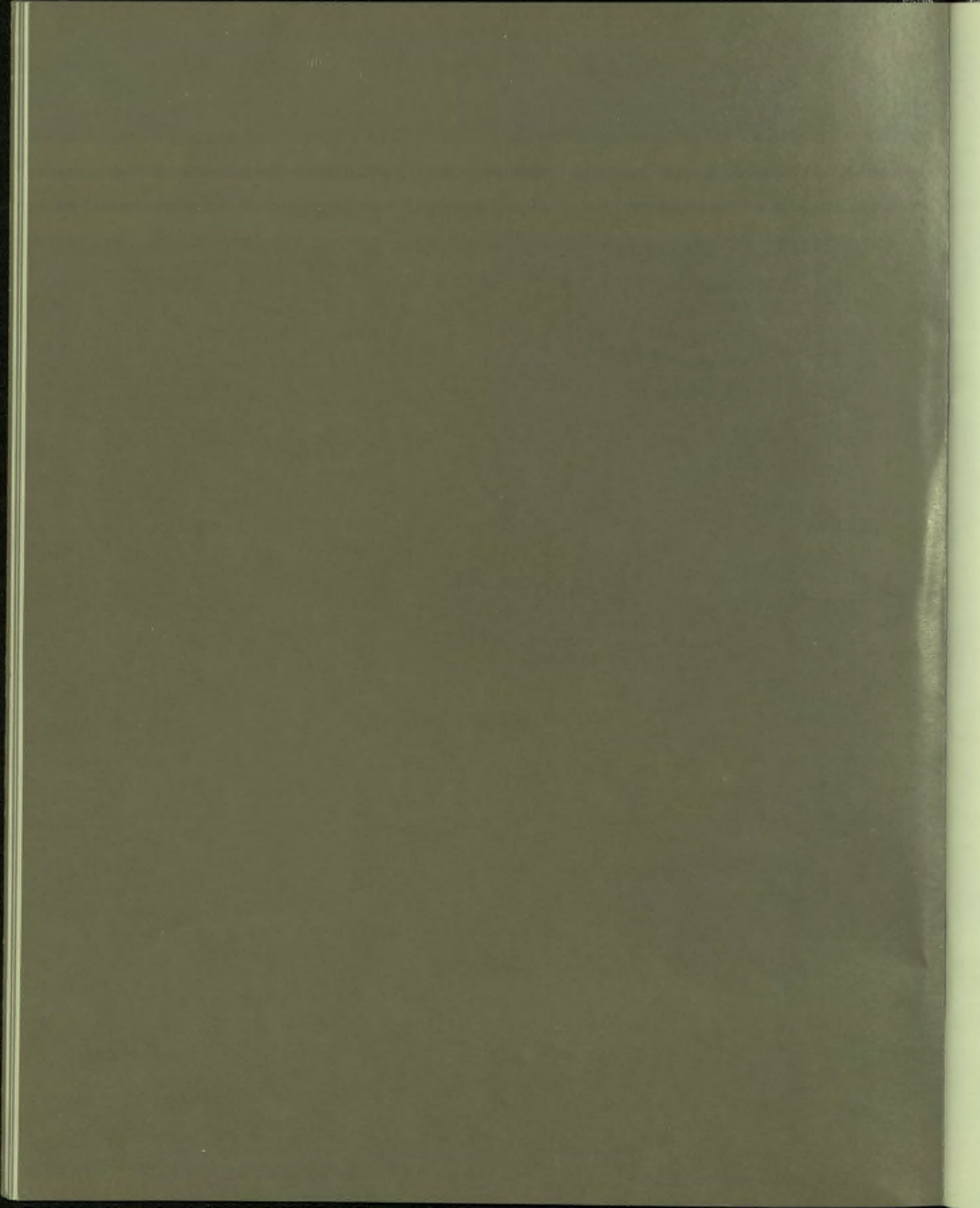
- |                       |  |
|-----------------------|--|
| File:                 | Tells you the name of the picture file you are working with. This information disappears whenever you press <b>INSERT HERE</b> , <b>SELECT</b> , and other keys.   |
| Now doing:            | Identifies the object type you are drawing on the screen. This information appears when you press <b>INSERT HERE</b> , <b>SELECT</b> , and other keys.   |
| Drag:                 | Tells you whether or not the drag feature is active. <b>Drag</b> is automatically <b>off</b> when inserting objects. <b>Drag</b> can be toggled on or off using the F11 key.   |
| Page:                 | Tells you whether the page will scroll when the cursor encounters the top or bottom edges of the screen. The default setting is <i>Scrolled</i> . A setting of <i>Not scrolled</i> indicates that the display will not scroll. |
| Active Function Keys: | The bottom line identifies function keys <b>F11</b> through <b>F14</b> that are currently active.  |

The lower portion of the status line describes additional features or option menus that can be displayed by pressing the appropriate function key. Function keys **F11** through **F14** are active at different times during graphics editing. The status line keeps you informed as to which features are available.

# 3

---

The PROSE PLUS Text  
Editor



## Chapter 3

---

### The PROSE PLUS Text Editor

Using the PROSE PLUS text editor, you can prepare memos, reports, programs, and other kinds of text. You can create, edit, move, replace, and format text using the *editing keypad* and certain *function keys* on your keyboard. A series of “pop-up” menus make it easy for you to select a wide variety of features including pictures, embedded rulers, margin wrap, line justification, text enhancements, and automatic pagination to name a few.

With PROSE PLUS, the text you prepare appears on the screen in the same format it has when printed on a printer. In addition, the screen shows where the page breaks are for printing. This part of the screen does not print out.

The information in this chapter is divided into two sections:

- Elementary text editing—all you need to know to get an editing job done
- More advanced text editing—additional functions that make editing easier and allow you to change your text format

#### **ELEMENTARY PROSE PLUS**

When you first display the PROSE PLUS text editing screen for creating a new document, the left margin is at column 1 and the right margin is at column 72. These are the *default* margins, and are identified in the status line at the bottom of the screen. They remain at these settings unless you change them using the “Margin change/*word wrap*” additional option. The Margin Change/Word Wrap section explains how to change margins.

With PROSE PLUS, the text you prepare appears on the screen in the same format it has when printed on a printer. In addition, the screen shows where the page breaks are for printing. This part of the screen does not print out.

The information in this chapter is divided into two sections:

Elementary text editing -- all you need to know to get an editing job done

More advanced text editing -- additional functions that make editing easier and allow you to change your text format

#### Elementary PROSE PLUS

When you first display the PROSE PLUS text editing screen for creating a new document, the left margin is at column 1 and the right margin is at column 72. These are the default margins, and are identified in the status line at the bottom

```
Now doing: Overtyping           Margins: 10, 70 Justified
┌── Attributes ─┘ ┌── Edit ─┘ ┌── Personalize ─┘ ┌── Format ─┘
```

You should always separate paragraphs with a blank line. PROSE PLUS recognizes a blank line as the space between two paragraphs. When you press the RETURN key, PROSE PLUS moves the cursor to the left margin of the next line down. If you press RETURN a second time, a blank line will occur indicating the beginning of a new paragraph.

In PROSE PLUS, the TAB key by default moves the cursor to the right in increments of eight spaces or columns, starting at column 9. For example, pressing the TAB key while the cursor is in a column from 1 to 8 moves the cursor to column 9. PROSE PLUS also treats the left margin as a tab stop. To change your tab stops, use the Personalize Menu described in this chapter. Tab settings you may have made with Set-up Menu A do not affect the TAB key when you are in PROSE PLUS.

Any time you attempt to do something that isn't possible, PROSE PLUS displays a message on the bottom line of your screen. Press RESUME to continue after a message appears or press HELP for more information.

#### Moving the cursor

The cursor marks the spot where the next character you type will appear. To change any text you have already entered, you first have to move the cursor to the text to be changed.



You move the cursor around in your text by pressing the arrow keys. Each time you press an arrow key, the cursor moves one character or one line in the direction of the arrow. Holding down an arrow key causes the cursor to move rapidly in the direction of the arrow. With a little practice, you will be able to move the cursor to the spot you want by holding down an arrow key for the appropriate length of time.

While in overtyping mode (see section on status line) the RETURN key moves the cursor to the left margin of the next line down without affecting the text. To move the cursor through more than one screen of text, press PREV SCREEN to move backward and NEXT SCREEN to move forward. If there is not enough text to allow this, the cursor will move to either the beginning (for PREV SCREEN) or the end (for NEXT SCREEN) of the text.

You can move the cursor anywhere on the screen. However, you cannot insert text beyond your right margin in a word wrapped or justified paragraph, or on the status line.

### Typing in Overtyping Mode

In overtyping mode—which is in effect unless you press INSERT HERE—you can enter text normally or replace text by typing over it. You do not have to delete the original text. For example,

1. Move the cursor to the spot where you want to overtype:


This is some text to practice inserting in overtyping mode.


2. Type in the new text. It replaces the existing text:

This is some text to see what happens when I type over other text.

You can replace text with spaces by overtyping with the space bar.

### Deleting characters with the delete key

If you make typing errors while entering your text, use the delete key, labeled . This key erases the character to the left of the cursor. You can hold down the delete key to delete a series of characters. The delete key works like a space bar only in the reverse direction.

The following is an example of using the  key:

1. **Before delete:**

I want to delete some characters I just typed rong.

2. Press  14 times.

**After delete:**

I want to delete some characters I just

### Inserting text with INSERT HERE

When you want to insert new text into existing text without erasing the original text, use the INSERT HERE key. To insert new text into existing text:

1. Move the cursor to the spot where you want to insert the text.

For example,

Let's see how I insert new text into existing text.

2. Press INSERT HERE. The status line changes indicating that you are now "Inserting text." Any existing text from the cursor to the end of the line moves temporarily to the next line, leaving an open space.

For example,

Let's see how I  
insert text into existing text.

3. Type in the text you want to insert and include any spaces you need.

For example,

Let's see how I can  
insert text into existing text.


Notice that the newly inserted text is in reverse video.

4. When you finish inserting the new text, press DO. The open space closes up, and the inserted text is no longer in reverse video. If the paragraph is word wrapped, the rest of it "fills" so that there are no unnecessary spaces. For example,

Let's see how I can insert new text into existing text.

When filling existing text, PROSE PLUS leaves two spaces after any period, colon, exclamation point, or question mark that is followed by a space and an uppercase letter.

When inserting text in a paragraph that is not word wrapped, press RETURN after inserting to move the cursor to a new line. In this case, when you press DO, the open space created when you pressed INSERT HERE closes up, but the rest of the paragraph remains the same as before.

While using the insert function, you cannot use the arrow, PREV SCREEN, NEXT SCREEN, or FIND keys to move the cursor around in your text. You can use the  key, but only to delete text just inserted.

### Deleting segments of text using REMOVE

To remove parts of text you have already created, use the REMOVE key. Here is how you would delete a text segment:

1. Move the cursor to one end of the segment to be deleted; it does not matter which end. For example,

```
This is some text to delete. This is some more text, but I
do not want to delete this.
```

2. Press REMOVE.

3. Move the cursor to the other end of the segment to be deleted. The segment can be any length. For example,

```
This is some text to delete. This is some more text, but I
do not want to delete this.
```

The segment becomes highlighted in reverse video as you move the cursor. (Press CANCEL to undo the selection.)

4. Press DO. The selected text disappears. For example,

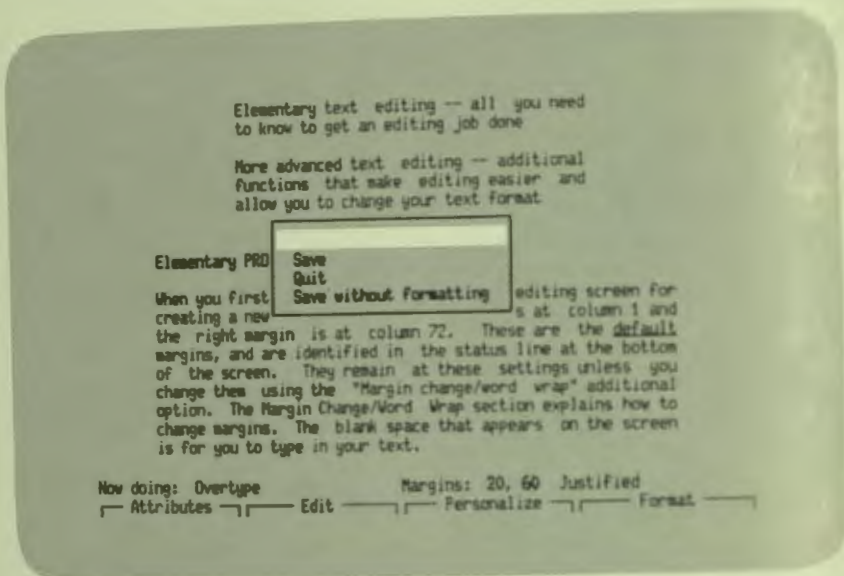
```
This is some more text, but I do not want to
delete this.
```

If the paragraph is word wrapped, any open spaces created by the deletion close up after you press DO, and the rest of the paragraph fills. If the paragraph is not word wrapped, the open space closes up, but the rest of the text is not changed.

### The EXIT Key

You can exit PROSE PLUS using either the EXIT key or the MAIN SCREEN key.

If you press EXIT, the Exit Menu appears with three exit options:



### Save

Choose "Save" to store your document back to disk. All margin, word wrap settings, attributes, and picture references are automatically saved along with the document.

If you are working on a previously created document, choosing "Save" creates a new version of your document. *P/OS* saves the previous version as well. You should delete old versions you no longer need so that you have more free space. You can use "Clean up old file versions" from the File Services Menu to delete all the old versions of a document.

**NOTE:** If you have pictures referenced in the document, both the document and picture files must be saved to the same directory.

### Quit

When you choose "Quit," you lose the results of the work you did in the text editor. However, changes to any pictures remain in separate files that you saved while using the graphics editor. If this was the first session creating the document, you lose everything you entered in the text editor except the picture files.

If you were working on an existing document, you lose only the changes and additions made during that session.

This is useful:

- If you are experimenting with PROSE PLUS to see how it works and do not want to save what you typed.
- If you made a lot of mistakes in your text and want to start again.

When you choose “Quit,” the following question appears:

Do you really want to discard edits from this session?  
(YES or NO)

Type YES (using either uppercase or lowercase letters) and press DO to leave this session without saving changes and return to the PROSE PLUS Main Menu.

Type NO and press DO (or just press DO) to remain in the editor without losing your changes.

#### **Save without formatting**

Choose “Save without formatting” to save the document without saving the margin, word wrap settings, attributes, and picture references. If you choose this exit option, the next time you edit the document it will not appear as you initially formatted it, nor will the picture references appear. If you try to edit it again, you will get unexpected results.

The main reason for removing these settings is to allow you to use *PRO/Communications* to transfer the document to a computer that does not use PROSE PLUS. Or, if you want to use an application that does not interpret the internal setting indications in the same way as PROSE PLUS, you would want to remove the settings.

#### **The MAIN SCREEN Key**

You can also exit PROSE PLUS by pressing the MAIN SCREEN key. When you press MAIN SCREEN, the document you are working on is automatically saved along with all formatting and picture references, and returns you to the P/OS applications menu containing PROSE PLUS.

## MORE ADVANCED PROSE PLUS

After you have used PROSE PLUS for a while, you may want to use its full potential. For example, you may want to move text around without retyping. Or, you may want to search for a particular word without moving the cursor with the arrow keys. You may want to tell PROSE PLUS exactly where to end each printed page in your document.

### Locating Text Segments with FIND

Use the **FIND** key to place the cursor on the first character of a specific text segment. The **FIND** key is useful when you know what text segment you want to change, but not where it is in the document. You can also look for repeated instances of the same segment.

Since PROSE PLUS moves **forward** through the document to the next instance of the text you typed, you may want to place the cursor at the beginning of the document beforehand by using the **PREV SCREEN** key or choose the “Top of document” option from the Edit Menu.

When you press the **FIND** key, the following *prompt* appears:

Enter the text to find and press DO:

Type in the word or group of characters you want to find and press **DO**. If PROSE PLUS cannot find the text, the following message appears:

Unable to locate text

PROSE PLUS can only look for a text segment that occurs on one line. If, for example, the first word of a two-word segment is at the end of a line, PROSE PLUS will display the message “Unable to locate text.” Press **RESUME** to continue after this message appears.

To search for the next instance of the same segment, press **DO** each time the “Enter the text to find ...” prompt appears. It is not necessary to reenter the text segment each time.

### PROSE PLUS Additional Options Menu

In addition to the PROSE PLUS functions performed with the special function keys, there are many PROSE PLUS editing capabilities available from the four PROSE PLUS Additional Options Menus. When entering or editing text with PROSE PLUS, press **F11**, **F12**, **F13**, or **ADDTNL OPTIONS** to display the menu which lists the option you want to use. The menu will appear at the top or the

bottom of your screen, depending on the position of the cursor at the time that you press the key for that menu. This enables you to see your work while you choose from the menu. The following table shows the options available on these menus.

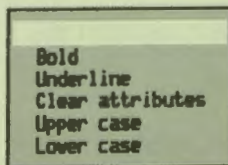
**Table**  
**PROSE PLUS Additional Options Menus**

<b>F11</b> <b>Attributes</b>	<b>F12</b> <b>Edit</b>	<b>F13</b> <b>Personalize</b>	<b>F14</b> <b>Format</b>
<b>Bold</b>	Relocate text	Columns 80/132	Center line
<b>Underline</b>	Pictures	Define a key	Fill text
<b>Clear attributes</b>	Top of document	List defined keys	Margin change/word wrap
<b>Upper case</b>	Bottom of document	Set tab stops	New page
<b>Lower case</b>	Get document	Status line ON/OFF	Paginate document
	Create work area	US Units/Metric	Layout page
	Switch work areas		
	Headers		

If you decide not to use any additional options, press EXIT to return to your text display.

#### **PROSE PLUS Attributes Menu**

You display this menu by pressing F11 while in PROSE PLUS. The options on the Attributes Menu enable you emphasize text with bolding and underlining, in addition to altering the case of large blocks of text. The options are:



When you use any of the attributes described above, keep in mind that the particular operation begins at what is called the *point of operation*. This is the point between the character or space beneath the cursor and the character or space to the left of the cursor. To illustrate this concept, let's assume we are bolding a segment of text. If you are bolding from left to right, the character beneath the cursor at the beginning of the operation is bolded. However, if you bold from right to left, the character beneath the cursor at the beginning of the operation is not bolded.

### **Bold**

An effective way to highlight important pieces of text in a document is to bold it. To bold a segment text, do the following:

1. Position the cursor at either end of the text segment you wish to bold.
2. Press F11 to display the Attributes Menu.
3. Choose "Bold" from the menu.
4. Position the cursor at the opposite end of the text segment you wish to bold.
5. Press DO to complete the bolding operation.

This text contains **bolded characters**.

### **Underline**

Text can also be highlighted by underlining. To underline a segment text, do the following:

1. Position the cursor at either end of the text segment you wish to underline.
2. Press F11 to display the Attributes Menu.
3. Choose "Underline" from the menu.
4. Position the cursor at the opposite end of the of the text segment you wish to underline.
5. Press DO to complete the underlining operation.

This text contains underlined characters.



**Clear attributes**

Text that has been bolded and/or underlined can be returned to normal text using the “Clear attributes” option. To clear bold and/or underline attributes from your text, do the following:

1. Position the cursor at either end of the text segment containing the attributes you wish to clear.
2. Press F11 to display the Attributes Menu.
3. Choose “Clear attributes” from the menu.
4. Position the cursor at the opposite end of the text segment containing the attributes you wish to clear.
5. Press DO. The attributes are now cleared.

**Uppercase**

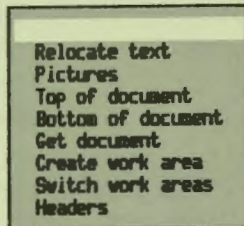
1. Position the cursor at either end of the text segment you wish to change to uppercase.
2. Press F11 to display the Attributes Menu.
3. Choose “Uppercase” from the menu.
4. Position the cursor at the opposite end of the text segment you wish to change. The text segment that will be changed becomes highlighted in reverse video as you move the cursor.
5. Press DO to change the case and complete the operation.

**Lowercase**

1. Position the cursor at either end of the text you wish to change to all lowercase.
2. Press F11 to display the Attributes Menu.
3. Choose “Lowercase” from the menu.
4. Position the cursor at the opposite end of the segment of text that you wish to change. The text segment that will be changed becomes highlighted in reverse video as you move the cursor.
5. Press DO to change the case and complete the operation.

**PROSE PLUS Edit Menu**

You display this menu by pressing F12 while in PROSE PLUS. The options on the Edit Menu enable you to move the cursor and text segments from place to place in your document. The options are:

**Relocate Text**

Sometimes you may want to move a text segment or picture reference frame to another part of your document, or you may want to copy the same text to several places in your document. "Relocate text" allows you to do these things. To relocate a text segment:

1. Move the cursor to one end of the text you want to relocate.  
For example,  
This is some text to move. This is some more text, but I do not want to move this.
2. Press REMOVE.
3. Move the cursor to the other end of the text you want to relocate. As you move the cursor, the text segment becomes highlighted in reverse video. For example,  
This is some text to move. This is some more text, but I do not want to move this.  
(If you change your mind, press CANCEL to cancel your selection.)
4. Press DO. The highlighted text disappears from the screen until the operation is completed.

5. Move the cursor to where you want to relocate the segment.  
For example,

This is some text, but I do not want to move  
this.

(If you want to put the text back in the original place in addition to relocating the text, leave the cursor positioned where you removed the text.)

6. Press F12.
7. Choose "Relocate text" and press DO. PROSE PLUS moves the text to the new location. For example,

This is some more text, but I do not want to move  
this. This is some text to move.

If the text is not filled properly, use the "Fill text" option from the Format Menu.

8. You can relocate the text as many times as you wish (until you press the REMOVE key again) by moving the cursor to the next place where you want the text relocated, choosing "Relocate text" from the Edit Menu, and pressing DO.

