

Tool Kit PRO/DATATRIEVE Documentation Supplement

AA-CK37A-TH

September, 1984

This manual provides information on using components of PRO/DATATRIEVE to develop PRO/DATATRIEVE applications for the Professional hard disk computer system.

OPERATING SYSTEM AND VERSION:	P/OS	V2.0 or later
	VMS	V3.2 or later
	RSX-11M	V4.1 or later
	RSX-11M-PLUS	V2.1 or later

SOFTWARE VERSION:	Professional Host Tool Kit	V2.0
	PRO/Tool Kit	V2.0
	DATATRIEVE-11	V3.0
	PRO/DATATRIEVE	V2.0

digital equipment corporation, maynard, massachusetts

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 © 1984 by Digital Equipment Corporation. All rights reserved.

The following are trademarks of Digital Equipment Corporation:

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

digital™

MASSBUS
PDP
P/OS
PRO/BASIC
PRO/Communications
Professional
PRO/FMS
PRO/RDT
PRO/RMS
PROSE

PROSE PLUS
Rainbow
RSTS
RSX
Tool Kit
UNIBUS
VAX
VMS
VT
Work Processor

Contents

	Page
How to Use This Manual	
Intended Audience	v
Structure	v
Related Manuals	vi
Conventions	vi
1 Components and Features of Tool Kit PRO/DATATRIEVE	
Interactive PRO/DATATRIEVE	1-2
PRO/DDMF	1-2
The PRO/DATATRIEVE Call Interfaces	1-3
PRO/DATATRIEVE Features	1-5
PRO/DATATRIEVE Logical Name Capability	1-6
PRO/DATATRIEVE Function Keys	1-6
The PRO/DATATRIEVE Call Interfaces	1-7
2 Developing PRO/DATATRIEVE Applications	
Creating Application Files	2-2
Using Logical Names	2-3
Writing the Application Installation File	2-5
Compressing the Application Dictionary	2-6
3 Using the PRO/DATATRIEVE Call Interfaces	
Writing the Source Program	3-1
Compiling the Source Program	3-2
Task Building the Application	3-2
Editing the Task Builder Command File	3-2
Editing the Overlay Descriptor (.ODL) File	3-4
Creating the Application Installation File	3-5
Running and Debugging the Application	3-6
Creating Distribution Kits	3-6
A Documentation Directory	
PRO/DATATRIEVE Documentation	A-1
PRO/RDT Documentation	A-2
Tool Kit PRO/DATATRIEVE Documentation	A-3
B PRO/DATATRIEVE SAMPLE APPLICATION	
C Sample PASCAL DAB	
Figures	
1-1 PRO/DATATRIEVE Remote Call Interface	1-4
1-2 PRO/DATATRIEVE Local Call Interface	1-5
3-1 LUNMAP Bit Settings	3-4

How to Use This Manual

This manual explains how to use Tool Kit PRO/DATATRIEVE components to create PRO/DATATRIEVE applications for the Professional hard disk computer system. It is one of seven manuals included in the Tool Kit PRO/DATATRIEVE documentation set.

Intended Audience

This manual assumes that:

- You have experience with PRO/DATATRIEVE or DATATRIEVE-11
- You have experience using the PRO/Tool Kit or the Professional Host Tool Kit
- You are familiar with your Professional P/OS operating system and user interface

If you intend to use the callable components of PRO/DATATRIEVE, you should also be familiar with a Tool Kit programming language such as BASIC-PLUS-2, PASCAL, COBOL, or FORTRAN.

Structure

This document contains three chapters, three appendixes, an index, and a master index for all manuals in the Tool Kit PRO/DATATRIEVE documentation set:

Chapter 1	Describes the components of Tool Kit PRO/DATATRIEVE and differences between Tool Kit PRO/DATATRIEVE and DATATRIEVE-11.
Chapter 2	Explains the Tool Kit PRO/DATATRIEVE development process.
Chapter 3	Explains how to use the callable components of Tool Kit PRO/DATATRIEVE.
Appendix A	Contains a documentation directory.
Appendix B	Contains a sample PRO/DATATRIEVE application.
Appendix C	Contains a sample definition of the DATATRIEVE Access Block in PASCAL.

Related Manuals

PRO/DATATRIEVE is derived from DATATRIEVE-11, so the following manuals are included in the Tool Kit PRO/DATATRIEVE documentation set to supply usage and reference information:

- *Introduction to DATATRIEVE-11*
- *DATATRIEVE-11 User's Guide*
- *DATATRIEVE-11 Guide to Writing Reports*
- *DATATRIEVE-11 Call Interface Manual*
- *DATATRIEVE-11 Reference Manual*
- *DATATRIEVE-11 Pocket Guide*

In the course of developing PRO/DATATRIEVE applications, you may need to refer to other documentation:

- For information about the implementation of function keys in PRO/DATATRIEVE, refer to Chapter 2 of *PRO/DATATRIEVE for Beginners*.
- For information about the Professional and Tool Kit environments, refer to Tool Kit documentation. Your Tool Kit user's guide is the primary resource for all application development activities and is an excellent starting point.
- For information about programming languages, refer to the language-specific documentation.

Conventions

- Programming examples appear in this manual in a dot matrix typeface. PRO/DATATRIEVE or program output lines displayed on your Professional are printed in black. Commands and statements you type on your keyboard are printed in red.
- Brackets ([]) are required syntax in directory specifications.
- The term "Professional" refers to the Professional 300 series computer equipped with a hard disk.
- The term "PRO/Tool Kit" refers to the Professional Developer's Tool Kit running on a Professional.
- The term "Host Tool Kit" refers to the Professional Developer's Tool Kit running on a VMS, RSX-11M, or RSX-11M-PLUS system.
- The term "Tool Kit" refers to both the PRO/Tool Kit and the Host tool Kit.
- Language names like "BASIC-PLUS-2," "COBOL," and "FORTRAN" refer to both the native and host versions of these Tool Kit languages.

Components and Features of Tool Kit PRO/DATATRIEVE 1

Tool Kit PRO/DATATRIEVE is a documentation-only option that explains how to:

- Build end-user applications that use interactive PRO/DATATRIEVE
- Use the PRO/DATATRIEVE V2.0 Remote Call Interface, Local Call Interface, and callable PRO/DATATRIEVE to create distributed data management applications

The components of Tool Kit PRO/DATATRIEVE applications are:

- Interactive PRO/DATATRIEVE

The task image (PRODTR.TSK) that you access from your Professional when you select PRO/DATATRIEVE from a menu. You can customize interactive PRO/DATATRIEVE by using the PRO/DATATRIEVE logical name capability and building an application end users can install on a menu.

- Callable PRO/DATATRIEVE

The callable PRO/DATATRIEVE task image (PROCALDTR.TSK) that the Local Call Interface accesses. Interactive PRO/DATATRIEVE and callable PRO/DATATRIEVE are both part of PRO/DATATRIEVE Version 2.0.

- The PRO/DATATRIEVE Local and Remote Call Interfaces

Routines in the PRO/DATATRIEVE Call Interface Object Module Library, DTCLIB.OLB, that let programs written in Tool Kit languages call PRO/DATATRIEVE, DATATRIEVE-11, VAX DATATRIEVE, or DATATRIEVE-20. DTCLIB.OLB is distributed with the PRO/Tool Kit Version 2.0 and the Host Tool Kit Version 2.0.

- PRO/RDT (Remote Data Transfer)

An information management tool that works with PRO/DATATRIEVE Version 2.0 to let you extract selected data from DATATRIEVE domains on other computers. PRO/RDT menus prompt you for the information needed to access and retrieve the DATATRIEVE data you want.