### **Pictures**

Choose "Pictures" to insert a picture reference into your document, or to change or view an existing one. If the cursor is on or between the top and bottom borders of a graphics reference frame at the time you choose "Pictures," that picture will be displayed. If the picture was created by PROSE PLUS, you are placed into the PROSE PLUS graphics editor. If the picture was created by an external application, there is no capability for editing it. The picture is merely displayed for your viewing.

If the cursor is located outside a picture reference frame PROSE PLUS assumes you are either creating a new picture or inserting an existing picture, and will display the File Selection Menu. If you choose to create a new picture, press INSERT HERE and the Name a File Form is displayed. Complete the form and press DO. The graphics editing screen will be displayed. The status line will indicate that a new picture file has been created.

If you choose to add a filed picture to your document, select the picture from the File Selection Menu. If the picture is not of a valid type, you will be notified. If the picture was created by PROSE PLUS, you are placed into the PROSE PLUS graphics editor. If the picture was created by an external application,

there is no capability for editing it. The picture is merely displayed for your viewing. At this point you are prompted to either continue with the insert operation or cancel it and return to the text editor.

Chapter 4 explains in greater detail how to draw a picture and how to edit an existing PROSE PLUS picture.

#### ***Top of document***

Use “Top of document” to move the cursor to the top line of the document you are editing. This is useful if you want to add text, beginning at the top of your document, or before you begin locating text segments with the FIND key.

#### ***Bottom of document***

Use “Bottom of document” to move the cursor to the last line of your document. This is useful if you want to add text, beginning at the bottom of the document.

#### ***Get document***

Using “Get document” you can retrieve another document stored on your Professional and place it into your current work area. The current work area can be either a temporary work area that you created during the editing session, or the main document work area containing the document you are editing. Portions of that document can be used in the text you are currently editing. Do not use “Get document” to retrieve a picture. To retrieve an existing document, do the following:

1. Move the cursor to the location in the current work area where you would like the retrieved document to appear. The line containing the cursor will follow the retrieved document.
2. Choose “Get document” from the Edit Menu. The File Selection Menu will be displayed.
3. Select a document from the menu and press DO. The document will be retrieved and inserted into the current work area.

#### ***Create work area***

When creating a document or modifying an existing document, it is sometimes useful to place some of the text into a temporary location, or work area for later use in the document. Text placed into a work area can be edited the same as any

other with the PROSE PLUS text editor. However, because the work area is a temporary location, what is in there when you leave the editor is lost. To create a work area, do the following:

1. Choose "Create work area" from the Edit Menu.
2. The status line will prompt you to enter a name for the work area and press DO. This name can be up to 15 characters long, including spaces.
3. The screen will clear and the new work area will be displayed and its name will appear in the left portion of the status line.

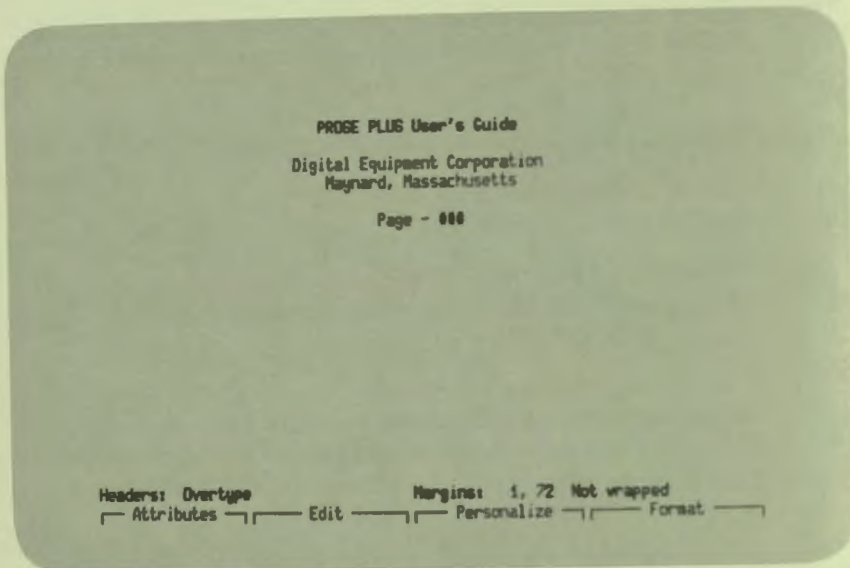
### **Switch work areas**

Once a work area has been created, it is possible to switch from one work area to another. PROSE PLUS allows a maximum of 5 temporary work areas to exist at any time. When a work area is "switched," the contents of the recalled work area replaces the text that was on the screen.

### **Headers**

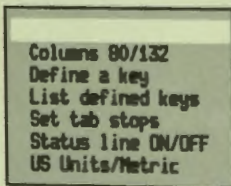
Choosing "Headers" allows you to enter up to ten header lines for your document. The header lines will appear at the top of each page starting at page two. The number of header lines that appear on each page is determined by entering a value between 0 and 10 in the "Number of headers" field on the "Layout page" form (described later in this chapter).

When you choose "Headers," the screen clears and the status line changes to indicate the current operation. PROSE PLUS will recognize up to the first ten lines of text as header lines. The actual number is defined by the "Number of headers" value on the "Layout page" form. You can use the various PROSE PLUS options to format or assign attributes to the header text. If you wish page numbers added to the header during automatic pagination, insert ### at any location in the header where you would like the page numbers to appear. You should include one "#" symbol for each digit in the highest expected page number. If you include more "#" symbols than there are digits in the page number, the page number will left justify automatically.



### PROSE PLUS Personalize Menu

You display this menu by pressing **F13** while in **PROSE PLUS**. The options on the Personalize Menu enable you to set defaults which are saved until you change them. The options are:



#### Columns 80/132

When you select "Columns 80/132" and press **DO**, the number of columns on the screen display changes. If the display was set to 80 columns, it changes to 132 columns. If the display was set to 132 columns, it changes to 80 columns.

**Define a Key**


With “Define a key,” you can perform up to 250 keystrokes with the pressing of a single key. Eight keys (F17, F18, F19, F20, PF1, PF2, PF3, and PF4) are user-definable. The keystrokes specified can include characters, RETURN, function keys, user-defined keys, and any other PROSE PLUS text editor operation.

To define a key,

1. Press **F13** to display the Personalize Menu.
2. Choose the “Define a key” option. Note that the status line changes to indicate that you are now defining a key with subsequent keystrokes.

**NOTE:** Keep in mind that once you start defining a key **every** keystroke becomes part of the definition. This fact is particularly important when entering **ARROW** keystrokes, since they are not reproduced from the same cursor location that they were defined at. Rather, they are reproduced from the cursor location at the time the defined key is pressed.

3. Type the key sequence that you want assigned to one of the definable keys. For example, if you want to define a key that will center your company’s name and address at the top of a letter, you would type:
  1. Digital Equipment Corporation
  2. F14
  3. DOWN ARROW
  4. DO
  5. RETURN
  6. 146 Main Street
  7. F14
  8. DOWN ARROW
  9. DO
  10. RETURN
  11. Maynard, MA 01754
  12. F14
  13. DOWN ARROW
  14. DO

Use the  key to delete any errors you may make in typing this sequence. Your Professional keyboard will beep if you try to enter more than 250 keystrokes into a key definition.

4. Press **DO** to complete the key sequence. Or, press **CANCEL** if you do not want to assign the sequence to a user-definable key.
5. At the prompt, type in a name for the sequence. This name can be up to 60 characters long, and can include letters, numbers, spaces, and punctuation.
6. Press the key to which you want the sequence assigned; **F17**, **F18**, **F19**, **F20**, **PF1**, **PF2**, **PF3**, and **PF4** are available for this purpose. A list on the screen shows which keys are already defined. You can re-define a key if you want.

**WARNING:** If you are using a previously defined key in the key definition sequence, remember that the previously defined key may not represent just one keystroke, but rather the number of keystrokes used to define the key. If you should change the definition of the previously defined key, any user-defined keys containing that previously defined key in its definition will **not** be updated automatically.

PROSE PLUS remembers the definitions that you assign to function keys from one session to the next. Now, any time that you are in PROSE PLUS and you press the function key you've just defined, you will see the name and address as described in this example:

Digital Equipment Corporation  
146 Main Street  
Maynard, MA 01754

Note that when you press a defined key, the execution of the key definition begins at the current cursor location.

The best method for learning the many ways that user-definable keys can work for you is to spend some time experimenting with them. Here are some possibilities:

- Using the **REMOVE** and **DO** keys, define a key to delete the current line.
- Using the **INSERT HERE** and **RETURN** keys, define a key to open up a blank line.



- Replace one word that appears many times in a document with another word (for example, replace “January” with “February”). Once you’ve defined a key to do this, an otherwise tedious chore becomes a simple, fast, and precise operation. Use the **FIND, REMOVE, INSERT HERE, and DO** keys.
- Define a key to choose an option from the **Attributes, Edit, Personalize, or Format** Menus. Or define a key to choose several additional options at one time.

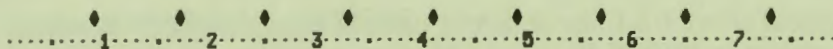
### **List defined keys**

“List defined keys” displays a list of user-defined keys that have been created with the “Define a key” option. Only keys which have been defined are listed. If there are no user-defined keys to be listed, the status line will indicate this to you.

### **Set tab stops**

Use “Set tab stops” to establish the columns where the cursor will stop each time the **TAB** key is pressed. The tab stops you set in **PROSE PLUS** will not affect other P/OS services. With the initial default setting, there is a tab stop at every 8 columns, starting at column 9. For example, if the cursor is in a column from 1 to 8, when you press the **TAB** key, the cursor jumps to column 9.

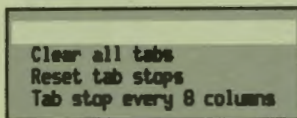
When you select “Set tab stops,” the status line changes and a ruler appears on your screen. Small diamonds above the ruler show the current tab stops. Using the **←** and **→** keys, move the cursor across the ruler to where you want a tab; then press **SELECT**.



To remove a tab stop, move the cursor to the diamond which represents the tab stop you want to remove and press **SELECT**.

When you have selected all the desired tab stops, press **DO**.

Pressing F11 displays the following menu:



“Clear all tabs” removes all tab stops.

“Reset tab stops” resets the tab stops to the settings that were in effect when you chose “Set tab stops.”

“Tab stop every 8 columns” sets the tab stops to the default settings of every 8 columns, beginning with column 9.

#### **Status Line ON/OFF**

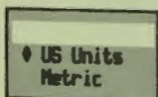
By initial default, 22 lines of your text are displayed on the screen. Lines 23 and 24 are reserved for the status line. If you want to display 24 lines of your text and no status line:

1. Press F13.
2. Choose “Status line ON/OFF.”

When you want to display the status line again, repeat this procedure. The status line cannot be turned off in the graphics editor.

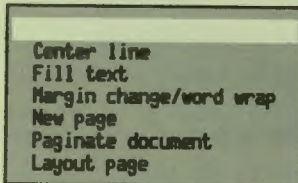
#### **US Units/Metric**

You can choose either US Units or Metric Units of measurement to determine page size values in PROSE PLUS (see “Layout page” later in this chapter). When you choose “US Units/Metric,” a menu is displayed from which you can select either system of measurement. This choice is saved from one session to another until you change it again. US Units is the initial default setting.



## PROSE PLUS Format Menu

You display this menu by pressing **ADDTNL OPTIONS** while in PROSE PLUS. The options on the Format Menu enable you to specify the organization and layout of text in your document. The options are:



### Center Line

To use “Center line,” first move the cursor to the line you want to center. Then choose “Center line” from the Format Menu. When you press **DO**, PROSE PLUS centers the line within the margins set for that paragraph.

### Fill Text

Use “Fill text” to clean-up paragraphs that allow word wrap. To fill text:

1. Move the cursor to the beginning of the paragraph you want to fill. For example,

```

This is a sample of text to be filled.
When I typed this      text in,
I  put in a lot of    spaces and carriage returns
that
I do not want now.
  
```

2. Press **ADDTNL OPTIONS**.
3. Choose “Fill text.”
4. Press **DO**. The paragraph fills. For example,

```

This is a sample of text to be filled. When I typed this text in, I put in
a lot of carriage returns that I do not want now.
  
```

Note that the cursor is at the end of the filled text.

“Fill text” will not operate on text set to “no wrap,” which is the default setting. Use “Margin change/word wrap” to turn on word wrap or justification. To get rid of spaces in text that is not wrapped, use **REMOVE**.

**Margin Change/Word Wrap**

When you begin creating text with PROSE PLUS, the left margin is set to column 1 and the right margin to column 72. Word wrap is off for each paragraph you enter. These are the default settings. (A blank line—two carriage returns—marks a separation between paragraphs.)

Word wrap is a very useful PROSE PLUS feature that **automatically** moves any word extending beyond the right margin to the beginning of the next line. You do not have to press the RETURN key when you get to the right margin. Word wrap should be turned **off** when entering tables, columns of text, programs, and the like.

To change margins or the word wrap setting for text, use “Margin change/word wrap.” PROSE PLUS uses the new setting(s) for the paragraph you selected and for subsequent paragraphs (moving forward through the text) until a previously embedded ruler is encountered.

To set margins or word wrap:

1. Move the cursor to the first line of the text for which you want to change margins and/or word wrap.

If you have not yet typed in the paragraph, before changing the setting, press RETURN twice to indicate a new paragraph.

If you move the cursor to a line within a paragraph, PROSE PLUS creates the new margin setting starting where the cursor is, but does not insert a blank line.

2. Press ADDTNL OPTIONS.
3. Choose “Margin change/word wrap.”
4. Press DO. The status line changes and a ruler appears in your text display. For example,

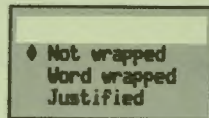
.....v.....1.....2.....3.....4.....5.....6.....7.....v.....

This is the last line of a paragraph.

This is a new paragraph. I want to turn justification on for this paragraph. I also want to change the margins.

Each dot or number on the ruler represents a column. The “V” symbols mark the location of the left and right margins.

5. To set a new left margin, move the cursor to the column you want and press **SELECT**. (Press **EXIT** to cancel your choice and return to the text display.)
6. The cursor is moved to the “V” marking the location of the right margin. Move the cursor to the column you want and press **SELECT**. You can use **SELECT** to toggle back and forth between margins until you’re satisfied with the settings you’ve made.
7. If you want to change the word wrap setting, press **F11**. A menu appears showing the possible word wrap settings; the current word wrap setting is indicated with a diamond. Move the pointer to the choice you want and press **DO**. You are returned to the margin ruler.



8. Press **DO** when you are ready to leave “Margin change/word wrap.” If you change the margins on wrapped text, **PROSE PLUS** rewraps the text so that it fits within the new margins. **PROSE PLUS** will continue to rewrap until the next embedded ruler is encountered.

To find out the word wrap and margin settings for any paragraph, move the cursor to it and look at the status line.

New text can be inserted into existing text containing different margin settings. For instance, you might want to insert a note with narrower margins than the body of text it is being inserted into.

This is accomplished using the following procedure:

1. Locate the cursor at the point where you want to insert the note.
2. Press **INSERT HERE** and hit **RETURN** to create a blank line for the start of a new paragraph.
3. Press **ADDTNL OPTIONS** and choose “Margin Change/Word Wrap.”
4. The margin ruler will appear on the screen.
5. Set the margins to the appropriate columns (usually narrower than the main body of text).

6. Press **DO** once and insert your text.
7. When you have finished typing your text, press **RETURN** to leave a blank line and end the paragraph.
8. Press **DO** to complete the operation.
9. If necessary go back and adjust the indenting of the text where the cursor currently is.

For formatting numbered or bulleted lists, use the following procedure:

1. Set the left margin at the column you want the text to line up against.
2. Enter your text according to the conventions described in this chapter.
3. Using the **ARROW** keys, place numbers or bullets for each of the list items **outside** the left margin.

Now, if you should change the margins, the list will remain intact along with the numbers or bullets. Numbered and bulleted lists can be destroyed by changing the format from not wrapped to wrapped.

### ***New page***

“New page” inserts a page break where you want your text to begin printing on the next page. This is particularly useful when you need to print a table or list on one page without breaking it up onto two pages. Or, when you need to start a section of a report on a new page.

To insert a page break,

1. Move the cursor to any spot on the first line to be printed on the next page. For example,
 

This is some text. I have written several pages of text--this is only part of what I have written--and I want to insert a page break so that the printer will print my text as I request. For example, I need to use page breaks to make sure that a table gets printed all on one page.
2. Press **ADDTNL OPTIONS**.
3. Choose “New page.”

4. Press **DO**. A thin double line appears, showing where the page will break, and the new page begins. For example,

This is some text. I have written several pages of text--this is only part of what I have written and I want to insert a page break so that

---



---

the printer will print my text as I request. For example, I need to use page breaks to make sure that a table gets printed all on one page.

Set page breaks in your text **after** all edits have been made, because a change in the text can affect your page breaks.

When printing documents, P/OS causes the printer to jump (form feed) to the next page on encountering a page break. The double line that appears in your screen display is not printed on the paper copy.

To delete an existing page break, move the cursor to the page break and press the **⌘X** key.

### ***Paginate document***

PROSE PLUS has the capability to automatically paginate your entire document. The "Paginate document" feature of PROSE PLUS always paginates from the top of the current page (as defined by any existing page breaks) or the top of the document, whichever is closer to the cursor, down. Choose "Paginate document" from the Format Menu. If the document is several pages in length, PROSE PLUS will require several moments to complete the pagination.

When the automatic pagination is completed, the bottom of the last page appears on your screen. Any header lines you requested (see next section) are inserted, and wide double lines mark the pages on your screen. If a paragraph is encountered in the document that begins on the last two lines of a page, PROSE PLUS will automatically begin the paragraph on the next page. PROSE PLUS also assures that a paragraph ending at the top of a page contains no less than three text lines. PROSE PLUS will move lines from the bottom of one page to the top of the next to assure this.

**NOTE:** If you insert a "New page" into a document after using "Paginate document," you must do another automatic pagination to get the headers on the new page.

**Layout page**

The size of your document when printed depends upon how you layout the pages of the document through PROSE PLUS, and how you have set your P/OS printer characteristics. "Layout page" allows to set the following values:

- Page length
- Page width
- Top margin
- Bottom margin
- Number of headers

When you choose "Layout page," a small form menu appears on the screen containing the items listed above, along with their current values. You can change the values by using the **↑** and **↓** ARROW keys to highlight the item to be changed. Type in a new value, and press **DO** when you are satisfied with the page layout.

Page length:	11 in.
Page width:	8 in.
Top margin:	1 in.
Bottom margin:	1 in.
Number of headers:	0

US letter (8 x 11 in.)
US legal (8 x 14 in.)
A3 (420 x 297 mm.)
A4 (297 x 210 mm.)
A5 (210 x 148 mm.)
A6 (148 x 105 mm.)

An alternative to entering values as described above, is to choose a pre-defined page layout from a list of standard sizes. When you first selected "Layout page," the status line changed and function key **F11** became active for selecting "Standard sizes." If you press **F11** with "Standard sizes" displayed in the status line, a small menu will appear on the screen. Using the **↑** and **↓** ARROW keys, highlight a page size of your choice and press **DO**. The "Layout page" menu will be updated with the new page length and page width values. You can still change any of the values in the "Layout page" form prior to pressing **DO**.



# 4

---

The PROSE PLUS  
Graphics Editor



## Chapter 4

---

### The PROSE PLUS Graphics Editor

Using the PROSE PLUS graphics editor, you can prepare charts, graphs, and other pictures for inclusion in memos, reports, and other kinds of text. You can create, edit, move, replace, and format the pictures using the editing keypad and certain function keys on your keyboard that you learned about in Chapter 2. The pictures you create can be as simple or as complex as you like.

With PROSE PLUS, creating a picture using predefined object types such as lines, arcs, and rectangles is easy to do. With the help of the editing keypad and several function keys, you can easily change the appearance of these basic objects on the screen to create a picture. The picture you create is saved and can be used again in other documents. When the picture is printed using a DIGITAL LA-50 or LA-100 printer, what appears on your screen is what the printer puts on paper. In addition, when you return to the text editor, the display shows where the picture appears in relation to the text, and how much space it uses on the page.

This chapter is divided into three sections:

- Getting Started with the Graphics Editor—getting around the screen
- Creating new pictures—all you need to know to create pictures
- Editing old pictures—making old pictures better

## GETTING STARTED WITH THE GRAPHICS EDITOR

To get started with the graphics portion of PROSE PLUS do the following:

1. If you are drawing a new picture or inserting an existing picture, place the text editor cursor at the location in the document where you would like the picture to appear. Any text that happens to be on the same line as the cursor will be moved to the line immediately beneath the bottom of the picture when it is completed. Note that any blank space left at the top of the picture in the graphics editor will appear in the document, and blank space left at the bottom of the picture will be truncated upon completion of the picture.<sup>1</sup>
2. Press F12 to display the text editor's Edit Menu and choose "Pictures." The File Selection Menu is displayed.
3. If you plan to insert an existing picture into the text, choose one of the pictures. Otherwise, press **INSERT HERE** to draw a new picture.

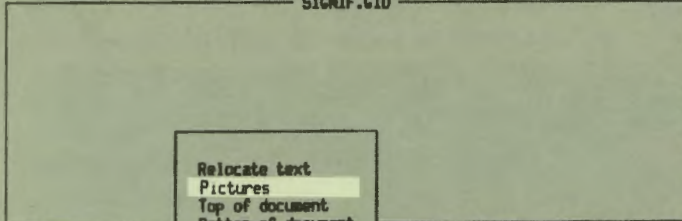
If you happen to choose a picture file that was not created using PROSE PLUS on a Professional, the picture is displayed for your viewing only. If this picture is greater than 24 lines in height, it is displayed at a reduced scale on the screen. PROSE PLUS distinguishes between picture files drawn with PROSE PLUS and those drawn by other applications. Only those pictures created with PROSE PLUS can be edited using the PROSE PLUS graphics editor. Since the picture cannot be edited, you are prompted in the status line to confirm that the displayed picture is what you want inserted into the current document. If you answer in the affirmative, you are returned to the text editor with the selected picture file referenced in the document.