- **PRO/DDMF** (the Distributed Data Manipulation Facility)

The PRO/DATATRIEVE Distributed Server that lets users on other DECnet nodes access PRO/DATATRIEVE data files and dictionaries on your Professional. PRO/DDMF substitutes an interface to DECnet for the interactive PRO/DATATRIEVE interface. PRO/DDMF is distributed with PRO/RDT Version 1.0. See the *PRO/RDT Handbook* for more information on PRO/RDT and PRO/DDMF.

By using these components in different combinations, you can perform many different data management activities:

- Using interactive PRO/DATATRIEVE, you can perform ad hoc data management functions such as organizing and reporting on data, or you can develop customized applications as described in Chapter 2 of this manual.
- Using PRO/RDT and PRO/DATATRIEVE, you can extract DATATRIEVE information from a remote host and manipulate it on your Professional to create the reports you need for your daily activities.
- Using PRO/DECnet, the Remote Call Interface, and a remote DDMF, your program can call PRO/DATATRIEVE on another Professional, DATATRIEVE-11 on a PDP-11 system, DATATRIEVE-20 on a DECSYSTEM-20, or VAX DATATRIEVE on a VMS system.
- Using the Local Call Interface and callable PRO/DATATRIEVE, your program can call PRO/DATATRIEVE on your own Professional.

Interactive PRO/DATATRIEVE

When you select PRO/DATATRIEVE from a P/OS menu, you are running PRODTR.TSK, the interactive PRO/DATATRIEVE task image. This program accepts PRO/DATATRIEVE commands as you type them and uses your Professional video screen as the default output device.

By using PRO/DATATRIEVE's logical name capability to identify application files and devices, you can customize the PRO/DATATRIEVE interface for end-user applications and integrate your application with the PRO/DATATRIEVE task image and message file. End users can then install your layered application and select it from a P/OS menu.

Chapter 2 describes how to create layered PRO/DATATRIEVE applications.

PRO/DDMF

PRO/DDMF, the Distributed Data Manipulation Facility, is distributed with PRO/RDT Version 1.0 (Remote Data Transfer). It is a server program, DDMF.TSK, that substitutes for PRO/DATATRIEVE when a program on a remote node calls PRO/DATATRIEVE. PRO/DDMF executes commands passed by the calling program and passes results back to the calling program.

Note

If your program calls PRO/DATATRIEVE on a remote Professional, that Professional must have PRO/DECnet software and PRO/RDT installed.

The PRO/DATATRIEVE Call Interfaces

The PRO/DATATRIEVE call interfaces let you write Tool Kit language programs that call PRO/DATATRIEVE on your Professional (the Local Call Interface) or PRO/DATATRIEVE, DATATRIEVE-11, VAX DATATRIEVE, or DATATRIEVE-20 on a remote node (the Remote Call Interface).

Note

You must have PRO/DECnet software installed on your Professional before you can use the Remote Call Interface. PRO/DECnet is not required to use the Local Call Interface.

The DTCLIB.OLB Object Module Library shipped with the PRO/Tool Kit and Host Tool Kit contains routines necessary for using the Remote and Local Call Interfaces. To use either call interface, you include calls in your program to subroutines contained in DTCLIB. When you task build the program, you link your program with DTCLIB and specifically with either the Remote Call Interface module, NCPOS, or the Local Call Interface module, LCPOS:

- NCPOS sends commands to and receives information from the DDMF task on the remote node.
- LCPOS sends commands to and receives information from the callable PRO/DATATRIEVE task on your Professional.

Chapter 3 in this manual describes how to link your program with DTCLIB routines.

When you run a program that calls DATATRIEVE on a remote node, the following tasks are active:

- The program linked to DTCLIB.OLB routines
- A DDMF task on the remote node

When you run a program that calls PRO/DATATRIEVE on your own Professional, the following tasks are active:

- The program linked to DTCLIB.OLB routines
- Callable PRO/DATATRIEVE (PROCALDTR.TSK) on your Professional

A DDMF task on a remote node and callable PRO/DATATRIEVE on your Professional serve the same functions but for different call interfaces:

- DDMF services calls to DATATRIEVE from programs running on other DECnet nodes.
- PROCALDTR.TSK services calls to PRO/DATATRIEVE from programs running on the local Professional.