---

<sup>1</sup>If your cursor happens to be on an existing picture's reference frame in the text editor when you choose "Pictures," then that picture is displayed for editing. More information on editing existing pictures is contained later in this chapter.

When the FIND key on the editing keypad is pressed, the cursor is moved to the nearest significant point on the screen. Significant points exist on every object drawn on the screen. The illustration below shows where they occur on

SIGNIF.GID



each objec  
not always  
example i  
pictures r  
to each o

ant point to the cursor is  
t object as the following  
ey is useful when drawing  
ent of objects in relation  
you wish to connect a line

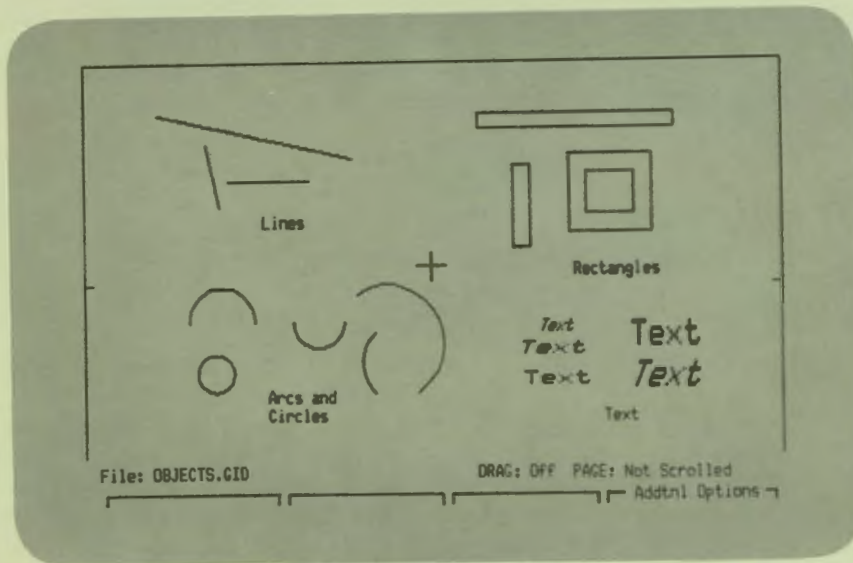
Now doing: Overtyping      Margins: 10, 70 Justified  
Attributes   Edit   Personalize   Format

### Help

Useful information about the graphics editor can be obtained at anytime while you're in the graphics editor by pressing the HELP key.

### The Graphics Editing Screen

The graphics editor allows you to draw pictures that fit on standard sized pages. The picture you create consists of one or more items that you draw called *objects*. An object is represented by any one of the following items:



Each picture can be up to 60 lines in height, and contain up to 300 separate objects. Of those 300 objects, 60 can be text objects, each containing up to 80 individual text characters. You draw an object by *inserting* it onto the screen. This is done using five basic steps:

1. Using the ARROW keys, move the cursor to the location on the screen where you wish to begin drawing the object.
2. Press INSERT HERE on the editing keypad to make the cursor ready to draw.
3. Select the object you wish to draw by pressing F12 until the selected object type appears on the status line.
4. Draw the object using the keyboard for text, or the ARROW keys on the editing keypad for all other object types.
5. Press DO when you have completed drawing the object.

Although the graphics editor allows you to draw pictures up to 60 lines in height, only 24 lines at a time can be displayed on your Professional's screen. Therefore, the graphics editor display is divided into 5 equal sections containing approximately 12 lines each, or about half the height of the screen. Each section is identified by small *tics* located to either side of the screen. The first two sections are marked by 1 tic on either side of the screen. If you press **NEXT SCREEN** you'll see that sections two and three are marked by 2 tics on either side of the screen. There are 3 tics between sections three and four, and 4 tics between sections four and five.

### **Moving the Cursor**

The graphics editor cursor appears as a blinking cross-hair when the editor is **not** in the insert mode. While in this state, it can be moved anywhere on the screen by pressing one of the four **ARROW** keys or the **FIND** key on the editing keypad. Single strokes of any of the **ARROW** keys cause the cursor to move in small increments in the direction of the arrow on the pressed key. Holding any of those keys down for a longer period of time causes the cursor to move rapidly in the direction of the arrow on the pressed key. If the bottom or top of the screen is encountered, the cursor will either wrap to the opposite side of the screen, or the page will scroll on the screen depending on how you have set the "Scroll" feature on the Additional Options Menu. Scrolling is discussed in greater detail later in this chapter. The cursor moves relative to the page and may appear to "jump" on the screen.

To move the cursor quickly through sections of the page press **NEXT SCREEN** to move down the page, and press **PREV SCREEN** to move up the page. The cursor maintains its exact position on the screen.

In all cases, if the bottom or top of the page is encountered, the cursor will stop.

### **Find**

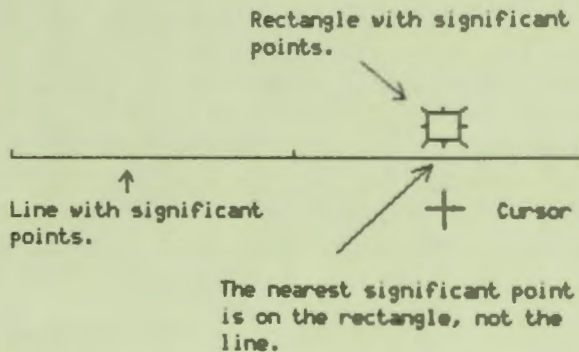
When the **FIND** key on the editing keypad is pressed, the cursor is moved to the nearest *significant point* on the screen. Significant points exist on every object drawn on the screen. The illustration below shows where they occur on each object:



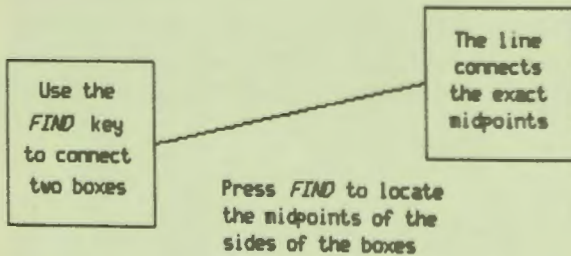
Significant Points °



The nearest significant point to the cursor is not always located on the nearest object as the following example illustrates:



The *FIND* key is useful when drawing pictures requiring accurate placement of objects in relation to each other. For example, if you wish to connect a line between the midpoints of the sides of two rectangles, do the following:





1. Position the cursor near the midpoint of the side of one of the rectangles and press **FIND**. The cursor will move to the midpoint of that side.
2. Press **INSERT HERE** and toggle the object type using **F12** until "Drawing lines" appears in the status line.
3. Draw a line to where you approximate the midpoint of the side of the second rectangle to be.
4. Press **FIND** and the line will be positioned at the midpoint of the second rectangle.
5. Press **DO** to complete the operation.

The **FIND** key is also useful to position text on the screen in relation to previously entered text. For example, if you have a line of text on the screen and you wish to position another line directly below it and flush left, do the following:

1. Position the cursor near the lower left corner of the first line of text and press **FIND**.
2. Press **INSERT HERE** and toggle the object type using **F12** until "Entering text" appears in the status line.
3. The text you type in will be aligned with the previous line.
4. Press **DO** to complete the operation.

The **FIND** key is **always** active. It can be used at anytime in the graphics editor to relocate the cursor to an object.

### **Drag**

The "Drag" feature allows you to move an object or the selection box around the screen without changing their size or shape. Movement of the object is controlled by the **ARROW** keys. Drag is automatically **off** when you choose to draw lines, rectangles, arcs and circles, or when you enter text. At any point during the drawing of one of these objects, you can press **F11** to turn drag **on**. While drag is on, the **ARROW** keys serve to move the object around the screen rather than changing its shape or dimensions. The status line tells you when drag is on or off, and informs you when the **F11** key can be used to toggle the feature.

### CREATING NEW PICTURES

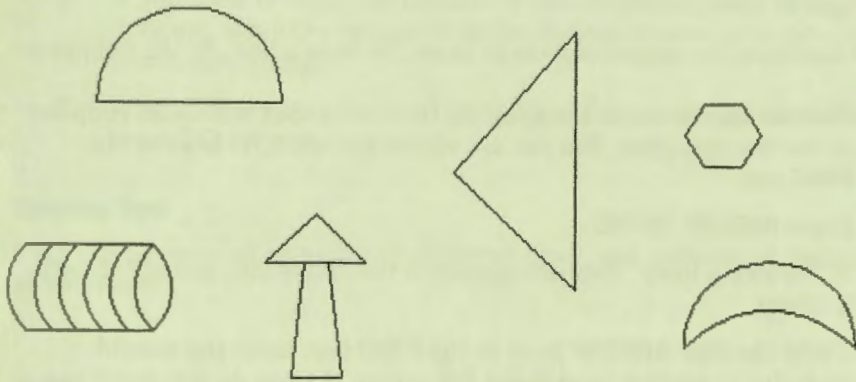
Drawing a picture in PROSE PLUS is as easy as the five basic steps described at the beginning of this chapter. First, you should have an image in your mind of what you plan to draw. Having a general outline of what you want to include in the picture is always a good idea. For instance,

- If the picture you draw is for an overhead transparency, what size text will be most legible?
- If you are drawing a flowchart, what shapes will you use?
- Would it save you time to create a library of different shapes that could be used many times in the picture?

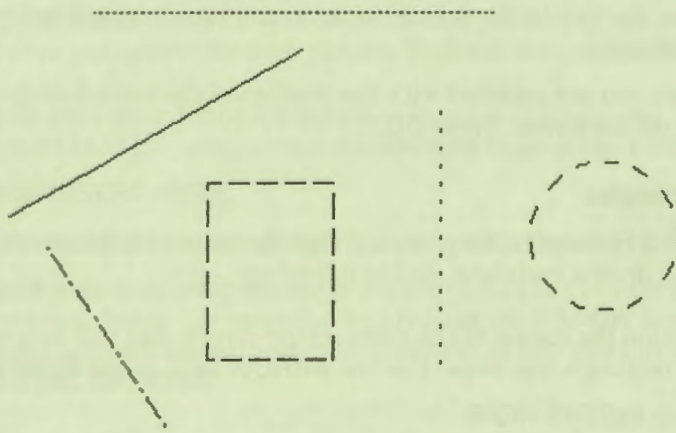
The picture you create can consist of one or more basic object types. These object types include:

- Lines
- Rectangles
- Text
- Arcs or circles

Certain characteristics of these objects can be changed including line styles, arc angles, character size, and other attributes. Combinations of these objects can result in additional shapes such as triangles, hexagons, semi-circles, and many others.



You can create other shapes by  
combining objects on the screen.



You can change the line style of  
objects you *draw* on the screen.

### Drawing Lines

Straight lines are the easiest objects to draw. To draw a line, do the following:

1. Position the cursor to the point on the screen that will be an endpoint of the line you draw. You can use either the **ARROW** keys or the **FIND** key.
2. Press **INSERT HERE**.
3. If "Drawing lines" does not appear in the status line, press **F12** until it does.
4. Using the four **ARROW** keys or the **FIND** key, move the second endpoint to another location on the screen. As you do this you'll see the line flashing on the screen.
5. If you want to change the style of the line, keep pressing **F13** to choose one of the six available line styles.
6. If you want to move the location of the line, press **F11** to turn drag **on**. Move the line to the desired location and turn drag **off** before continuing.
7. When you are satisfied with the position of the second endpoint of the line on the screen, press **DO**.

### Drawing Rectangles

When drawing rectangles, keep in mind that the cursor marks one corner of the rectangle. To draw a rectangle, do the following:

1. Position the cursor to the point on the screen that will be a corner of the rectangle you draw. Use the **ARROW** keys or the **FIND** key.
2. Press **INSERT HERE**.
3. If "Drawing rectangles" does not appear in the status line, press **F12** until it does.
4. Using the **ARROW** keys or the **FIND** key, form the rectangle on the screen. As the rectangle forms, its sides will flash on the screen.
5. If you want to change the line style of the rectangle, keep pressing **F13** to choose one of the six available line styles.

6. If you want to move the location of the rectangle, press F11 to turn drag **on**. Move the rectangle to the desired location and turn drag **off** before continuing.
7. When you are satisfied with the shape of the rectangle on the screen, press DO.

### Entering Text

Text can be entered using many different sizes, and a choice of two styles as shown here:

Normal                  Normal                  Normal

*Italic*                  *Italic*                  *Italic*

PROSE PLUS allows up to 60 text objects, each containing not more than 80 text characters and spaces for each picture. To enter text, do the following:

1. Move the cursor to the approximate location of where you would like to start the text, using either the ARROW keys or the FIND key.
2. Press INSERT HERE.
3. If the status line does not say "Entering text," press F12 until it does.
4. Enter your text using the *main keyboard*. You can change the size at any point during the operation by pressing the ARROW keys. You can toggle between normal and italic typefaces at any point by pressing ADDTNL OPTIONS.

**NOTE:** Normal and italic text **cannot** be combined in the same text object. A line of text with an italic word in the middle is comprised of at least three text objects.

5. If you want to move the location of the text, press F11 to turn drag **on**. Move the text to the desired location and turn drag **off** before continuing.
6. When you have finished entering the text (remember the 80 character per object limit), press DO.

### Drawing Arcs or Circles

When drawing arcs and circles, keep in mind that the cursor marks one endpoint of the arc. To draw an arc, do the following:

1. Position the cursor where you would like one endpoint of the arc to be. Use the **ARROW** keys or the **FIND** key.
2. Press **INSERT HERE**.
3. If the status line does not say "Drawing Arcs or circles," keep pressing **F12** until it does.
4. Using the **ARROW** keys increase the size of the arc in the direction you choose. Note that the **ARROW** and **FIND** keys do not change the arc angle—only its size.
5. If you wish to change the angle of the arc, keep pressing the **ADDTNL OPTIONS** key to choose one of the available arc angles. A complete circle can be formed using the arc angle feature.
6. If you would like to change the line style, keep pressing **F13** to choose one of the six available styles.
7. You can also change the location of the arc by enabling the drag feature. Do this by pressing **F11**. Then move the arc to whatever location you desire and press **F11** to turn off the drag feature before continuing.
8. When you are satisfied with the object on the screen, press **DO**.

### Changing an Existing Object

Even though you have completed drawing an object, you can still move it to a different location on the screen, or change its characteristics if you want to.

#### *Moving an Existing Object*

To move an existing object, do the following:

1. Place the cursor on the object using the **ARROW** keys or the **FIND** key, and press **DO**. The object is now made active and flashes to indicate that it is active. The status line changes to show the current operation. **Drag** is automatically turned **on** whenever you make an object active.

2. Move the object to its new location using the **ARROW** keys.
3. When you are satisfied with the new location, press **DO** to complete the operation.

A group of objects can be moved using the **SELECT** key described later in this chapter.

### ***Changing the Characteristics of an Existing Object***

To change the characteristics of an existing object, do the following:

1. Place the cursor on the object using the **ARROW** keys or the **FIND** key, and press **DO**. The object is now made active and flashes to indicate that it is active. The status line changes to show the current operation. **Drag** is automatically turned **on** whenever you make an object active.
2. Press **F11** to turn **drag off**.
3. Alter the size, shape, and style of the object as though you were drawing a new object. You can even change the object type.
4. When you are satisfied with how the object looks, press **DO** to complete the operation.

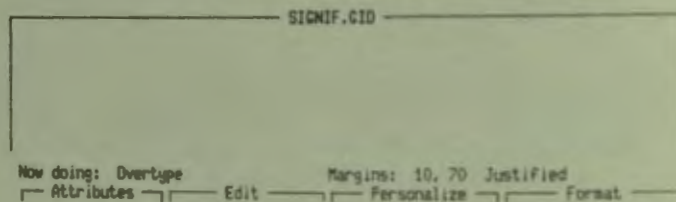
### **Exiting Back to the Text Editor**

To automatically save the picture file you have just created or edited, press **EXIT**. The picture will be saved and you are returned back to the text editor at the location in the document you were at before. A picture reference frame is placed into the document signifying the total height of the picture you created or edited.

any of those keys down for a longer period of time causes the cursor to move rapidly in the direction of the arrow on the pressed key.

To move the cursor quickly through sections of the page press **NEXT SCREEN** to move down the page, and press **PREV SCREEN** to move up the page.

When the **FIND** key on the editing keypad is pressed, the cursor is moved to the nearest significant point on the screen. Significant points exist on every object drawn on the screen. The illustration below shows where they occur on



If you edited a picture that was already referenced in the document, the changes you made to the picture could have an effect on the height of the picture reference frame in the document. For example, if you added to the height of the existing picture by placing more objects below the existing picture, the height of the picture reference frame would be increased.

If you do not wish to save the changes you made, use the "Quit" option described later in this chapter.

#### Additional Functions

Additional editing features are provided in the **PROSE PLUS** graphics editor to:

- Make an object active, or select a group of objects
- Permanently remove an object or group of objects
- Cancel an operation that has not been completed



**DO**

As described in an earlier section, the **DO** key is used to complete a drawing function. It is also used to make active an existing object. Before an existing object in a picture can be moved around the screen or changed, it must be made active. An object that is active flashes on the screen and can be changed using the features previously described in this chapter. To make an object active, move the cursor onto some portion of the object using the **ARROW** keys or the **FIND** key. Then press **DO**. Now the object is active and you can move it around the screen with the **ARROW** keys or modify it. You can even change it to another object by pressing **F12**.

**NOTE:** If the object that is made active is text, and you change the object type, the text is permanently removed and can only be recovered by cancelling the operation. The **CANCEL** key is discussed below.

**CANCEL**

Pressing **CANCEL** automatically terminates whatever operation is currently taking place without affecting completed operations.

**SELECT**

The **SELECT** key on the editing keypad is used to identify an object or group of objects on the screen. When **SELECT** is pressed, the status line indicates that you are "Selecting a group." You can draw a box around an object or group of objects using the same techniques you learned for drawing rectangles. Selected objects can be removed, copied, and saved.

If you wish to remove the objects you can do so by pressing **REMOVE**.

If you want to copy the objects to another location on the screen, press **DO** now. The objects will temporarily disappear from the box. Now move the select box to the desired location on the screen and press **INSERT HERE**. You can reposition the box and copy the objects as many times as you like. Pressing **CANCEL** before pressing **INSERT HERE** terminates the operation without any changes to the original picture. Pressing **CANCEL** after **INSERT HERE** has been pressed terminates the operation, but leaves the changes intact.

Selected objects can be saved using the "Save" feature on the Additional Options menu. The "Save" feature is discussed in greater detail later in this chapter.

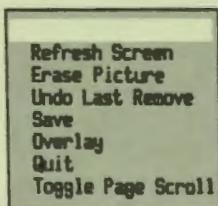
**REMOVE**

The REMOVE key on the editing keypad is used to delete an object or a group of objects in a picture. Individual objects are removed by moving the cursor to any point on the object and pressing REMOVE. The object is deleted from the picture. The FIND key, described earlier, can be used to place the cursor on the object to be removed.

Groups of objects are removed by drawing a select box around the group of objects and pressing REMOVE. The objects must be completely enclosed by the rectangle, although a portion of the object can be on the edge of the select rectangle.

**Additional Options Menu**

When pressed, the ADDTNL OPTIONS function key displays the Additional Options Menu, provided that "Addtnl Options" appears in the status line as shown below: The Additional Options Menu looks like this:



Each of these options are described in the sections that follow.

**Refresh screen**

When you choose "Refresh screen," PROSE PLUS blanks the screen and redraws the entire picture as it currently exists. This is a useful feature, particularly after you have removed an object that partly covered another. When an object is removed, portions of other objects it touches may seem to be erased. By refreshing the screen any partially erased objects are redrawn.

**Erase picture**

Choosing "Erase picture" **permanently** removes the currently displayed picture and deletes it from your system. Because of the permanent nature of this operation, you are prompted for confirmation of the action before the screen is erased.

**Undo last remove**

At anytime during your picture editing session, the **last** object or group of objects that were removed can be restored to the screen by choosing "Undo last remove." This only works with the last remove. Previous removes cannot be recovered.

**Save**

The "Save" option allows you to save an entire picture, or a portion of it, without having to exit back to the text editor. To save an entire picture, do the following:

1. Choose "Save" from the Additional Options Menu. The Name a File Form is displayed.
2. Complete the form and press **DO**. The picture will be saved under the file specification named on the form.

**NOTE:** The file you specify must be in the same directory as the document. PROSE PLUS cannot reference a picture file in a directory other than the current directory. So, although the file can be saved to another directory using the "Save" feature, it cannot be used unless the document is contained in the same directory, and the current directory is changed through P/OS File Services.

A group of objects within a picture can be saved using the following procedure:

1. Press **SELECT** and draw a rectangle around one or more objects that you wish to save.
2. Press **DO**. The selected objects will disappear from within the rectangle.
3. Press **ADDTNL OPTIONS** and choose "Save." The Name a File Form is displayed.
4. Complete the form and press **DO**. The group of objects will be saved under the file specification named on the form.
5. Press **CANCEL** to restore the objects and remove the select rectangle.

### Overlay

The "Overlay" option allows you to take a previously saved picture or group of objects saved from a picture, recall it, and place it into the picture you are currently editing. When you choose "Overlay" the File Selection menu is displayed. Choose a file containing the picture or group of objects you want to overlay and press DO. The status line informs you as to whether or not the file you have selected is a valid picture or group.

If the file was saved as an entire picture, then the entire picture is overlaid on your screen. As the name implies, an overlaid picture does not erase what is currently on the screen. If the file you are overlaying was saved as a selected group, in other words selected using the **SELECT** key before saving, then only what was contained in the select box during the save operation is overlaid on the current picture. These objects can be moved around the screen using the editing features described in this chapter.

1. The select box and its contents are displayed on the screen.
2. The status line prompts you for confirmation that it is the correct picture to overlay. Press **DO** if it is, or **CANCEL** if it is not.
3. If you press **DO** the contents of the box will disappear, but the select box will continue to flash.
4. Using the **ARROW** keys or **FIND** key, move the select box to the location on the screen where you want it.
5. Press **INSERT HERE** to overlay the picture.

### Quit

If you do not wish to save the changes you have made in the current picture editing session, choose "Quit." When you choose this option, you are prompted to confirm the action. If you enter **YES**, you are returned to the text editor without changing the picture reference in the document. If you enter **NO**, the action is cancelled and the picture is left exactly as it was before you selected the "Quit" option. You can then continue with your picture editing.

If you are creating a new picture, and you enter **YES**, no picture file is saved and no picture reference frame is inserted into the document.

**Toggle page scroll**

The “Toggle page scroll” feature controls what happens to the cursor when it encounters either the top or bottom edge of the screen. The cursor always wraps to either the left or right sides of the screen. In other words, if the cursor moves off the screen on the right side, it reappears on the left side. And, if the cursor moves off the screen on the left side, it reappears on the right. Toggling page scroll off and on only affects the vertical positioning of the cursor. If page scroll is set to **Not scrolled** and the cursor is moved off the top or bottom edge of the screen, it will reappear at the opposite edge.

When page scroll is **Scrolled**, the page scrolls down if the cursor encounters the top edge of the screen. If it encounters the bottom edge of the screen, the page scrolls up. In either case, the cursor never leaves the page. If the cursor encounters either the top or bottom of the page, it stops.

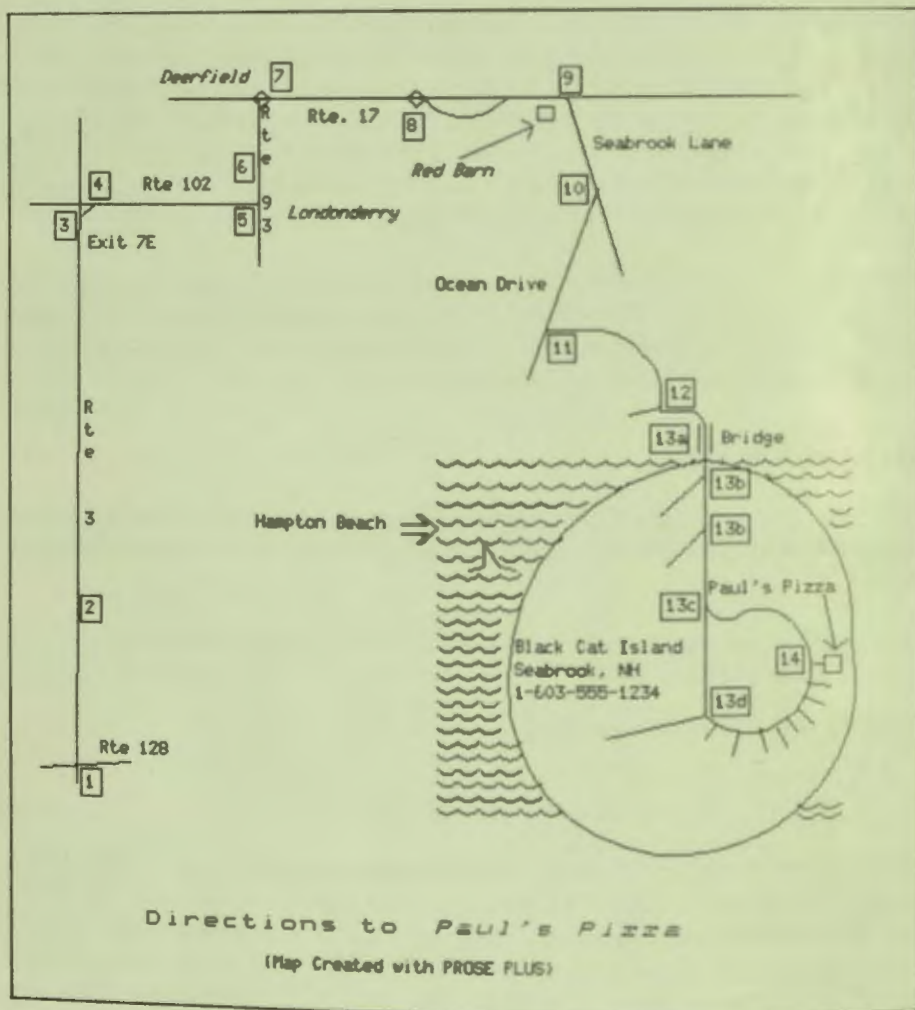
**EDITING OLD PICTURES**

Once you have exited from the graphics editor, and returned to inputting text in your document, you can still make changes to the picture using the following technique:

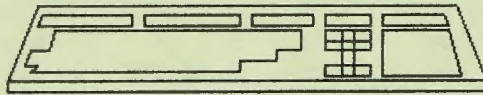
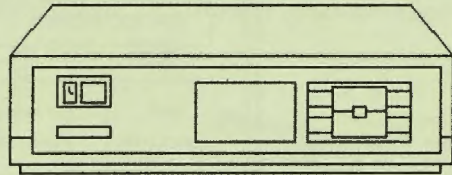
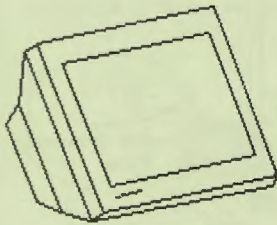
1. Move the cursor to any point on, or within, the picture reference frame in the text.
2. Press F12 to display the Edit Menu.
3. Choose “Pictures” from the menu. The screen will clear and the picture will be displayed on the graphics editor screen.

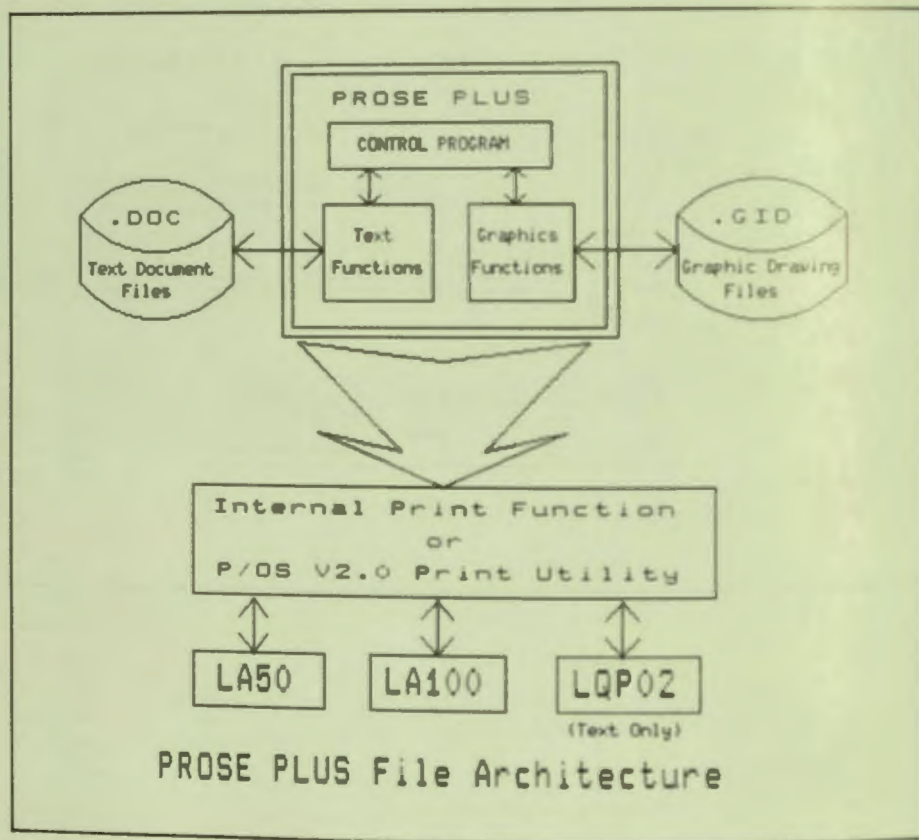
Edit the picture using the techniques described in the previous section. When you exit the graphics editor, the newly edited picture replaces the one previously referenced in the document. The old picture remains saved on your Professional, but cannot be referenced without first changing its filename through P/OS File Services. Refer to your *Professional 300 Series User's Guide for Hard Disk System* for more information on renaming files.

PICTURES CREATED USING PROSE PLUS



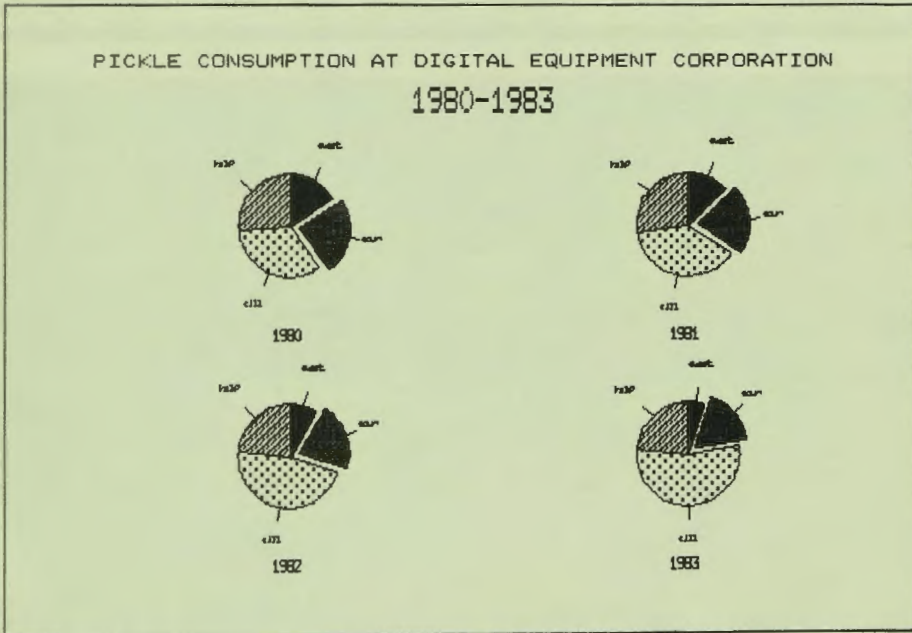
*Professional 300 Series Publications*

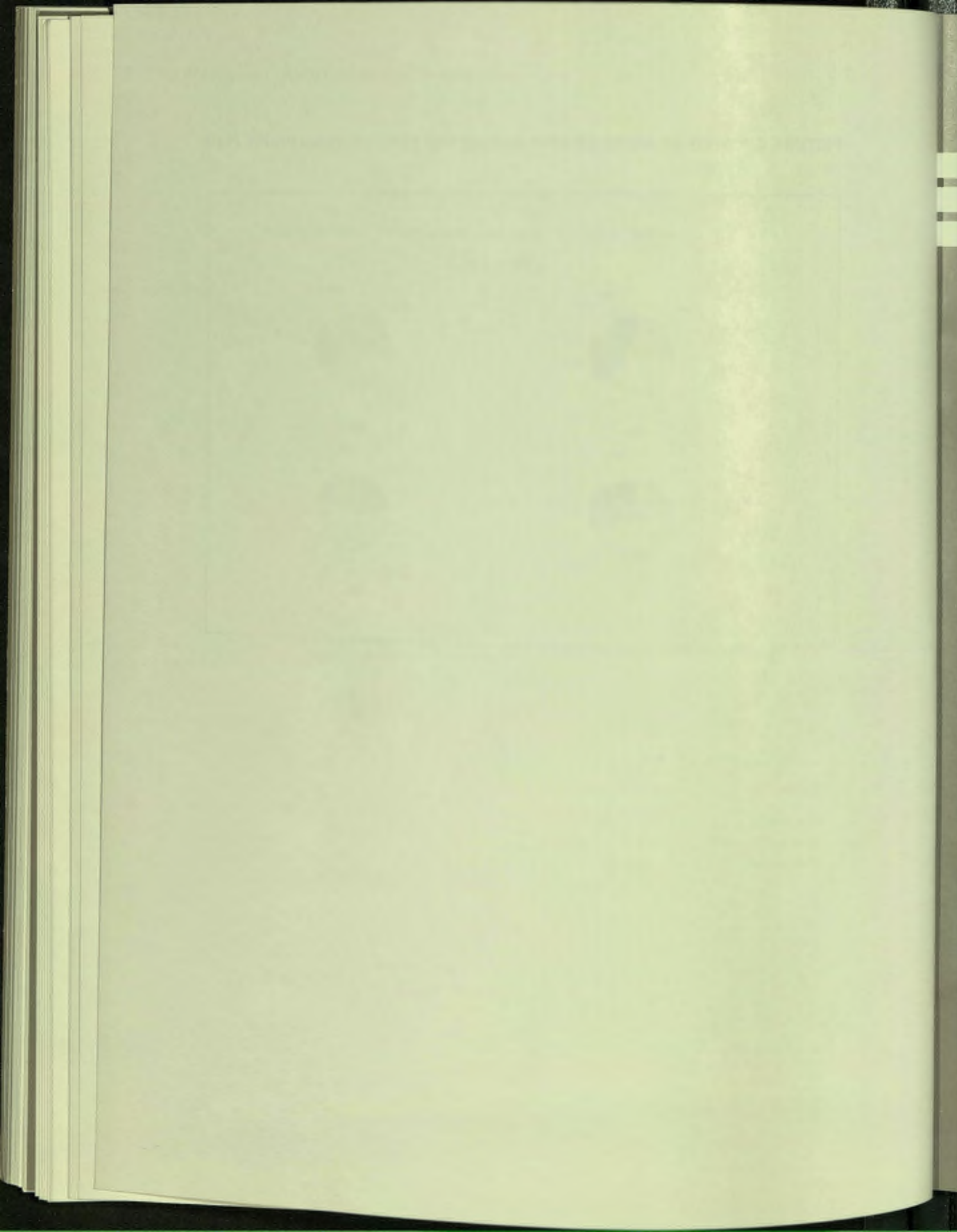






PICTURE CREATED BY ANOTHER APPLICATION AND PRINTED USING PROSE PLUS





---

# Appendix A



## Appendix A

---

### Differences between P/OS V1.7 and V2.0

There are several important differences between using PROSE PLUS on a Professional running P/OS Version 1.7 and Version 2.0. If you are running version 1.7 of P/OS note the following:

1. You cannot use the "View" option while the printer is printing a file.
2. When using the "View" option, PROSE PLUS clears the screen before each picture.
3. If you reference an externally created picture, the picture will scroll up two lines on the screen after it is drawn. This is necessary to make room for the prompt at the bottom of the screen.
4. There is no support for Mini-Exchange in Version 1.7.
5. You must set the correct printer under P/OS Set-up.



---

## Glossary





## Appendix B

---

### Glossary

**application (or application program)**

A computer program that meets some specific user need. For example, an application program can control an inventory or monitor a manufacturing process. Applications for the Professional are available from DIGITAL and other companies.

**auxiliary keypad**

The set of keys on the far right of the Professional keyboard. When you install P/OS, you use this keypad to type in the time and date. Some applications interpret these keys as special function keys; with other applications these keys do nothing.

**cursor**

In the text editor, the blinking rectangle or underscore on the screen that shows where the next typed character will appear, if appropriate. In the graphics editor, the cursor appears as crosshairs on the screen.

**default**

Information that the Professional assumes as input if you do not explicitly provide it.

**directory**

A group of files stored on a volume (a disk or diskette). A directory can include all or only some of the files on the volume. When you initialize a volume, P/OS creates a directory for you called USERFILES. You can delete this directory and you can create your own directories.

**disk**

A magnetic, file-storage medium. The disk (also called hard or "Winchester" disk) is permanently precision-mounted to its drive, and has far more storage capacity than a floppy diskette.

**diskette**

A magnetic, file-storage medium, also called floppy disk. Diskettes for the Professional are 5¼" in diameter and are encased in a protective cover. You store private and seldom-used files on diskettes, copy files onto them for safekeeping and use them to bring new programs to your Professional.

**diskette drive**

A unit that holds and operates one or more diskettes. All Professionals have a dual diskette drive capable of holding two diskettes.

**editing keypad**

The set of gray keys between the main keyboard and the auxiliary keypad. Like the keys of the auxiliary keypad, these keys can function differently for different applications.

**editor**

An application program that allows you to create and modify text files. You can change text with the Professional editor (PROSE) by choosing commands from a menu and by using function keys.

**file**

A collection of information treated as a unit. A file may contain a memo, a program, or other information. Its fundamental purpose is analogous to that of a conventional office file. A file is the means by which information is stored on disk or diskette so it is efficiently usable by both P/OS and a user. A file occupies one or more blocks on a volume and has a file name.

**form**

A request for information that is displayed on the Professional screen when P/OS requires information that you provide.

**function key**

A type of key on the Professional keyboard that, when pressed, instructs P/OS or a P/OS application to do something specific. The HELP, DO, INTERRUPT, HOLD SCREEN, PRINT SCREEN, and arrow keys perform the same function all the time. Other function keys, such as PREV SCREEN, RESUME, CANCEL, EXIT, F5, F17, and so on, have functions that depend on the application currently in use.

**install**

Describes the copying of the P/OS software from the P/OS system diskette(s) onto the disk. Or to copy a P/OS application from an application diskette onto the disk, integrating that application into the menu system.

**keyboard**

The component that enables you to communicate with the Professional. It contains the main keyboard (similar to a typewriter keyboard), a row of function keys at the top, the editing keypad, and the auxiliary keypad.

**menu**

A list of available P/OS services or functions appearing on the Professional screen. You tell P/OS what to do by making a selection from a menu.

**overtyping mode**

The way PROSE handles what you type in a file unless you use the INSERT HERE key. In overtyping mode, PROSE deletes existing text if you type over it.

**P/OS**

The Professional's operating system. The operating system manages the computer.

**program**

A sequence of instructions used by a computer to perform specific operations. A programmer writes a program in PRO/BASIC or some other programming language by first creating a program source file at the keyboard.

**prompt**

A request displayed on the screen for you to type in certain information. On menus, a prompt appears on a line below the menu list.

**PROSE**

The editor supplied with P/OS. See also editor.

**protection**

The preventing of accidental deletion of your Professional 350 files. You can protect files with "Change file(s) protection" with write-protect tabs on your diskettes, with "Back up disk file(s)" and "Copy file(s)," and by keeping diskettes and the Professional 350 in a safe place.

**scrolling**

The movement of lines of text up or down the monitor screen.

**system unit**

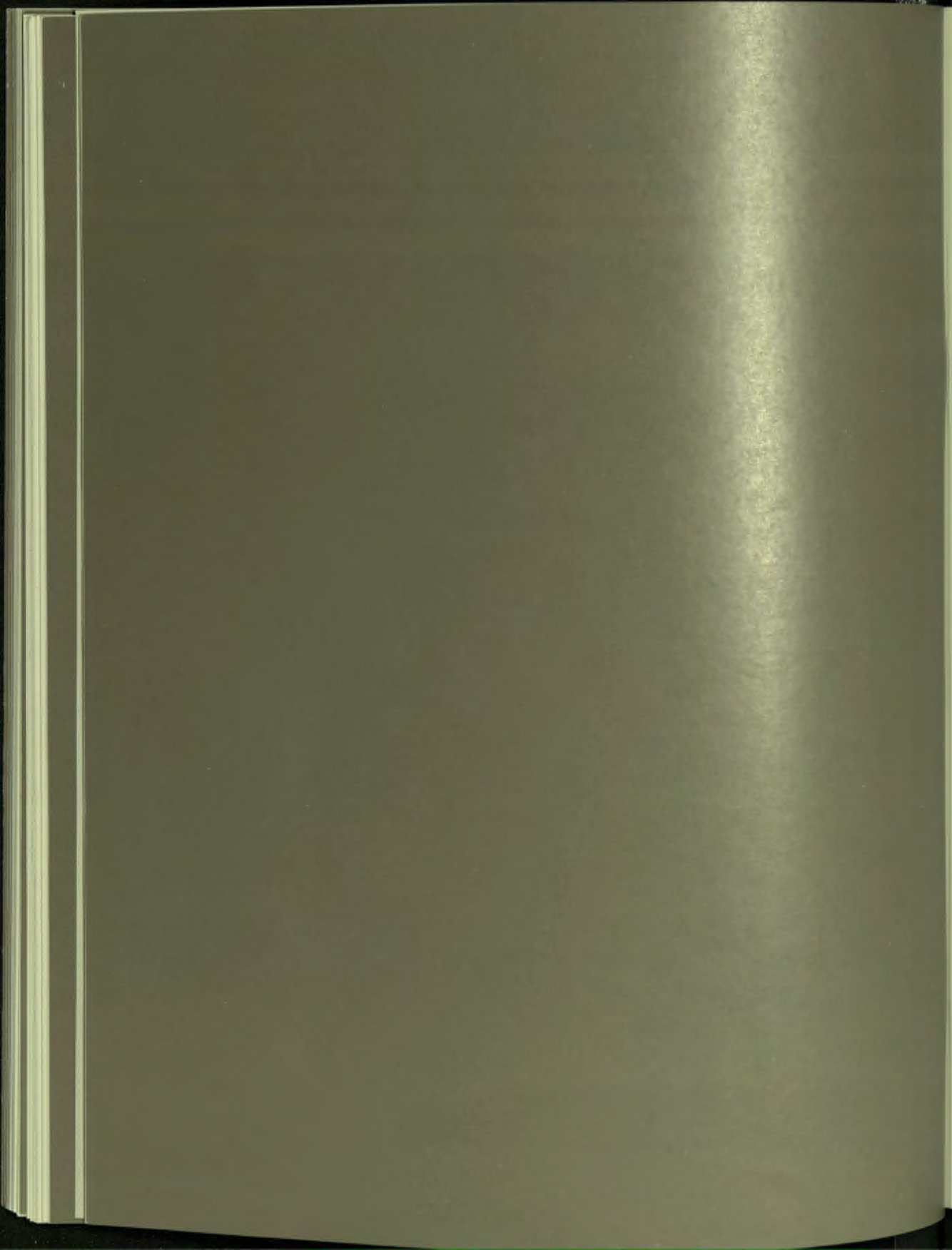
The part of your Professional that contains the diskette drive, disk, option modules, power supply, and the computer itself. The system unit is the main component of the Professional.