The Remote Call Interface uses PRO/DECnet software to access a remote node. It then uses the DDMF task on the remote node to access PRO/DATATRIEVE data files and dictionaries. Figure 1-1 shows how calling programs access data files and DATATRIEVE files and dictionaries on remote systems.

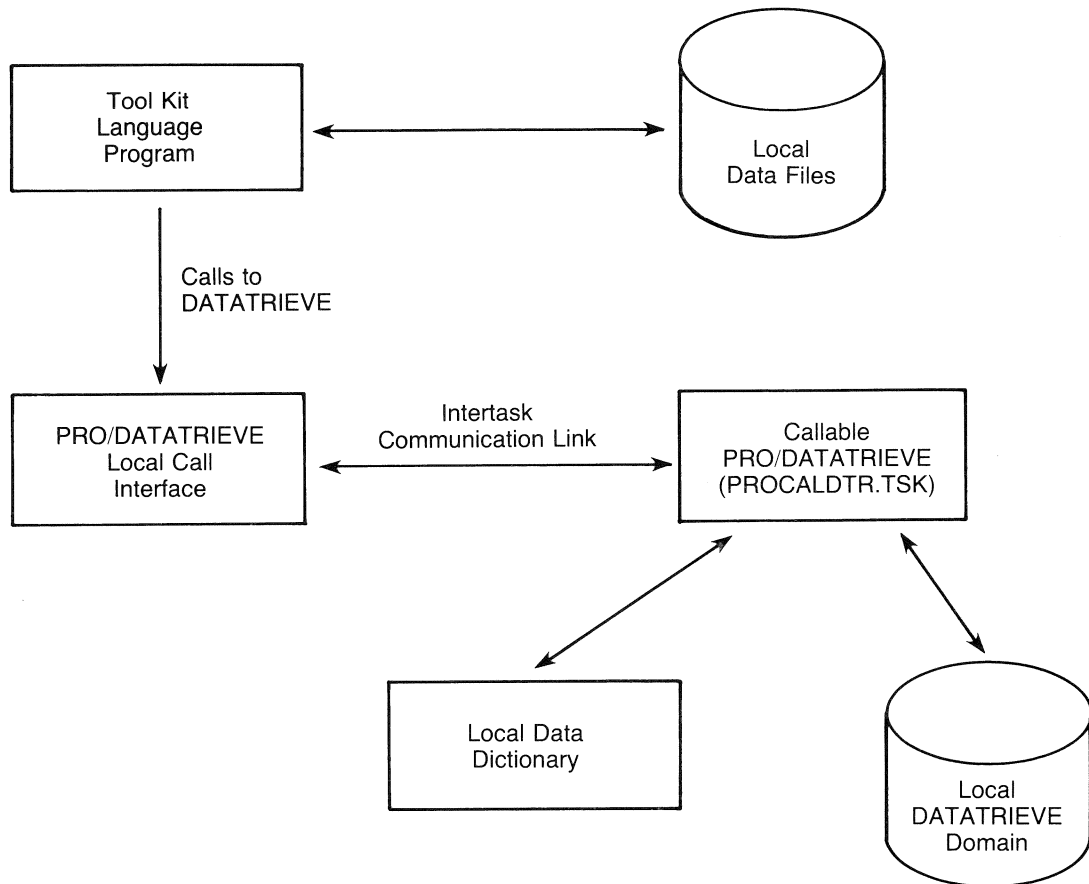


Figure 1-1: PRO/DATATRIEVE Remote Call Interface

The Local Call Interface uses PROCALDTR.TSK to access PRO/DATATRIEVE data files and dictionaries on the local Professional system. Figure 1-2 shows how calling programs access data files and DATATRIEVE files and dictionaries on local systems.