**word wrap**

With PROSE, the automatic dropping to the next line of any word you type that extends past the right margin of your text. You can set word wrap off or on.

---

# Index



# Index

---

- Additional options, 56
- Additional Options Menus, 20
- Advanced PROSE PLUS, 20
- Arc angles, 52
- Arcs and circles, 45
- ARROW keys, 45
- Attributes
  - Bold, 22
  - clear, 23
  - Lowercase, 23
  - Underline, 22
  - Uppercase, 23
- Attributes Menu, 21
  
- Blank lines, 14
- Blinking characters, 8
- Bold, 22
- Bottom of document, 26
  
- CANCEL key, 56
- Center line, 33
- Changing objects
  - characteristics, 53
- Characters
  - deleting, 15
- Clear attributes, 23
- Columns 80/132, 28
- Control characters, 8
- Create new document, 7
- Create work area, 26
- Cursor
  - graphics, 45
  - moving, 14, 45
  
- Define a key, 29
- Delete
  - characters, 15
  - text segments, 17
- Displays
  - graphics, 43
  - text editor, 13
- DO key, 45, 55
- Documents
  - editing, 13
  - printing, 8
- Drag, 48
- Drawing arcs and circles, 52
- Drawing lines, 50
- Drawing rectangles, 50
  
- Edit existing document, 7
- Edit Menu, 24, 44
- Editing
  - pictures, 59
- Elementary PROSE PLUS, 13
- Entering text
  - graphics, 51
- Erase picture, 56
- EXIT key, 17
- Exiting PROSE PLUS, 17
  
- F12, 42, 45
- Features, 1
- Fill text, 33
- FIND key, 20, 45, 47
- Format Menu, 33

## INDEX

- Function keys
  - EXIT, 17
  - FIND, 20
  - INSERT HERE, 15
  - NEXT SCREEN, 14, 45
  - PREV SCREEN, 14, 45
  - REMOVE, 17
  - SELECT, 31
  - TAB, 31
- Get document, 26
- Graphics, 25
  - additional features, 55
  - additional options, 56
  - CANCEL key, 55
  - DO key, 55
  - drag, 48
  - editing, 59
  - editor, 25, 41
  - Erase picture, 57
  - EXIT key, 55
  - exiting, 55
  - FIND key, 45, 47
  - getting started, 42
  - INSERT HERE key, 47
  - objects, 42
  - Overlay, 58
  - Quit, 58
  - Refresh screen, 56
  - REMOVE key, 55
  - Save, 57
  - screen, 43
  - Scroll, 59
  - select box, 58
  - SELECT key, 55
  - significant points, 45
  - Undo last remove, 57
- Groups, 53
- Help
  - graphics, 42
- HELP, 42
- INSERT HERE, 42
- INSERT HERE key, 15, 48
- Installation, 2
- Line styles, 52
- Lines, 45
- List defined keys, 31
- Lowercase, 23
- Main Menu, 6
- MAIN SCREEN key, 17
- Margin change/word wrap, 34
- Margins, 14
- Moving an object, 53
- Moving objects, 53
- Name A File Form, 7
- New page, 35
- NEXT SCREEN key, 14, 45
- Object
  - arcs or circles, 45
- Objects, 44
  - lines, 45
  - moving, 53
  - rectangles, 45
  - text, 45
- Overlay, 50
- Page break, 35
- Paginate document, 37
- Paragraphing, 14
- Personalize Menu, 28
- Picture
  - maximum size, 45
  - size, 45
  - tics, 45
- Picture reference frame, 55
- Pictures, 25, 43
  - arc angles, 52
  - arcs and circles, 52
  - changing an object, 53
  - creating, 47
  - editing, 59
  - externally drawn, 43
  - line styles, 52
  - lines, 50
  - moving an object, 53
  - objects, 40
  - other shapes, 40
  - rectangles, 50
  - referencing, 43
  - text, 51
- PREV SCREEN key, 14, 45
- Printers
  - graphics, 39
- Printing
  - format, 13
- Printing documents, 8
- Quit, 58

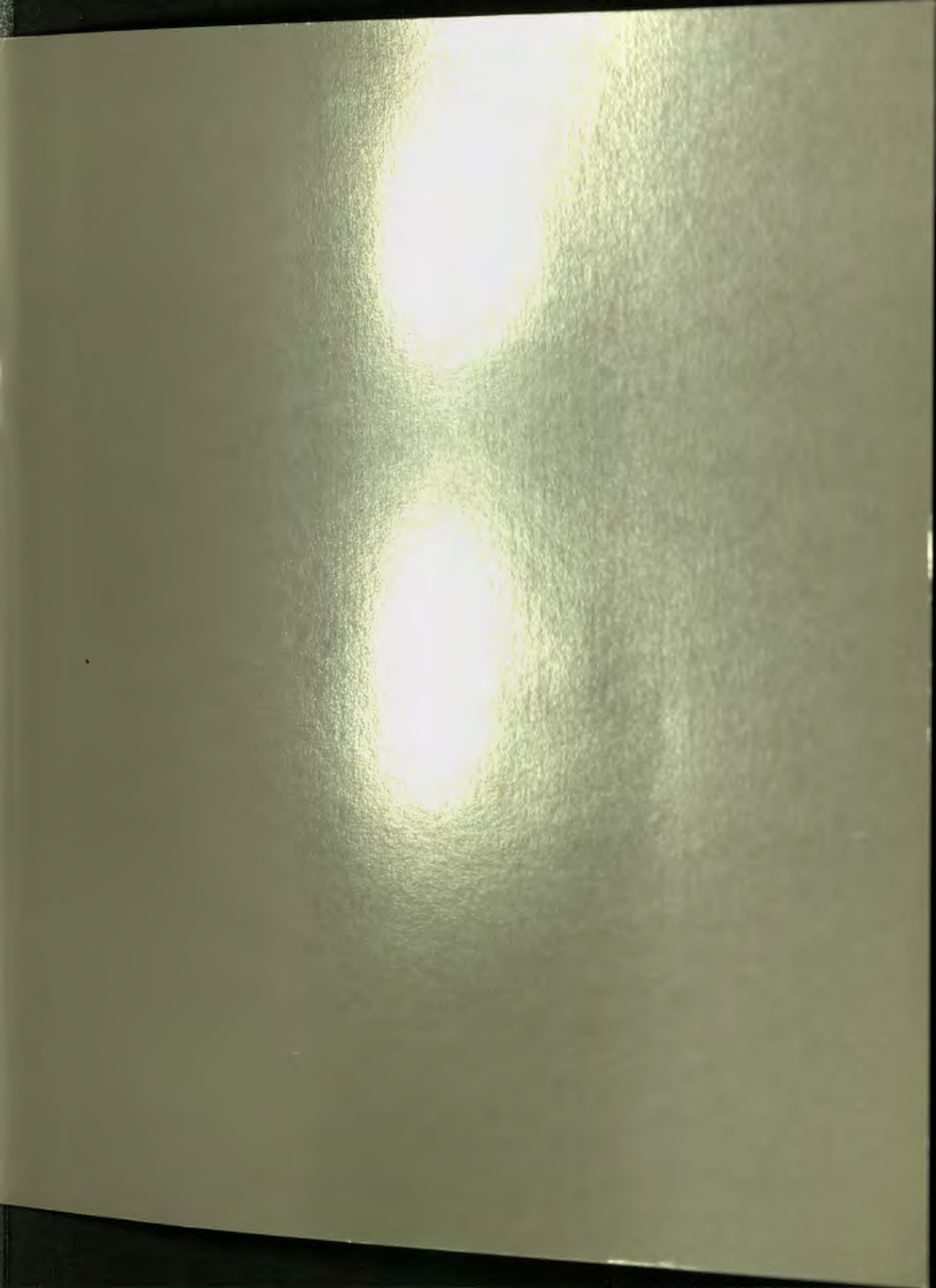


- Rectangles, 45
- Refresh screen, 56
- Relocate text, 24, 26
- REMOVE key, 17, 26, 56
- RETURN key, 14
  
- Save, 57
  - groups, 57
- Screen
  - graphics editing, 42
- Scroll, 59
- Select box, 58
- SELECT key, 31, 55
- Set tab stops, 31
- Significant points, 45
- Status line on/off, 32
- Switch work areas, 27
  
- TAB key, 14
- TAB key, 31
- Text, 45
  - italic, 51
  - picture reference frame, 55
  - sizes, 51
  - standard, 51
  - types, 51
- Tics
  - picture size, 45
- Top of document, 26
  
- Underline, 22
- Undo last remove, 57
- Uppercase, 23
  
- Word wrap
  - setting on/off, 34

AA PRO R755A  
Professional PROSE PLUS  
user's guide

**SHREWSBURY LIBRARY**  
Digital Equipment Corporation  
333 South Street SHR1-3/G18  
Shrewsbury, MA 01545  
(DTN) 237-3271

GAYLORD



digital



0211993

SHREWSBURY LIBRARY  
DIGITAL EQUIPMENT CORPORATION  
SHR1 34318  
DTN 237-3400

Printed in U.S.A.  
AA-RT55A7H

Professional™  
300 series



Developer's Tool Kit

SHREWSBURY LIBRARY  
DIGITAL EQUIPMENT CORPORATION  
SHR1-3/G18  
DTN 237-3400

DECLIT  
AA  
PRO  
P098A

digital  
software



DECLIT AA PRO P098A

PRO/RMS-11 : an introduction

Digital Equipment Corporation  
SHREWSBURY LIBRARY



94-0031049/33  
NORTHEAST TECHNOLOGY  
CENTRAL  
SOFTWARE LIBRARY  
MANUAL NO. 00318

**PRO/RMS-11:  
An Introduction**

Order No. AA-P098A-TK

November 1982

This document introduces the concepts and operations of Record Management Services (RMS-11) for the Professional Operating System (P/OS).

**OPERATING SYSTEM AND VERSION:** P/OS V1.0 or later  
**SOFTWARE VERSION:** RMS-11 Version 2.0

DECLIT  
AA  
PRO  
P098A

37962

digital equipment corporation • maynard, massachusetts

First Printing, November 1982

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright © 1982 by Digital Equipment Corporation  
All Rights Reserved.

Printed in U.S.A.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	RSX
DEC/CMS	EduSystem	UNIBUS
DECnet	IAS	VAX
DECsystem-10	MASSBUS	VMS
DECSYSTEM-20	PDP	VT
DECUS	PDT	<b>digital</b>
DECwriter	RSTS	

ZK2170

#### HOW TO ORDER ADDITIONAL DOCUMENTATION

In Continental USA and Puerto Rico call 800-258-1710

In New Hampshire, Alaska, and Hawaii call 603-884-6660

In Canada call 613-234-7726 (Ottawa-Hull)  
800-267-6146 (all other Canadian)

##### DIRECT MAIL ORDERS (USA & PUERTO RICO)\*

Digital Equipment Corporation  
P.O. Box CS2008  
Nashua, New Hampshire 03061

\*Any prepaid order from Puerto Rico must be placed  
with the local Digital subsidiary (809-754-7575)

##### DIRECT MAIL ORDERS (CANADA)

Digital Equipment of Canada Ltd.  
940 Belfast Road  
Ottawa, Ontario K1G 4C2  
Attn: A&SG Business Manager

##### DIRECT MAIL ORDERS (INTERNATIONAL)

Digital Equipment Corporation  
A&SG Business Manager  
c/o Digital's local subsidiary or  
approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532



## CONTENTS

	Page
PREFACE	
DOCUMENT OBJECTIVES . . . . .	v
INTENDED AUDIENCE . . . . .	v
ASSOCIATED DOCUMENTS . . . . .	v
CHAPTER 1	
INTRODUCTION	
CHAPTER 2	
DATA STORAGE	
DISK STRUCTURE . . . . .	2-1
CHAPTER 3	
FILE STRUCTURE	
RECORD FORMATS . . . . .	3-1
Fixed-Length Records . . . . .	3-1
Variable-Length Records . . . . .	3-2
Variable-Length with Fixed Control (VFC) Records . . . . .	3-2
Stream Records . . . . .	3-4
Undefined Records . . . . .	3-4
FILE ORGANIZATIONS . . . . .	3-4
Sequential Organization . . . . .	3-4
Special Case . . . . .	3-5
Relative Organization . . . . .	3-5
Indexed Organization . . . . .	3-6
ACCESS MODES . . . . .	3-11
Record Access Modes . . . . .	3-11
Block Access Modes . . . . .	3-12
Access Mode Switching . . . . .	3-13
FILE STRUCTURE INTERDEPENDENCIES . . . . .	3-13
File Organization and Access . . . . .	3-13
Summary . . . . .	3-15
FILE DESIGN . . . . .	3-16
CHAPTER 4	
OPERATIONS	
FILE PROCESSING . . . . .	4-1
I/O Buffers and Data Transfer . . . . .	4-2
Access Sharing . . . . .	4-3
RECORD PROCESSING . . . . .	4-3
Record Context . . . . .	4-4
Record Access Modes . . . . .	4-5
Record Transfer Modes . . . . .	4-6
BLOCK PROCESSING . . . . .	4-6
Block Access Modes . . . . .	4-7
MACROS AND SYMBOLS . . . . .	4-7
INDEX	

## FIGURES

FIGURE	1	Physical Disk Structure . . . . .	2-2
	2	Logical Disk Structure . . . . .	2-3
	3	Virtual-to-Logical Block Mapping . . . . .	2-4
	4	Fixed-Length and Variable-Length Record Formats .	3-3
	5	VFC Record Format . . . . .	3-3
	6	Sequential File Organization . . . . .	3-5
	7	Relative File Organization . . . . .	3-6
	8	Indexed File Organization -- Single Key . . . . .	3-8
	9	Indexed File Organization -- Multiple Keys . . . . .	3-9

## TABLES

TABLE	1	File Structure Interdependencies . . . . .	3-15
	2	File and Record Attributes . . . . .	3-16

## PREFACE

### DOCUMENT OBJECTIVES

This document introduces the major concepts of RMS-11: record formats, file organizations, and record access modes.

It also introduces the RMS-11 operations and defines key terms required for understanding RMS-11 capabilities and functions.

This document does not provide reference or usage information. You should read it, however, before you proceed to the other RMS-11 documents. See the Associated Documents section below.

### INTENDED AUDIENCE

This document is intended for all users of RMS-11, including MACRO-11 and high-level language programmers. Nonprogramming users, programming users who are new to the Professional Operating System, and programming users who are new to RMS-11 will find this document useful.

### ASSOCIATED DOCUMENTS

The PRO/RMS-11 Macro Programmer's Guide manual is a reference document for MACRO-11 programmers that describes the macros and symbols that make up the interface between a MACRO-11 program and the RMS-11 operation routines. Chapter 4, Operations, of this manual provides a background for the reference information.



## CHAPTER 1

### INTRODUCTION

The PDP-11 computer allows you to store large amounts of information. The combination of the software supplied by DIGITAL and the software that you write allows you to access and manipulate that information. This means that you can store, retrieve, and process (modify and delete) data according to your requirements.

All businesses, for example, need to store detailed and up-to-date personnel information. At a minimum, this information would include each employee's name, address, and social security number.

The employer will want to be able to retrieve this information in some orderly fashion, perhaps in alphabetical order by employee name or by social security number for tax purposes.

The employer also must be able to readily process the information: add information on new employees, delete information on employees who leave, and modify information that is incorrect or changes (the employee moves).

The ability to store and retrieve information and to process that information readily and in some orderly fashion implies that the information is stored in some orderly fashion, such as in records. A record is a logical unit of data, that is, an item or collection of items that are related in some manner.

The information on each employee -- name, address, and social security number -- would constitute one record. The information on each manufactured part in inventory -- part name, number, and price -- would constitute another, different type of record.

To keep records of one type separate from records of another type, records are organized into files. A file contains groups of records of the same type. One or more files, depending on the amount of data, would contain all the records of a specific type.

In a small business, all the employee records would be stored in one file; all the inventory records would be stored in another.

How the data is used helps determine how the records are stored in and retrieved from files so they can be processed, or used for a specific purpose. Data storage and retrieval is also called access.

To obtain a complete report on all employees, the employer might simply get each record, one after another. New records might be put at the end of the file, one after another. To withhold state income tax, the employer might use the address data to identify the records of employees who reside in that state.

## Introduction

The software that provides data access is RMS-11. RMS-11 is a set of routines that allows your programs to gain access to and process -- add, modify, and delete -- records and files. RMS-11 provides the connection between your program and the stored data that your program requires.

This document is an introduction to RMS-11. Data storage is described briefly in Chapter 2. File structure and access are described in Chapter 3.

The RMS-11 operation routines that provide file and record processing are introduced in Chapter 4.

## CHAPTER 2

### DATA STORAGE

The data that your programs use typically is stored on mass-storage devices, called disks. The operating system software controls these hardware devices, and allows your programs to access the data stored on them. Each device is controlled by a device driver, or software that handles the input/output (I/O) -- the writing of data into and reading of data out of storage.

The Files-11 Ancillary Control Processor (FCP) is the P/OS file control processor that catalogues and maintains files on the disks, and issues I/O requests to the device drivers.

The smallest unit of information stored on hardware devices is the bit. A bit is an area of disk or tape surface for which the magnetic orientation can be changed to one of two values, conventionally designated 0 and 1.

Information typically is grouped into units of 8 bits, called bytes. Bytes are used, for example, to represent alphanumeric characters in memory with the DEC multinational character set. Other ways of representing data, particularly numeric data, may require 2 or more bytes. A word, for example, consists of 2 bytes (or 16 bits).

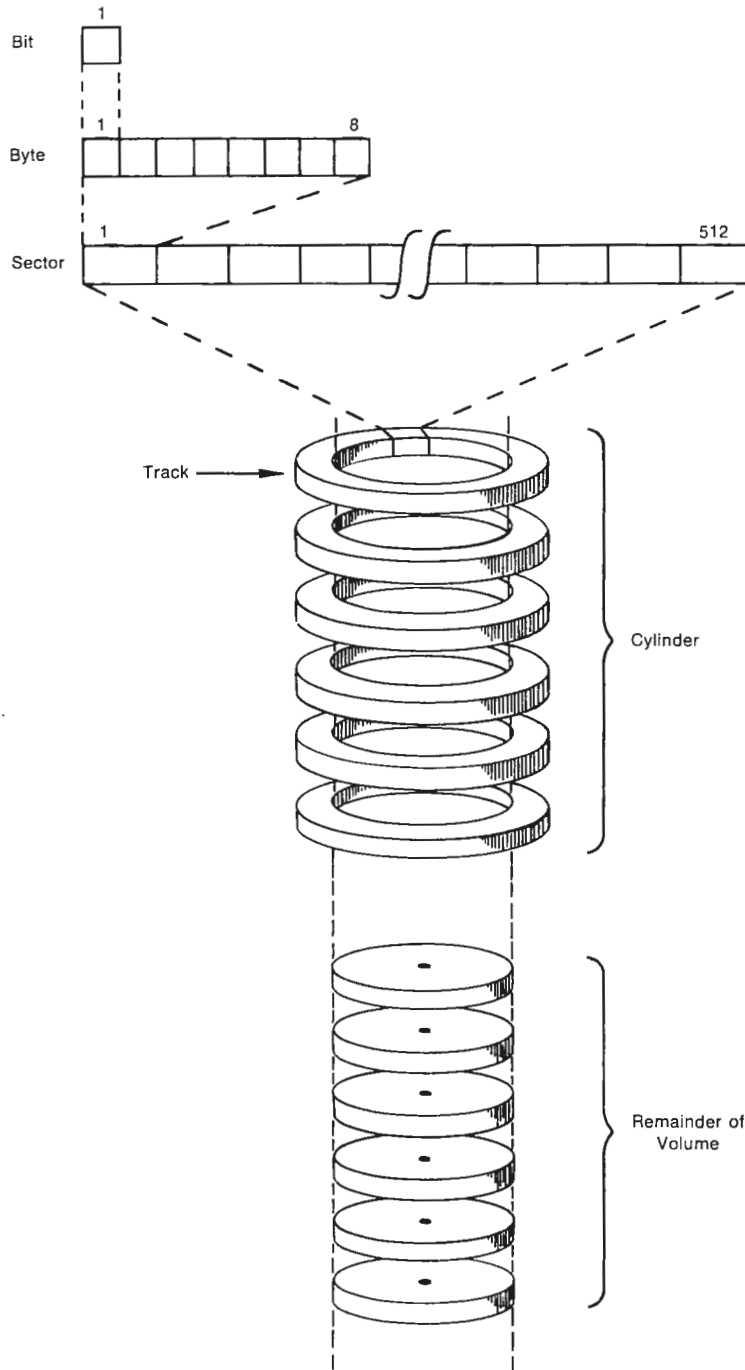
#### DISK STRUCTURE

A disk is made up of one or more circular platters, arranged one above the other to form a cylinder. A disk is mounted on a drive, which is the electromechanical unit for reading and writing data on the platters. The disk platters spin and the read/write heads of the drive move across the platters to access the data. Data is usually written on both sides of a platter.

Figure 1 shows how data is stored on a disk. On most disks supplied by DIGITAL, a sector consists of 512 bytes. A track consists of all the sectors at a single radius on one disk platter, and a cylinder consists of all the tracks at the same radius on all the platters.

The disk drive can access all tracks on a single cylinder without changing the position of the read/write heads. Any sector on a disk thus has a unique physical address: its cylinder number, its track number, and its sector number on that track.