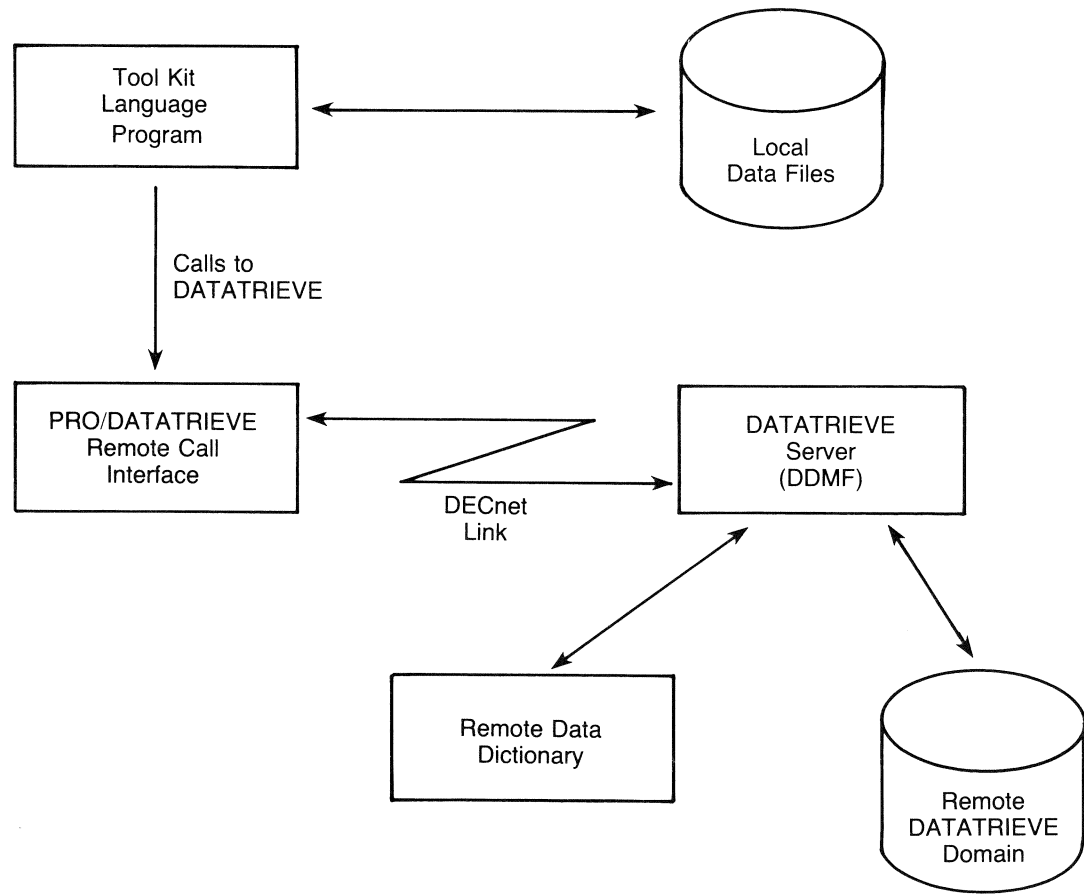


Figure 1-2: PRO/DATATRIEVE Local Call Interface

PRO/DATATRIEVE Features

PRO/DATATRIEVE and DATATRIEVE-11 are essentially the same, but the following differences do exist:

- PRO/DATATRIEVE has an extended logical name capability; DATATRIEVE-11 does not.
- PRO/DATATRIEVE recognizes function keys; DATATRIEVE-11 does not.
- The PRO/DATATRIEVE Remote Call Interface and Local Call Interface are separate and are activated by different Call Interface Object Module Library routines; the DATATRIEVE-11 Remote Call Interface functions as the local call interface through a DECnet loopback link.

Because of language differences, applications developed with VAX DATATRIEVE may require modification to work with PRO/DATATRIEVE.