# Data Storage Disk Structure



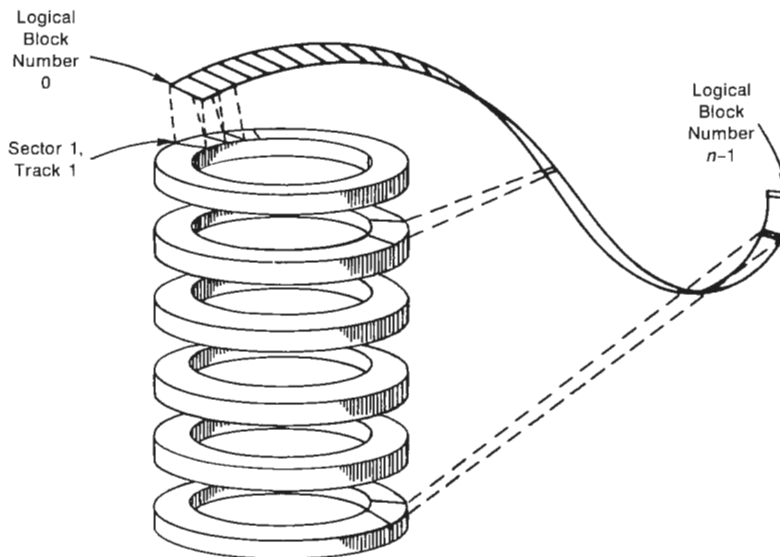
ZK-1073-82

**Figure 1: Physical Disk Structure**



## Data Storage Disk Structure

The device drivers impose a simpler logical addressing structure on each disk. They treat the disk as a single, logically contiguous, series of data units, called blocks. A block contains 512 8-bit bytes. Logical blocks are numbered sequentially, from 0 to  $n-1$ , where  $n$  is the number of blocks on the disk. Figure 2 illustrates this logical structure. Blocks, which are logically adjacent, frequently correspond to sectors, which are physically adjacent on the disk.



ZK-1074-82

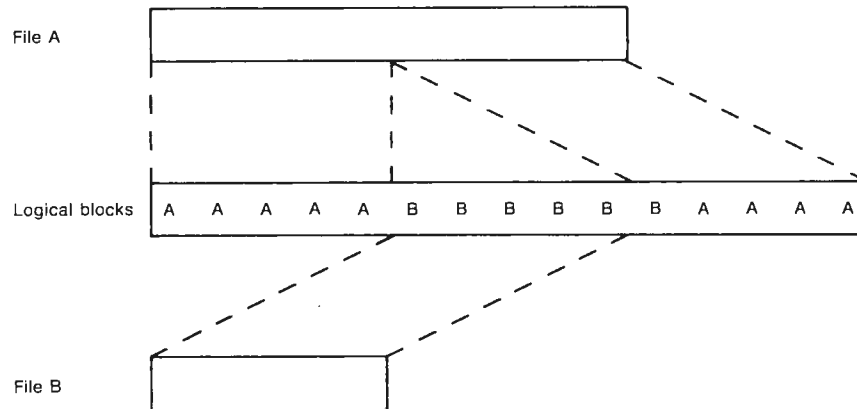
Figure 2: Logical Disk Structure

On disk, a file is simply a series of blocks, which contain your data organized into records. You can store multiple files on a disk. The file control processor treats each file as a unit, ignoring any blocks on the disk except those in the file being processed.

Figure 3 illustrates how blocks in a file -- called virtual blocks -- may be mapped to logical blocks. Virtual blocks are numbered sequentially in a file from 1 to  $n$ , where  $n$  is the number of blocks in a file.

The blocks in a file, however, need not be logically contiguous. As files are created or extended, the file control processor may allocate blocks to the file that are not next to each other on the disk. The blocks in a file, then, are virtually contiguous.

## Data Storage Disk Structure



ZK-1075-82

**Figure 3: Virtual-to-Logical Block Mapping**

Note that a virtual block number (VBN) and a logical block number (LBN) refer to the same physical unit of disk storage space. But although a virtual block also has an LBN, a logical block has a VBN only if it is allocated to a file.

To access files, the file control processor translates VBNs to LBNs and issues an I/O request to the device driver. The device driver, in turn, translates the LBNs to the physical location (cylinder, track, and sector) that is to be read or written.

Disk storage offers two main advantages: 1) it allows both sequential and random access and 2) it allows access sharing. Sequential access means that virtual blocks containing data are stored and retrieved consecutively, one after another.

On the other hand, random access (also called direct access) means that a specific block containing data can be located and retrieved without a search of all the blocks that precede it in the file. The time needed to access the data may thus be improved over sequential access.

In addition, disk storage allows access sharing. This means that more than one user can access the same disk at one time, and more than one user can be allowed to open the same file at one time.

## CHAPTER 3

### FILE STRUCTURE

The operating system software -- file control processor, device drivers -- handles files. Your programs, however, must be able to access the records within the files so they can process the data within the records.

RMS-11 allows you to define the internal structure of files (the size and arrangement of records within files) and provides operations that allow your programs to read and write records in files. RMS-11 thus provides the interface between the operating system and your programs.

You define the internal structure of a file when you create it by selecting record format and file organization.

Your selection of file organization may have to take into account the access modes that you will use to store and retrieve your data.

This chapter introduces the concepts of record format, file organization, and access modes. Chapter 4 introduces the RMS-11 operations.

#### RECORD FORMATS

RMS-11 does not handle, or process, data within records. Your program does that. However, to retrieve or store a record for your program, RMS-11 must know how large that record is. The record formats that allow you to define for RMS-11 the size of your data records are:

- Fixed length
- Variable length
- Variable length with fixed control (VFC)
- Stream

These are the standard RMS-11 record formats. In addition, you can specify undefined as the record format for non-RMS-11 files.

#### Fixed-Length Records

In a file that contains fixed-length records, every record is the same size. The size (number of bytes) is fixed at file-creation time and cannot be changed for the life of the file.

## File Structure Record Formats

Typically, fixed-length records are used to contain data that is always the same length. This is the most efficient use, although your program need not supply data for every byte of a fixed-length record. RMS-11 will simply store and return the fixed-size record regardless of whether every byte in the record contains data that is meaningful to you. Your program must be able to recognize that some bytes in the record are "unneeded."

For example, you could use a fixed-length record to contain information on manufactured parts including, say, part number, date of manufacture, and price. Each of these fields would always be the same length and would always be filled.

If you also used a field for a part name, that field would vary in length with the names of the parts. However, as long as most of the fields contain data whose length does not vary, fixed-length record format may still be the most efficient way to store your data.

### Variable-Length Records

In a file that contains variable-length records, records can be of different lengths, up to a maximum size that you specify. This maximum is fixed at file-creation time and cannot be changed for the life of the file.

Because the number of bytes of data per record is not fixed, each variable-length record is only as long as it needs to be to contain the record data. To keep track of the length of each record for reading and writing, RMS-11 maintains a length field which is prefixed to each record.

Thus, a variable-length record will require slightly more space than a fixed-length record to hold exactly the same amount of data. However, if the number of bytes of data in a record must vary, fixed-length records would waste space in unneeded bytes, and variable-length records may therefore be more efficient.

Figure 4 compares the fixed-length and variable-length record formats.

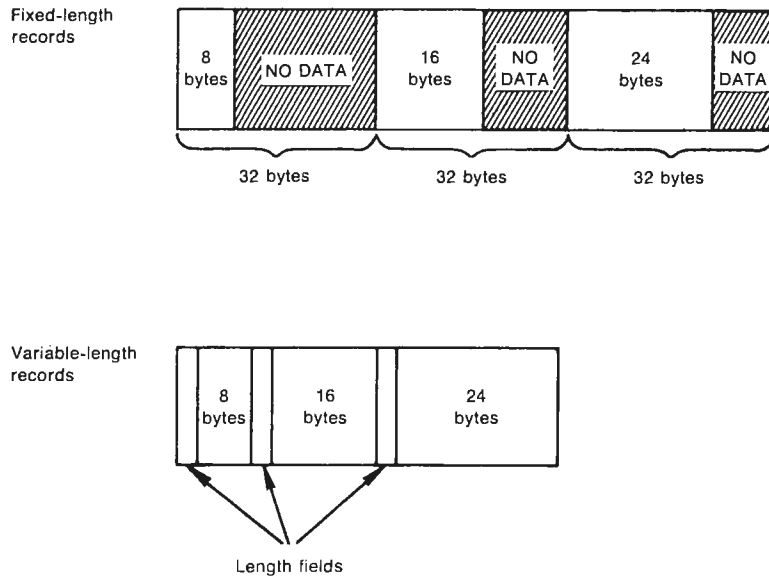
Typically, variable-length records contain text data that varies in length: for example, the names and addresses of employees. Using fixed-length records for this information could result in many unneeded bytes and inefficient use of storage space.

### Variable-Length with Fixed Control (VFC) Records

A VFC record is much like a variable-length record except that in each record in the file a fixed-length control area precedes the variable-length data. This format allows you to construct records with additional data that labels, or identifies, the contents of the variable-length portion of the record.

For example, in a text file, each line of text constitutes the variable-length portion of a record. The fixed-length control area of each record contains a line number that indicates the sequence in which the line occurs in the text file: line 3, say, in a text file of 10 lines (that is, 10 records). An editing program can use the line numbers to locate a particular line of text easily in the file.

## File Structure Record Formats



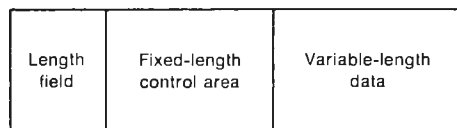
ZK-1077-82

**Figure 4: Fixed-Length and Variable-Length Record Formats**

At file-creation time, you specify two sizes for the records in the file: the maximum size permitted for the variable-length portion and the fixed size of the control area.

As for variable-length records, RMS-11 keeps track of the length of each record by maintaining a length field for each record. Thus, in memory, a VFC record consists of a length field, a fixed-length control area, and variable-length data.

Figure 5 illustrates the VFC record format.



ZK-1078-82

**Figure 5: VFC Record Format**

## File Structure Record Formats

### Stream Records

A stream record consists of a continuous series of ASCII characters delimited by a special character or sequence of characters called terminators: form feed, line feed, or vertical tab are all terminators.

For example, a text file might consist of a series of stream records terminated by carriage-return/line-feed characters pairs.

RMS-11 considers each record a series of bytes; the length of a stream record is determined by the position of the terminator.

### Undefined Records

A file with undefined records either may have no record format or may contain records that are not in one of the four standard RMS-11 formats described in the previous sections.

RMS-11 considers a file with undefined records as a series of blocks; that is, it stores and retrieves data in units of 512-byte blocks. Your program must be able to interpret the contents of the blocks.

RMS-11's support of both undefined and stream records provides compatibility with non-RMS-11 files. This means that, with some limitations, RMS-11 can process files created under other file systems.

## FILE ORGANIZATIONS

The arrangement of records within files directly affects access flexibility, or how quickly and easily RMS-11 can access those records. Your selection of file organization, therefore, should take access mode into consideration. For more information on access mode selection, see the Access Modes section and the File Structure Interdependencies section.

RMS-11 makes three file organizations available:

- Sequential
- Relative
- Indexed

### Sequential Organization

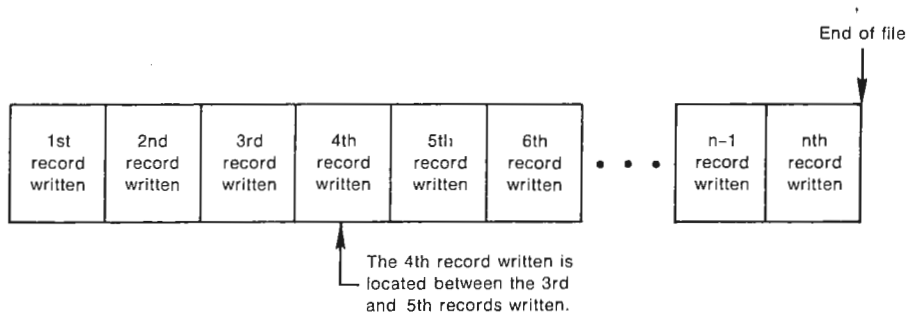
In a sequential file, records normally are arranged within the file in the order in which they were written; that is, the first record written is the first record in the file, and the record written most recently is the last record in the file.

## File Structure File Organizations

You can add records to a sequential file only at the logical end of the file. You can delete a record or series of records in a sequential file only by truncating the file from a specific point to the logical end of the file, which effectively deletes the record(s). You can, however, update a record -- retrieve it, change its contents, and rewrite it in the file -- as long as the length of the record has not changed.

Sequential file organization is supported for all record formats. In addition, sequential files allow shared access for reading data.

Figure 6 illustrates sequential file organization.



ZK-1079-82

Figure 6: Sequential File Organization

Special Case - In a sequential file with fixed-length records, the records are numbered consecutively from 1 to  $n$ , where  $n$  is the number of records in the file. The record numbers are known as relative record numbers. In this case, records may not be arranged in the file in the order in which they were written. You can, for example, specify that relative record number 5 be written before relative record number 4. When relative record number 4 is written later, it will be inserted before relative record number 5. Fixed-length records in a sequential file can be retrieved by random access.

In addition, RMS-11 supports the use of unit-record devices (such as terminals and line printers) by treating them as sequential files. That is, your program can write to and read from a unit-record device in much the same manner as it would to a sequential file.

### Relative Organization

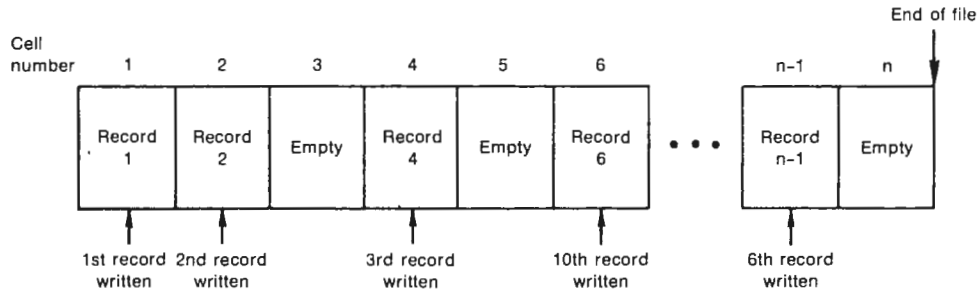
A relative file consists of a series of cells, or fixed-length units of storage. Each record within a relative file is stored in a cell; however, not every cell need contain a record. The cells are numbered consecutively from 1 to  $n$ , where  $n$  is the number of cells in the file. As for sequential files with fixed-length records, the cell numbers are known as relative record numbers.

To randomly access a record in a cell, your program must keep track of the relative record number. One way to do so is to associate a value within the record (for example, an order number) with the relative record number.

## File Structure File Organizations

Because records are stored by relative record number, they need not be arranged within the file in the same order in which they were written. For example, order number 4 may be completed and its record written to the file before order number 3. Thus, order number 4 will be stored in cell 4, and cell 3 will remain empty until order number 3 has been completed and its record written to the file.

Figure 7 illustrates relative file organization.



ZK-1080-82

Figure 7: Relative File Organization

The records in a relative file can be fixed length, variable length, or VFC. Since there can be only one record per cell, RMS-11 determines the cell size based on the maximum record size that you specify. This means that with variable-length and VFC records there may be unneeded bytes in a cell (much like fixed-length records in which there may be unneeded bytes).

Unlike sequential files, relative files permit the following capabilities:

- Random access by relative record number, regardless of record format
- Record deletion
- Access sharing for both reading and writing data

### Indexed Organization

In an indexed file, records are arranged in ascending order by key. A key is a data field within the record that RMS-11 uses to determine the order in which to access the records in the file. This is the only case in which RMS-11 interprets data within records; thus, a record can be identified by its contents, instead of by its position within the file.



## File Structure File Organizations

When you create an indexed file, you must define one field of the record as the primary key. A key is defined by its location within the record, its data type, and its length. When a record is stored in that file, RMS-11 inserts the record in order based on the value that it finds in the primary key field; that is, RMS-11 inserts the record after a record with a lower or equal value in the primary key field, and before a record with a higher value in the primary key field.

You can optionally define other fields of a record as alternate keys. These keys specify alternate access orders for the retrieved records; they do not affect the order in which the records are arranged within the file.

The data type of values in key fields can be byte (character) string, signed integer, unsigned binary number, or packed decimal number.

For each data field defined as a primary or alternate key, RMS-11 constructs an index. A primary index contains the values in the primary key fields, and an alternate index contains the values in alternate key fields.

An indexed file is a hierarchical, or tree, structure consisting of levels of records. The highest level(s) are the index records. The data records comprise the lowest level. An indexed file consists of at least two levels of records: index and data. Each index record contains the highest key value in a group of records at the next lower level plus a pointer to that group of records.

Thus, each key value provides a logical access path to locate a specific record or set of records in a file. The indexes also allow your program to retrieve your records in a specific order. RMS-11 stores the indexes in the file itself.

For example, to maintain employee files in alphabetical order by employee name, you could create an indexed file and specify the employee-name field of the record as the primary key. RMS-11 would construct a primary index consisting of employee names, in alphabetical order, and insert the records in the file in that order.

Figure 8 illustrates an indexed file with only a primary key defined; note that the data records contain other fields, including address and badge number.

In many cases, employee records may need to be retrieved in different orders; for example, the payroll may be processed by badge number rather than by employee name. Thus, you might want to define the badge-number field as an alternate key for the file.

Figure 9 illustrates an indexed file with an alternate key defined.

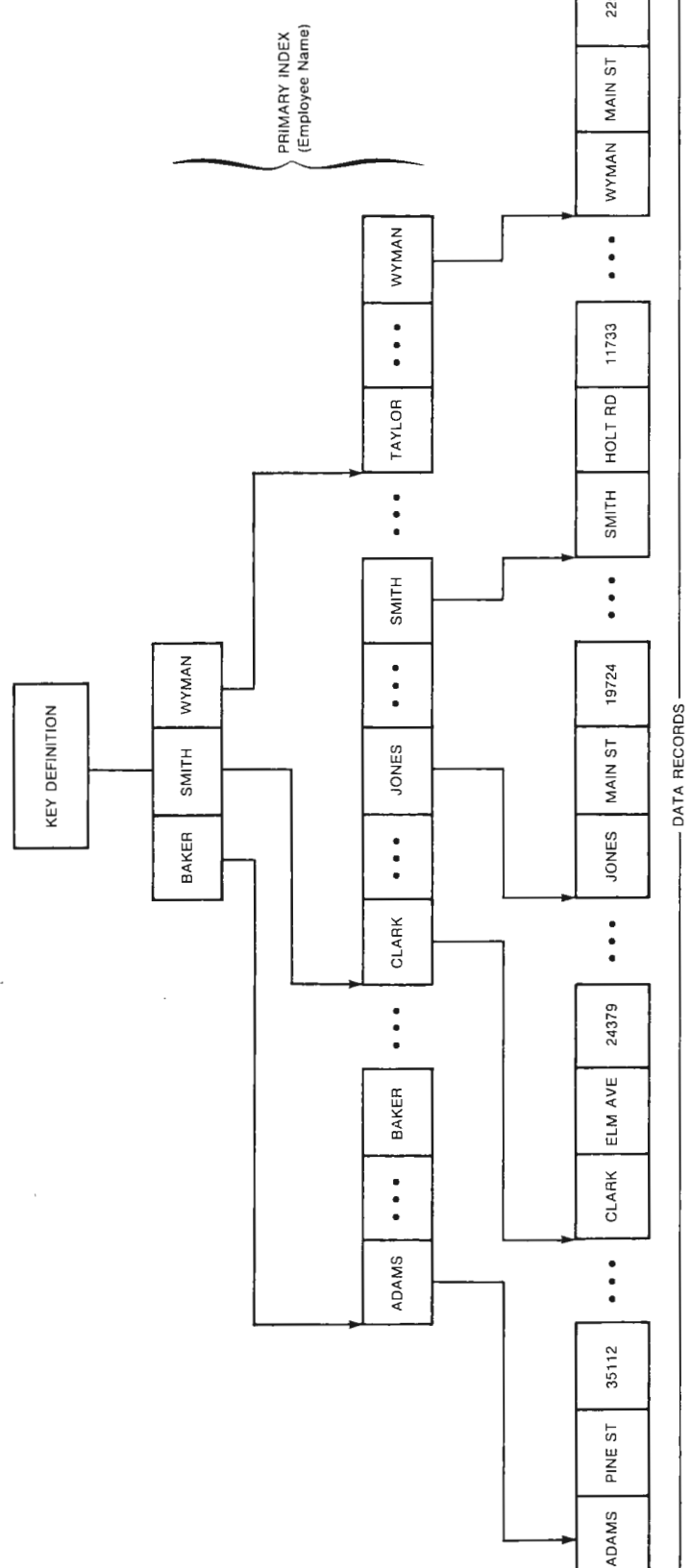


Figure 8: Indexed File Organization -- Single Key

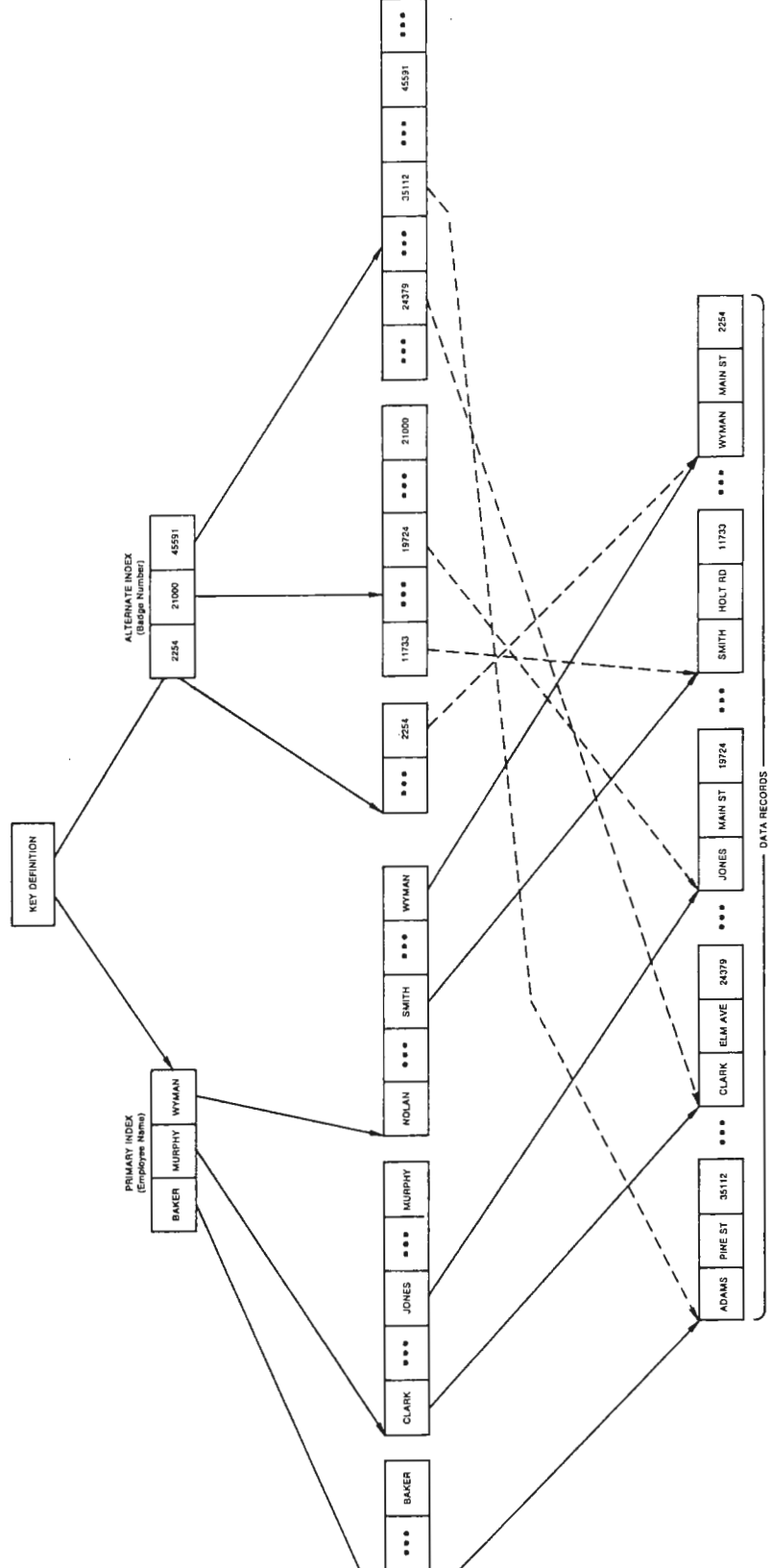


Figure 9: Indexed File Organization -- Multiple Keys

## File Structure File Organizations

You can optionally specify three key value characteristics for individual keys:

1. Duplicate values for primary and/or alternate keys.

This means that more than one record in the file can have the same value for the key.

In the employee file example, duplicate key values could be allowed for employee name but not for badge number.

By default, RMS-11 does not allow duplicate values for primary keys but does allow duplicate values for alternate keys.

2. Changeable values for alternate keys.

This means that the data in the key field can be modified. For example, if employee address was also an alternate key in the employee file, key values would be allowed to change in case the employee moves.

If the field does change, RMS-11 updates the appropriate index to reflect the new key value.

By default, RMS-11 allows alternate key values to be changed. Note, however, that alternate key values can change only if duplicate key values are also allowed. Primary key values cannot be changed.

3. Null values for alternate keys.

This means that the data in the key field can be null, that is, have no value. It is possible, for example, that employee information may be incomplete when the record is put into the file, and that one of the incomplete fields is a key.

RMS-11 makes no alternate index entry for such a record; if the record is subsequently modified to include the information, RMS-11 makes an entry in the appropriate index. Note that primary key values cannot be null, nor can a null value later be changed unless the key was specified to be changeable.

By default, RMS-11 treats any value in the key field as a true key value.

When keys are character strings, you can also specify that keys are segmented; that is, the keys consist of separate fields of the record. For example, you can define a key in an employee file as consisting of both the employee-name and badge-number fields. RMS-11 will concatenate the segments to form the key.

Indexed files permit fast sequential access (in key order) and random access by key value. Indexed files allow only fixed- and variable-length record formats.

## File Structure File Organizations

### ACCESS MODES

The access modes are the methods that RMS-11 uses to store or retrieve the contents of files. The contents of files can be either records or blocks.

The record access modes are:

- Sequential
- Random by record file address (RFA)
- Random by key

The block access modes are:

- Sequential
- Random by virtual block number (VBN)

In the sequential access modes, RMS-11 stores or retrieves records or blocks consecutively, one after another.

Sequential access is efficient when you need to read all the records or blocks in a file. For example, you would use sequential record access mode with an employee file to perform weekly payroll processing because you would always want to access all of the employee records.

In the random access modes, RMS-11 stores or retrieves records or blocks directly, using an identifier unique to the required record or block. This means that your program, not the sequence of records or blocks within the file, establishes the order in which records or blocks are processed.

Random access is efficient when you need to "jump around" in a file, rather than access all the records or blocks one after another. For example, if your program needs to process only the records for employees who receive sick pay during a payroll period, you would use a random record access mode to access only those records, by employee name.

### Record Access Modes

For sequential access, record storage or retrieval begins at a point in the file and continues with consecutive records through the file. Your program issues a series of requests to RMS-11 to successively retrieve the next record in the file.

Sequential access does not allow you to backspace through a file; you must reopen the file, or rewind to the first record in the file, and begin again at the first record.

For random by RFA access, RMS-11 uses the record file address (RFA) as an identifier to gain direct access to a specific record in a file, without a successive search of all the records that precede it. The RFA is a unique record identifier that RMS-11 establishes for every record that it writes to a file.

When RMS-11 stores a record in a file, it establishes the RFA for that record and returns the RFA information to your program. Your program can then use the RFA to retrieve the record.

## File Structure Access Modes

Note that because only RMS-11 can establish the RFA, you cannot store a record by RFA (that is, you cannot specify an RFA for the record).

RFA access can be used with files of any organization and record format; it is the fastest way to read a record randomly in an indexed file.

For random by key access, your program specifies an identifier that allows RMS-11 to gain direct access to a specific record without a successive search of all the records that precede it. For sequential files with fixed-length records or for relative files, the identifier is a relative record number (RRN). For indexed files, the identifier is a key value.

To randomly read a record in an indexed file, your program specifies a key value and the index (primary, first alternate, and so on). RMS-11 searches the index to locate the record with the specific key value. For byte (character) string keys, your program can also specify the number of characters on which the match is to be made; this is called generic match.

In addition to specifying a key value or RRN, you can also specify match criteria:

- An exact match on the key value or RRN, that is, only the record with the specified key value or RRN
- A greater-than match on the key value or RRN, that is, the next record after the record with the specified key value or RRN
- A greater-than-or-equal-to match on the key value or RRN, that is, either 1) the record with the specified key value or, if there is no such record, the record with the next higher key value; or 2) the record in the cell with the specified RRN or, if that cell is empty, the first record in a cell after the specified RRN

For example, in an employee file the second alternate key might be the social security number. If you want to read the first record that contains a social security number beginning with 175, you would specify key number 2 (the index), key value 175, and that the match is to be an exact match on the first three characters of the key.

### Block Access Modes

For sequential access to blocks, RMS-11 stores or retrieves data as a consecutive series of blocks. Your program issues a series of requests to RMS-11 to successively access the next block(s) in the file.

For random by VBN access, your program specifies the virtual block number (VBN) as the identifier of the first block to be accessed. RMS-11 uses the VBN to gain direct access to the specified block(s) without a successive search of all the blocks that precede it.

## File Structure Access Modes

### Access Mode Switching

RMS-11 allows your programs to change access modes -- either among the three record access modes or between the two block access modes -- at any time during file processing, as long as the file organization allows the access modes selected.

However, you must make the basic selection of either record access or block access when you open the file. You can switch between record and block access only by closing and reopening the file.

Access mode switching is most useful when a random access mode is used to retrieve the first record or block of a series, and then sequential access mode is used to retrieve subsequent records or blocks of that series.

For example, if an indexed employee file uses the department code as one of the alternate keys, your program could print a report on all the employees in a specific department. It would do this by using key record access mode to locate the first record whose department field matches the specified key and then switch to sequential record access mode to consecutively retrieve all the remaining records of the employees in that department.

### FILE STRUCTURE INTERDEPENDENCIES

As the previous sections indicate, your selection of record format, file organization, and access mode(s) cannot be made independently of each other. For example, if you want to use VFC records, you cannot choose indexed file organization; and if your records are undefined, you must use a block access mode.

### File Organization and Access

You can use sequential record access mode with any RMS-11 file organization. The organization of the file determines the order in which the records are stored and retrieved.

You can use random by RFA record access mode and VBN access mode with any file organization. For random by RFA record access mode, your program must save the RFAs established by RMS-11 when it stores the records in the file.

Random by key record access mode also can be used with any file organization. For sequential files, however, the record format must be fixed length.

You can use sequential block access mode with a file of any organization, including non-RMS-11 files.

The block access modes may improve access time because RMS-11 ignores file structure and record characteristics. Your program, however, must be able to interpret the contents of the blocks.

**Sequential Access to Sequential Files.** If you use sequential record access mode with a file of sequential organization, your program must open the file and read through the records one after the other beginning with the first record. Even though the record you require may be near the end of the file, your program must read every record that precedes it. In addition, records can be added sequentially only at the end of a sequential file.

## File Structure File Structure Interdependencies

**Sequential Access to Relative Files.** If you use sequential record access mode with a relative file, RMS-11 will search for successive record cells, ignoring cells that are empty. For example, if a relative file contains records in cells 1, 2, 4, and 8, and all other cells are empty, RMS-11 will check each cell from 1 through 8 but will read only RRNs 1, 2, 4, and 8, in that order.

To retrieve a specific record in a relative file, using sequential access, your program first must retrieve all the records in all the preceding cells.

To insert a record sequentially in a relative file, RMS-11 puts the record in the next cell after the cell just accessed (the cell with an RRN one number higher than the current cell), if that cell is empty. If the cell contains a record, RMS-11 will return an error code.

**Sequential Access to Indexed Files.** If you use sequential record access mode to retrieve records in an indexed file, you must specify a key for RMS-11 to use to establish the order in which it will read the records sequentially. The key you specify can be the primary key or any alternate key. You can use sequential record access mode to retrieve records in the order represented by any index for the file.

RMS-11 will retrieve the records in ascending order by key value; that is, if you specify the employee-name key, all employee records will be read beginning with the A's.

To store, or insert, a record sequentially in an indexed file, RMS-11 uses the primary key value to insert the record in order in the file, and updates the primary and alternate indexes according to the key fields within the record.

In a series of sequential insertions, RMS-11 verifies that the primary key of each new record has a value equal to or greater than the primary key of the previously inserted record.

**Random Access to Sequential Files.** If you use random by key record access mode with a sequential file, the record format must be fixed length. Your program specifies the RRN of a record in the file and the match criterion, and RMS-11 stores or retrieves that record.

When you are reading a record from a file, RMS-11 will retrieve that record or return an error code if the specified record position lies beyond the end of the file.

Note that RMS-11 does not initialize sequential files. Thus, if you attempt to read a record from a sequential file with fixed-length records to which records have been randomly written, you may get "garbage" data. That is, if you specify an RRN that is not beyond the end of the file but at which no record has been stored, RMS-11 will return whatever data happens to be at that location.

When you are writing records to a file and specify an RRN, RMS-11 will store the record in that RRN's position in the file regardless of whether a record with that RRN already exists. If the specified record position lies beyond the end of the file, RMS-11 will extend the file and store the record at the designated RRN.

If you know the RFA, you can use random by RFA record access mode to read sequential files. This is the only random record access mode in which your program can read sequential files of variable-length or VFC record format.



File Structure  
File Structure Interdependencies

Random Access to Relative Files. In random by key record access mode, your program specifies the relative record number of a record's cell in the file. RMS-11 then stores or retrieves the record in that cell.

When you are reading records from a file and specify an exact match on an RRN, RMS-11 will retrieve the record in that cell or return an error code if the cell is empty.

When you are writing records to a file, RMS-11 will store the record in the cell if the cell is empty or return an error code if the cell contains a record. Only on an UPDATE record operation (see Chapter 4) will RMS-11 write a record into a cell that already contains a record.

If you know the RFA, you can use random by RFA record access mode to read relative files.

Random Access to Indexed Files. In random by key record access mode, your program supplies the key value of a record in the file. RMS-11 then retrieves the record associated with that key value.

If you know the RFA, you can use random by RFA record access mode to read indexed files. RFA access is faster than key access because it bypasses the searching of the index, but your program must keep track of the RFAs.

Summary

Table 1 summarizes the RMS-11 file structure interdependencies.

Table 1: File Structure Interdependencies

Device	File Organization	Record Access Mode	Record Format
Disk	Sequential	Sequential, Random by RFA	Fixed, Variable, VFC, Stream
	Relative	Random by key	Fixed
	Indexed	Sequential, Random by key, Random by RFA	Fixed, Variable, VFC
		Sequential, Random by key, Random by RFA	Fixed, Variable

NOTE: The block access modes can be used to access files of any organization and any record format. A block access mode must be used to access a file with undefined record format.

**File Structure**  
**File Structure Interdependencies**

**FILE DESIGN**

To create a file that will allow your program to store, retrieve, and process your data records readily and efficiently, you must specify to RMS-11 the file and record attributes. The attributes consist of the device, file organization, and record format, plus additional values that describe the characteristics of your files and records.

You can design files with the attributes you require and create them by using the CREATE operation routine (see Chapter 4, Operations).

RMS-11 requires this information to access your data. The information is stored in the file directory and, for relative and indexed files, in the file prologue, and is passed to RMS-11 as required for the file and record operations. The prologue consists of the first blocks of a relative or indexed file and is used to store specific relative or indexed file attributes that cannot be stored in the file directory.

Table 2 lists the RMS-11 file and record attributes.

Table 2: File and Record Attributes

Attribute	Explanation
<b>File Attributes:</b>	
File specification	System node, device, account, file name, extension, and version
File organization	Sequential, relative, or indexed
Allocation	Initial size of the file, in blocks
Extension	The number of blocks to be added to the file each time its size must be increased
Bucket size <sup>1</sup>	For relative and indexed files, the number of blocks per bucket
Contiguity	Whether disk space is to be allocated to the file in continuous, adjacent logical blocks
Location	Where the file is to be physically placed on the disk
Protection	The type of access different categories of user may have
Maximum record number	For relative files, the maximum number of records the file can contain

(continued on next page)

1. See the File Processing section in Chapter 4 for more information.

File Structure  
File Design

Table 2: File and Record Attributes (Cont.)

Attribute	Explanation
<b>Record Attributes:</b>	
Record format	Fixed length, variable length, VFC, stream, or undefined
Record size	The size of the record in bytes: for variable-length and stream records, record size is a maximum length; for fixed-length records, record size is the exact length of each record; for VFC records, record size is the maximum allowable length of the variable portion
Control field size	For VFC records, the size in bytes of the fixed-length portion of the record
Block spanning <sup>1</sup>	For sequential files, the ability for records to span physical block boundaries
Record-output handling	Carriage control: how a record is to be treated when written to a terminal or line printer
<b>Indexed File Key Attributes:</b>	
Key number	The number of the key
Key position	The location of the key within the record
Key size	The length of the key, in bytes
Key data type	Byte (character) string, signed integer, unsigned binary number, or packed decimal number
Key name	The name for the key (optional)
Duplicate key values	The ability to have more than one record in the file with the same primary or alternate key value

(continued on next page)

1. See the File Processing section in Chapter 4 for more information.

File Structure  
File Design

Table 2: File and Record Attributes (Cont.)

Attribute	Explanation
<b>Indexed File Key Attributes (Cont.):</b>  Changeable key values  Null key values  Segmented keys  Bucket fill size <sup>1</sup>	  The ability to modify alternate key values  The ability to have null alternate key values  The ability to use separate fields of the record as the key value  The level to which buckets are to be loaded with records

1. See the File Processing section in Chapter 4 for more information.

## CHAPTER 4

### OPERATIONS

The RMS-11 routines operate either on a file as a whole or on a record or block within a file. The operation routines allow your programs to add, retrieve, modify, and delete files, records, or blocks.

At the file processing level, RMS-11 makes requests of the file processor to access files, by means of the directory and file operations.

At the record processing level, RMS-11 stream and record operations provide access to individual records.

At the block processing level, RMS-11 block operations provide access to the individual virtual blocks of a file, regardless of file organization and record format.

#### FILE PROCESSING

The RMS-11 directory and file operations are the interface between your program and your data files.

The directory operations affect only file specification entries in directories. RMS-11 operations construct and use file specification strings and file identifiers, including wildcard file specification strings. The directory operations are:

- ENTER -- places a file specification in a directory
- REMOVE -- deletes a file specification from a directory
- RENAME -- replaces an existing file specification with a new one
- PARSE -- returns file specification information to your program
- SEARCH -- examines one or more directories for a specified file and returns the file specification and location in a form that can be used by other directory operations or by file operations

## Operations File Processing

The file operations provide access to files as whole entities (that is, they do not provide access to records within files). These operations are:

- CREATE -- creates a new file with the attributes you specify and opens it for processing
- OPEN -- makes an existing file available for processing
- CLOSE -- terminates access to a file
- ERASE -- deletes a file and removes its directory entry, if one is specified
- EXTEND -- increases the allocated size of an open file
- DISPLAY -- returns file information to your program

### I/O Buffers and Data Transfer

For each file that is open for processing, RMS-11 requires that you provide at least one I/O buffer for its internal use. RMS-11 either transfers the data from disk to an I/O buffer, or from an I/O buffer to disk.

Data in sequential files is transferred in blocks. The user can define whether records in files can cross block boundaries. If you specify block spanning when you create a file, records may continue across block boundaries. This means that they can be stored most efficiently within a file.

When records are restricted by block boundaries, they must be less than or equal to 512 bytes. Unneeded bytes may remain after each record to the end of that block if the next record will not fit within those bytes.

Data in relative and indexed files is transferred in buckets. A bucket consists of one or more blocks that RMS-11 treats as a unit. The bucket size, or number of blocks in a bucket, is user specified when a file is created, and cannot be changed unless the file is redesigned. Although records may span block boundaries, they cannot span bucket boundaries.

For indexed files, you can also specify a bucket fill size. This file attribute directs RMS-11 to load records into buckets only to the particular level -- number of bytes -- specified, leaving free space within each bucket. If a large number of random insertions are to be made to the file, filling buckets only partially can improve processing time.

Buckets are an RMS-11 concept; thus, when RMS-11 requests an I/O operation for a file, it directs the file control processor to move a bucket by specifying the VBN for the first block in the bucket and the size of the bucket in bytes.

## Operations File Processing

### Access Sharing

The degree to which other users can access files is determined by the protection code and by access sharing.

The protection code is a file attribute (see Table 2) that describes the types of access granted to different users or potential users of the file. This attribute is defined for the file at the time the file is created, and is checked by the file control processor to determine whether the user of an accessing program has the right to process or delete the file. The protection code can also be changed when the file is closed.

In many cases, more than one program must access the same file at the same time. RMS-11, therefore, permits access sharing, which is controlled by the accessing programs at the time a file is opened for processing.

Each program to open a file supplies two items of information:

1. The types of operations it will perform on records
2. The types of record operations it will allow other accessing programs to perform

The file control processor will check the file's protection code to ensure that the intended operations are allowed. If so, the file control processor will then check that the operations requested and the operations allowed are compatible with the operations being performed and with the operations allowed by all current accessing programs (if any) of the file. To share access to the file, accessing programs must not only meet the protection criteria but must also specify compatible access sharing.

To protect data in a file in which access sharing is allowed, RMS-11 provides bucket locking. When a set of programs that are concurrently accessing a relative or indexed file allows access sharing for write operations, each bucket moved from disk is locked against access by other programs until RMS-11 unlocks it, usually after completion of the operation, or at the start of the next operation on the stream.

Access sharing also depends on file organization. All files can be shared for reading by any number of programs, but sequential files cannot be shared for writing. Relative and indexed files can be shared for writing by multiple programs.

### RECORD PROCESSING

Once file access is established, the RMS-11 stream and record operations are the interface between your program and the records your program requires.

An access stream is a path between your program and the records in a file (or blocks in a file; see the Block Processing section).

The stream operations are:

- CONNECT -- initiates an access stream
- DISCONNECT -- terminates a stream

## Operations Record Processing

- FLUSH -- writes the contents of I/O buffers (modified records) to the file
- FREE -- releases control of the record or block most recently accessed by the stream
- REWIND -- resets stream context to the beginning of the file (to the first record in the file)
- WAIT -- suspends processing until an outstanding asynchronous operation is completed

The record operations process records within files. These operations are of three types: locate, read, and write. Locate and read operations do not affect the data contents of the records; for example, a FIND operation merely returns the position of a record in a file. Write operations, on the other hand, may result in altered data; for example, an UPDATE operation replaces the contents of an existing record with new data.

The record operations are:

- FIND -- reads a record from a file to an I/O buffer and sets the current-record stream context to that record
- GET -- reads a record from a file to an I/O buffer and then to a user buffer, and sets the current-record stream context to that record
- PUT -- writes a record from a user buffer to an I/O buffer and then to a file
- UPDATE -- transfers a modified record from a user buffer to an I/O buffer and then to a file, overwriting the previous version of the record in the file
- DELETE -- removes an existing record from a relative or indexed file
- TRUNCATE -- effectively deletes all records in a sequential file from the current record through the logical end of the file

### Record Context

Each record operation processes one record at a time along the path established by the access stream. RMS-11 keeps track of the stream's position in the file so that it knows which record is the target of the operation. The stream's position is called its context, which is either the current record or the next record, depending on the operation. The stream's position changes at the completion of the operation.