For example, the default data alignment for VAX DATATRIEVE is MAJOR-MINOR, while the default for PRO/DATATRIEVE and DATATRIEVE-11 is LEFT-RIGHT. Consequently, the record definition and the actual records may not be the same size. To ensure that PRO/DATATRIEVE record definitions map to the actual records, add an ALLOCATION IS MAJOR-MINOR clause to the record definition.

For best results, use PRO/DATATRIEVE or DATATRIEVE to develop PRO/DATATRIEVE applications.

PRO/DATATRIEVE Logical Name Capability

With PRO/DATATRIEVE's logical name capability, you can assign logical names to devices and files in the application installation file. You can also use logical names in PRO/DATATRIEVE statements and commands wherever you can use a file specification or device name. See the section on logical names in Chapter 2 for more information.

PRO/DATATRIEVE Function Keys

You can use the MAIN SCREEN, DO, CANCEL, HELP, and EXIT function keys and the INTERRUPT/DO key sequence when using PRO/DATATRIEVE much as you can when using Professional menus. If you press other function keys on the Professional keyboard while using PRO/DATATRIEVE, you hear a beeping noise. This indicates that the key does not perform a specific function in PRO/DATATRIEVE.

Function keys perform in the following ways:

- **DO** – Executes a PRO/DATATRIEVE command or statement or ends an input line. Using the DO key is identical to using the RETURN key.
- **MAIN SCREEN** – When pressed in response to the DTR> prompt, leaves PRO/DATATRIEVE and returns to the Main Menu. Pressing MAIN SCREEN while PRO/DATATRIEVE is performing a calculation or executing a command or statement returns you to the Main Menu after PRO/DATATRIEVE finishes its current operation. In Guide Mode, pressing the MAIN SCREEN key returns you to the DTR> prompt.
- **CANCEL** – Cancels a command or statement. PRO/DATATRIEVE ignores the command or statement and prompts again for input. Using the CANCEL key is identical to using CTRL/U.
- **HELP** – When pressed in response to the DTR> prompt or after you type HELP topic, displays the introductory HELP frame. To get HELP on a topic, you must use the HELP command and press DO or RETURN. In Guide Mode, the HELP key displays a list of possible responses. In ADT, the HELP key provides information on ADT questions.

- EXIT – The effect of the EXIT key depends on what PRO/DATATRIEVE is doing:
 - Pressing EXIT 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 EXIT in response to any other PRO/DATATRIEVE prompt, including Guide Mode, returns you to the DTR> prompt.
 - Pressing EXIT while PRO/DATATRIEVE is calculating or executing a command or statement returns you to the Main Menu *after* PRO/DATATRIEVE finishes its current operation.

Using the EXIT key is identical to using CTRL/Z.

- INTERRUPT/DO – The effect of an INTERRUPT/DO sequence also depends on what PRO/DATATRIEVE is doing:
 - Pressing INTERRUPT/DO in response to *any* PRO/DATATRIEVE prompt returns you to the Main Menu.
 - Pressing INTERRUPT/DO while PRO/DATATRIEVE is calculating or executing a command returns you to the DTR> prompt.

Using INTERRUPT/DO is identical to using CTRL/C.

See Chapter 2 in *PRO/DATATRIEVE for Beginners* for more information on PRO/DATATRIEVE function keys.

The PRO/DATATRIEVE Call Interfaces

Tool Kit PRO/DATATRIEVE has separate Remote and Local Call Interfaces. The DATATRIEVE-11 Remote Call Interface, as described in the *DATATRIEVE-11 Call Interface Manual*, includes the local call interface.

The modules NC11M and NCRSTS in the DATATRIEVE-11 Call Interface Object Module Library are replaced in the PRO/DATATRIEVE Call Interface Object Module Library by NCPOS and LCPOS. NCPOS activates the Remote Call Interface and LCPOS activates the Local Call Interface.

Developing PRO/DATATRIEVE Applications **2**

This chapter supplements the *DATATRIEVE-11 User's Guide* and is intended for experienced users of PRO/DATATRIEVE or DATATRIEVE-11. For introductory information, see *Introduction to DATATRIEVE-11* or *PRO/DATATRIEVE for Beginners*. For syntax and usage rules for PRO/DATATRIEVE statements and commands, refer to the *DATATRIEVE-11 Reference Manual*.