In general, the current record is the record that was the target of the just-completed operation (the record operated upon), and the next record is the record that follows the current record.

For example, when you use a CONNECT stream operation to establish an access stream to a file opened by means of the OPEN or CREATE file operation, the current-record context is undefined (no record operation has been performed), and the next-record context is the first record in the file.



## Operations Record Processing

If the file is a sequential file and you use a sequential-access FIND record operation, the target will be the next record, which was established by the CONNECT operation as the first record in the file. After the FIND operation is completed, the current record will be the found record (the first record in the file), and the next record will be the record after the found record (the second record in the file).

If you then use a GET record operation, the target will be the current record, which was established by the FIND operation as the first record in the file. After the GET operation is completed, the current record will be the retrieved record (the first record in the file), and the next record will be the record after the retrieved record (the second record in the file).

To modify the record, you can use the UPDATE record operation. Its target will be the current record, which was established by the GET operation as the first record in the file. After the UPDATE operation is completed, the current record will be undefined (it was overwritten), and the next record will be the record that follows the overwritten record.

After a DISCONNECT stream operation, there is no current-record or next-record context.

Although only one record can be processed at a time by means of a stream, more than one stream can be connected to a relative or indexed file. For example, you can open an indexed file and, with one stream, use the primary index to access records randomly and, with a second stream, use an alternate index to access records sequentially.

### Record Access Modes

For the FIND, GET, and PUT record operations, your program specifies record access mode, which determines which record is the target of the operation.

A sequential-access FIND operation can be used to position to records from a file of any organization. The target of a sequential-access FIND operation is the next record, as established by a previous CONNECT or REWIND stream operation or a FIND or GET record operation.

Repeated sequential-access FIND operations may be time-consuming for files with a large number of records. For relative and indexed files and sequential files with fixed-length records, you can use a key-access FIND operation, the target of which is the record whose RRN or key value matches the one that you specify for the operation.

If you know the RFA for a disk file of any organization, you can also use an RFA-access FIND operation, the target of which is the record whose RFA matches the one that you specify for the operation.

Exactly the same dependencies exist for GET operations, which are the same as FIND operations except that records are moved to the I/O buffer and then on to the user buffer. In addition, the target of a sequential-access GET operation depends on whether it was immediately preceded by a FIND operation. If so, the target is the current record; otherwise, the target is the next record.

## Operations Record Processing

A sequential-access PUT operation can be used to write records to a file of any organization. The target of a sequential-access PUT operation depends on the file organization: for sequential files, the target is the end of the file and for relative files, the target is the next record. For indexed files, RMS-11 inserts the record in primary key order and updates the indexes.

The target of a key-access PUT operation for a relative file or for a sequential file with fixed-length records is the cell specified by the RRN. For an indexed file, there is no target; RMS-11 inserts the record and updates the indexes.

Note that you cannot specify RFA access for PUT operations. For more information, see the Access Modes section in Chapter 3.

### Record Transfer Modes

Your program can use either of two record transfer modes to perform record operations: move mode or locate mode.

In both modes, a user buffer, established by your program, is required in addition to the I/O buffer. For read operations in move mode, RMS-11 first transfers the data into the I/O buffer, then copies the record into the user buffer for processing by the program.

For write operations in move mode, your program builds or modifies the record in the user buffer. Then, upon a write request, RMS-11 copies the record into the I/O buffer before writing it to the file.

In locate mode, on the other hand, your program can, under some circumstances, access the record in the I/O buffer directly. This may reduce the amount of data movement, thereby reducing processing time.

### BLOCK PROCESSING

Your program can bypass RMS-11 file and record structures and use block access to process 512-byte blocks one after the other. Your program identifies the starting VBN of the file and the number of bytes to be accessed.

When you use block access, RMS-11 regards the file as a series of blocks, rather than as a series of records. RMS-11 keeps track of the stream's position, or block context, which is either readable block or writable block, depending on the operation.

Block access requires minimum time and space for processing; however, your program must be able to interpret the contents of the blocks.

The block access operations are:

- READ -- gets blocks from a file
- WRITE -- puts blocks in a file

## Operations Block Processing

### Block Access Modes

For the READ and WRITE operations, your program specifies block access mode -- sequential or random by VBN -- which determines which block is the target of the operation. Upon completion of an operation, RMS-11 resets the readable-block and writable-block contexts.

### MACROS AND SYMBOLS

The RMS-11 operation routines are fully accessible only from a MACRO-11 program. High-level languages restrict your options for some operations; see your particular language documentation for information on its support of RMS-11.

The interface between a MACRO-11 program and the RMS-11 operation routines is defined by the RMS-11 macros and symbols. Your program uses the RMS-11 macros to call the RMS-11 operation routines.

Information is passed between the calling program and an RMS-11 operation routine by means of control blocks, which consist of structured series of data fields that contain the information. The information passed includes the attributes that your program supplies to RMS-11 for file and record access and the returned values that RMS-11 supplies to your program as a result of an operation. RMS-11 macros also allow your program to declare and manipulate the control blocks.

The fields of a control block are referenced by symbols that name the locations of fields within control blocks (field-offset symbols) and the codes and bit masks used within the fields (code and mask symbols).

The RMS-11 control blocks are:

- Area allocation (ALL) block -- contains information about a file area
- File date (DAT) block -- contains file creation and revision dates
- File access block (FAB) -- contains information about the file, including device, file attributes, access sharing, record format, record blocking, record-output handling
- File key (KEY) block -- contains index and key information for an indexed file
- File name (NAM) block -- contains file specification information
- File protection (PRO) block -- contains file owner and protection information
- Record access block (RAB) -- contains information about a stream or record and record access mode
- File summary (SUM) block -- contains file and area information for an indexed file

## Operations Macros and Symbols

In addition to calling RMS-11 operations and declaring and manipulating control blocks, RMS-11 macros allow your program to:

- Declare and manipulate memory space (pools)
- Declare RMS-11 facilities that are required for certain operations based on file organization
- Extract from a macro library definitions for RMS-11 macros and symbols

## INDEX

- Access, 1-1
  - block, 4-6
  - direct, 2-4
  - modes, 3-11
  - random, 2-4
  - sequential, 2-4
  - sharing, 2-4, 4-3
    - read, 3-5 to 3-6
    - write, 3-6
  - stream, 4-3
- Access modes, 3-1, 3-11
- and file organization, 3-13
- block, 3-11 to 3-13, 4-7
  - sequential, 3-12
  - VBN, 3-12
- random access to
  - indexed files, 3-15
  - relative files, 3-15
  - sequential files, 3-14
- record, 3-11, 4-5
  - FIND operations, 4-5
  - GET operations, 4-5
  - key, 3-12
  - PUT operations, 4-6
  - RFA, 3-11
  - sequential, 3-11
- sequential access to
  - indexed files, 3-14
  - relative files, 3-14
  - sequential files, 3-13
- switching, 3-13
- Access sharing, 2-4, 4-3
  - read, 3-5 to 3-6
  - write, 3-6
- ALL control block, 4-7
- Alternate keys, 3-7
- Attributes, 3-16, 4-3
- Bit, 2-1
- Block, 2-3
  - access, 4-6
  - context, 4-6
    - readable block, 4-6
    - writable block, 4-6
  - logical, 2-3
  - operations, 4-6
  - processing, 4-1, 4-6
  - spanning, 4-2
  - virtual, 2-3
- Block access, 3-11
- Block access modes, 3-12 to 3-13
  - sequential, 3-12
  - VBN, 3-12
- Bucket, 4-2
  - locking, 4-3
- Bucket fill size, 4-2
- Byte, 2-1
- Cell, 3-5
- Changeable keys, 3-10
- CLOSE operation, 4-2
- CONNECT operation, 4-3
- Context, 4-4
  - block, 4-6
    - readable block, 4-6
    - writable block, 4-6
  - current record, 4-4
  - next record, 4-4
- Contiguity
  - virtual, 2-3
- Control blocks, 4-7
  - ALL, 4-7
  - DAT, 4-7
  - FAB, 4-7
  - KEY, 4-7
  - NAM, 4-7
  - PRO, 4-7
  - RAB, 4-7
  - SUM, 4-7
- CREATE operation, 4-2
- Current record
  - see Context
- Cylinder, 2-1
- DAT control block, 4-7
- Data storage, 2-1
- Data type
  - key values, 3-7
- DELETE operation, 4-4
- Designing files, 3-16
- Device, 2-1
  - disk, 2-1
    - driver, 2-1
    - unit-record, 3-5
- Direct access
  - see Random access
- Directory
  - operations, 4-1
- DISCONNECT operation, 4-3
- Disk
  - drive, 2-1
  - structure, 2-1
    - cylinder, 2-1
    - platter, 2-1
    - sector, 2-1
    - track, 2-1
- DISPLAY operation, 4-2
- Drive, 2-1
- Duplicate keys, 3-10
- ENTER operation, 4-1
- ERASE operation, 4-2

## INDEX

- EXTEND operation, 4-2
- FAB control block, 4-7
- File, 1-1
  - design, 3-16
  - non-RMS-11, 3-4
  - operations, 4-2
  - processing, 4-1
  - specification, 4-1
  - structure, 3-1
- File control processor, 2-1, 4-3
- File organization, 3-1, 3-4
  - indexed, 3-6
    - alternate keys, 3-7
    - changeable keys, 3-10
    - duplicate keys, 3-10
    - index, 3-7
    - keys, 3-6
    - null keys, 3-10
    - primary key, 3-7
    - segmented keys, 3-10
  - random access to
    - indexed files, 3-15
    - relative files, 3-15
    - sequential files, 3-14
  - relative, 3-5
    - cell, 3-5
    - RRN, 3-5
  - sequential, 3-4
    - sequential access to
      - indexed files, 3-14
      - relative files, 3-14
      - sequential files, 3-13
- File specification, 4-1
  - wildcard characters, 4-1
- Fill size
  - see Bucket fill size
- FIND operation, 4-4
  - access modes, 4-5
- Fixed-length record format, 3-1
- FLUSH operation, 4-4
- Format
  - see Record format
- FREE operation, 4-4
- Generic match, 3-12
- GET operation, 4-4
  - access modes, 4-5
- I/O, 2-1
  - buffers, 4-2
- Index, 3-7
- Indexed file organization, 3-6
  - alternate keys, 3-7
  - changeable keys, 3-10
  - data records, 3-7
  - duplicate keys, 3-10
  - index, 3-7
  - index records, 3-7
  - keys, 3-6
    - data type, 3-7
    - null keys, 3-10
    - primary key, 3-7
    - segmented keys, 3-10
    - tree structure, 3-7
- Input/output
  - see I/O
- KEY control block, 4-7
- Key record access mode, 3-12
- Keys, 3-6
  - alternate, 3-7
  - changeable, 3-10
  - data type, 3-7
  - duplicate, 3-10
  - match criteria, 3-12
  - null, 3-10
  - segmented, 3-10
- LBN, 2-4
- Locate mode, 4-6
- Locking buckets, 4-3
- Logical block, 2-3
- Logical block number
  - see LBN
- Macros, 4-7
- Mass-storage device, 2-1
- Match criteria, 3-12
- Mode
  - see Access modes
  - see Record transfer modes
- Move mode, 4-6
- NAM control block, 4-7
- Next record
  - see Context
- Non-RMS-11 files, 3-4
- Null keys, 3-10
- OPEN operation, 4-2
- Operations, 4-1
  - block, 4-6
  - CLOSE, 4-2
  - CONNECT, 4-3
  - CREATE, 4-2
  - DELETE, 4-4
  - directory, 4-1
  - DISCONNECT, 4-3
  - DISPLAY, 4-2
  - ENTEND, 4-2
  - ENTER, 4-1
  - ERASE, 4-2
  - file, 4-2
  - FIND, 4-4
    - access modes, 4-5
  - FLUSH, 4-4
  - FREE, 4-4
  - from high-level languages, 4-7
  - GET, 4-4

## INDEX

- access modes, 4-5
- OPEN, 4-2
- PARSE, 4-1
- PUT, 4-4
  - access modes, 4-6
- READ, 4-6
  - access modes, 4-7
  - record, 4-4
- REMOVE, 4-1
- RENAME, 4-1
- REWIND, 4-4
- SEARCH, 4-1
- SPACE, 4-6
  - access modes, 4-7
  - stream, 4-3
- TRUNCATE, 4-4
- UPDATE, 4-4
- WAIT, 4-4
- WRITE, 4-6
  - access modes, 4-7
- Organization
  - see File organization
- PARSE operation, 4-1
- Platter, 2-1
- Primary key, 3-7
- PRO control block, 4-7
- Processing blocks, 4-1, 4-6
- Processing files, 4-1
- Processing records, 4-1, 4-3
- Prologue, 3-16
- PUT operation, 4-4
  - access modes, 4-6
- RAB control block, 4-7
- Random access, 2-4
  - key, 3-12
    - match criteria, 3-12
  - RFA, 3-11
  - VBN, 3-12
- READ operation, 4-6
  - access modes, 4-7
- Record, 1-1
  - data, 3-7
  - index, 3-7
  - operations, 4-4
  - processing, 4-1, 4-3
  - see also Record format
- Record access modes, 3-11
  - key, 3-12
  - RFA, 3-11
  - sequential, 3-11
- Record file address
  - see RFA
- Record format, 3-1
  - fixed-length, 3-1
  - stream, 3-4
  - undefined, 3-4
  - variable-length, 3-2
  - VFC, 3-2
- Record transfer modes, 4-6
  - locate mode, 4-6
  - move mode, 4-6
- Relative file organization, 3-5
  - cell, 3-5
  - RRN, 3-5
- Relative record number
  - see RRN
- REMOVE operation, 4-1
- RENAME operation, 4-1
- REWIND operation, 4-4
- RFA, 3-11, 3-14 to 3-15
- RRN, 3-5, 3-12, 3-14 to 3-15
- SEARCH operation, 4-1
- Sector, 2-1
- Segmented keys, 3-10
- Sequential access, 2-4
  - block, 3-12
  - record, 3-11
- Sequential file organization,
  - 3-4
  - RRN, 3-5
- SPACE operation, 4-6
  - access modes, 4-7
- Spanning blocks, 4-2
- Storage
  - data, 2-1
- Stream
  - access, 4-3
  - operations, 4-3
- Stream record format, 3-4
  - terminators, 3-4
- SUM control block, 4-7
- Switching access modes, 3-13
- Symbols, 4-7
  - code and mask, 4-7
  - field-offset, 4-7
- Terminators
  - stream records, 3-4
- Track, 2-1
- Tree structure
  - indexed file organization, 3-7
- TRUNCATE operation, 4-4
- Undefined record format, 3-4
- Unit-record devices, 3-5
- UPDATE operation, 4-4
- Variable with fixed control
  - see VFC
- Variable-length record format,
  - 3-2
- VBN, 2-4
  - access, 3-12, 4-6
- VFC record format, 3-2
- Virtual block, 2-3
- Virtual block number
  - see VBN

## INDEX

WAIT operation, 4-4  
Wildcard characters, 4-1  
Word, 2-1  
WRITE operation, 4-6  
    access modes, 4-7



READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

---

---

---

---

---

---

---

---

---

---

Did you find errors in this manual? If so, specify the error and the page number.

---

---

---

---

---

---

---

---

---

---

Please indicate the type of user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) \_\_\_\_\_

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_  
or Country

Do Not Tear - Fold Here and Tape

**digital**



No Postage  
Necessary  
if Mailed in the  
United States

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS ZK1-3/J35  
DIGITAL EQUIPMENT CORPORATION  
110 SPIT BROOK ROAD  
NASHUA, NEW HAMPSHIRE 03061



Do Not Tear - Fold Here

Cut Along Dotted Line

digital



\*000644\*



SHREWSBURY LIBRARY  
DIGITAL EQUIPMENT CORPORATION  
SHR1 3/G18  
DTN 237-3400

DECLIT  
AA  
PRO  
P193B

software **digital**

DECLIT AA PRO P193B  
IAS/RSX-11 ODT reference  
manual supplement

**Digital Equipment Corporation**  
**SHREWSBURY LIBRARY**

94-0031049/30



# IAS/RSX-11 ODT Reference Manual Supplement

Order No. AA-P193B-TK

May 1983

This supplement to the *IAS/RSX-11 ODT Reference Manual* identifies specific sections of the manual that do not apply when using ODT in the context of the Professional Developer's Tool Kit.

REVISION/UPDATE INFORMATION: Documentation Supplement

SOFTWARE VERSION: Professional Developer's Tool Kit V1.5

Digital Equipment Corporation  
SHREWSBURY LIBRARY

DECLIT  
AA  
PRO  
P193B

27976

DIGITAL EQUIPMENT CORPORATION  
Maynard, Massachusetts 01754

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.


The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The specifications and drawings, herein, are the property of Digital Equipment Corporation and shall not be reproduced or copied or used in whole or in part as the basis for the manufacture or sale of items without written permission.

Copyright © 1982, 1983 by Digital Equipment Corporation  
All Rights Reserved

The following are trademarks of Digital Equipment Corporation:

CTI BUS	MASSBUS	Rainbow
DEC	PDP	RSTS
DECmate	P/OS	RSX
DECsystem-10	PRO/BASIC	Tool Kit
DECSYSTEM-20	PRO/Communications	UNIBUS
DECUS	Professional	VAX
DECwriter	PRO/FMS	VMS
DIBOL	PRO/RMS	VT
	PROSE	Work Processor
	PROSE PLUS	

## CONTENTS

---

	<b>Page</b>
1.0	I/D Space and Supervisor Mode ..... 1
2.0	Interrupting and Resuming a Debugging Session Using CTRL/C ..... 1
3.0	Commands Not Supported ..... 1
4.0	Setting Breakpoints that Refer to I/D Space ..... 2
5.0	Redirecting ODT Output ..... 2
6.0	PDS Command Line ..... 3
7.0	The Trace Debugging Program ..... 3





## IAS/RSX-11 ODT Reference Manual Supplement

---

This manual identifies those areas of the *IAS/RSX-11 ODT Reference Manual* that do not apply in the P/OS operating environment.

Although P/OS is based on RSX-11M-PLUS, it does not include all of the RSX system features. The supplement describes specific sections of the *IAS/RSX-11 ODT Reference Manual* that should be ignored in the P/OS environment and identifies the specific RSX system feature not supported.

### **1.0 I/D Space and Supervisor Mode**

P/OS does not include RSX support for instruction and data space (I/D space) capabilities. In addition, P/OS does not include supervisor mode registers and, therefore, does not include supervisor-mode libraries.

The following sections in the *IAS/RSX-11 ODT Reference Manual* do not apply when using ODT in a P/OS environment:

- Section 1.2.4 Linking to Enable RSX-11M-PLUS Features
- Section 1.2.4.1 Enabling Instruction and Data Space
- Section 1.2.4.2 Enabling Supervisor-Mode Library Debugging

### **2.0 Interrupting and Resuming a Debugging Session Using CTRL/C**

P/OS does not support the CTRL/C sequence in the terminal driver. The Professional keyboard replaces this function with the INTERRUPT/DO function keys. The ability to interrupt or resume a debugging session using CTRL/C is not pos-

sible in the P/OS environment. Therefore, the following sections in the *IAS/RSX-11 ODT Reference Manual* do not apply to P/OS:

- Section 1.5 Interrupting a Debugging Session
- Section 1.5.1 Resuming a Debugging Session (RSX-11M-PLUS)
- Section 1.5.2 Resuming a Debugging Session (IAS)—(P/OS is also not an IAS system.)

### **3.0 Commands Not Supported**

Because P/OS does not include support for I/D space, the following commands listed in Table 2-3 do not apply in the P/OS environment:

- The "I" command (Access Data Space)
- The "I" command (Access Instruction Space)
- The "U" command (sets the current mode of ODT to user mode)
- The "Z" command (sets the current mode of ODT to supervisor mode)

### **4.0 Setting Breakpoints That Refer to I/D Space**

Since P/OS does not support I/D space, breakpoints never refer to I/D space. Therefore, the last two (shaded) paragraphs in Section 3.1.1 do not apply.

### **5.0 Redirecting ODT Output**

*Note:* The information on LUN assignments in Section 6.1.4 of the *IAS/RSX-11 ODT Reference Manual* does not apply directly to the P/OS environment.

This section describes how to direct ODT output to the printer port on the Professional, so you can use a debugging terminal connected to that port. You can use any of the VT100 or LA series terminals running at 9600 baud for that purpose. Note that when connecting a debugging terminal to the printer port, you should use a console cable, not a printer cable.

The device control LUN registers \$0D and \$1D contain the logical unit numbers of the ODT output devices. ODT assigns both LUNs to the Professional video terminal (video monitor/keyboard, device TT1:). This means all output goes to the video display by default.

To debug with a debugging terminal instead of the Professional video terminal, you must change the value in the \$0D register to reflect a LUN assignment of the printer port (device TT2:).

To make a LUN assignment to the printer port,

1. At task build time, include the following command line in the command file:

ASG = TT2:n

where n is a decimal number.

This ASG option assigns the LUN (n) to the printer port.

- 2 After you start your program, open register \$0D and deposit the LUN number (n) of the printer port. Use the following ODT command to do this:

\$0D/n

where n is the octal equivalent of the decimal number used in the ASG statement above.

ODT will then prompt at the debugging terminal.

## **6.0 PDS Command Line**

Since P/OS is not an IAS system, the PDS command line is invalid for P/OS. Therefore, Section 8.1.3 does not apply.

## **7.0 The Trace Debugging Program**

P/OS does not support Trace. Therefore, Sections 8.2 through 8.2.2 do not apply in the P/OS environment.





--- Do Not Tear - Fold Here and Tape ---

**digital**

No Postage  
Necessary  
if mailed in the  
United States

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Professional 300 Series Publications  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET  
MAYNARD, MASSACHUSETTS 01754

--- Do Not Tear - Fold Here ---

Cut Along Dotted Line

digital



\*0006376\*