This chapter describes the development process for creating PRO/DATATRIEVE applications that an end user can install on a Professional and select from a menu. Chapter 3 describes the additional steps involved in developing Tool Kit language applications that use the PRO/DATATRIEVE Remote or Local Call Interface.

Developing a PRO/DATATRIEVE application consists of the following steps:

1. Writing the PRO/DATATRIEVE application files. The next section describes the types of application files a PRO/DATATRIEVE application uses.
2. Using logical names to identify application files and devices. The section entitled "Using Logical Names" describes PRO/DATATRIEVE's logical name capability.
3. Writing the application installation file. This file identifies your application files and installs the PRO/DATATRIEVE task image. It also assigns logical names to your application dictionary and may assign logical names to other application files and devices. See the section in this chapter entitled "Writing the Application Installation File."
4. Installing, executing, and debugging the application on a Professional. Your Tool Kit user's guide explains how to install applications. After you install an application on your Professional, select it from a menu to identify problems. If you find errors in the application, edit the application files and retry the application. Continue this process until the application works the way you want.
5. Compressing the application dictionary. The section entitled "Compressing the Application Dictionary" describes this step.
6. Creating distribution kits with the Application Diskette Builder. This step is described in your Tool Kit user's guide.

Creating Application Files

The minimum PRO/DATATRIEVE application consists of a dictionary file that contains the application definitions. It may also include other dictionary files, command files, data files, and an initialization file that sets dictionary and environment defaults. The PRO/DATATRIEVE V2.0 application, for example, consists of the following files:

PRODTR.TSK	The PRO/DATATRIEVE task image
PROCALDTR.TSK	The callable PRO/DATATRIEVE task image
PRODTR.MSG	The PRO/DATATRIEVE message file
PRODTR.DIC	The default data dictionary
PRODTR.INS	The PRO/DATATRIEVE installation file
EMPLOYEES.DAT	A sample data file

Unless you are developing a Tool Kit language application that calls PRO/DATATRIEVE (as described in the next chapter), your application does not need its own task (.TSK) or message (.MSG) file, because it uses the PRO/DATATRIEVE task image and message file.

See the *DATATRIEVE-11 User's Guide* and *DATATRIEVE-11 Reference Manual* included in this documentation set for additional information on developing applications.

All application file names must conform to the following P/OS format for naming files:

volumename:[directory]filename.typ

Volume Name (volumename):

A volume name can consist of from 1 to 12 alphanumeric characters and should not duplicate any of the following names reserved for P/OS:

BIGDISK	DZ2	TT
BIGVOLUME	LB	TT1
DISKETTE1	LP	TT2
DISKETTE2	SY	XK
DW1	SYSDISK	XT
DZ1	TI	

If you omit a volume name, P/OS uses the current default.

Directory ([directory])

A directory name can consist of from one to nine alphanumeric characters. It must be enclosed in square brackets ([]) or angle brackets (<>). If you omit a directory name, P/OS uses the current default.

File Name (filename)

A file name can consist of from one to nine alphanumeric characters. P/OS converts lowercase letters in file names to uppercase letters. Spaces, punctuation, and other special characters are not valid characters in file names.

File Type (.typ)

You can explicitly specify any three characters you like as a file type. However, if you omit the file type PRO/DATATRIEVE uses the following default file types that describe the internal organization of the file:

Dictionary files	.DIC
Command files	.CMD
Data files	.DAT
Initialization file	.INI

The default for PRO/DATATRIEVE command files created with the EXTRACT command is .CMD. You can override this default when you use the EXTRACT command. The default for data files identified in the DEFINE DOMAIN command and created with the DEFINE FILE command is .DAT. You can override this default by specifying a file type.

See your Tool Kit user's guide for more details on naming P/OS files.

Using Logical Names

Most applications that run on the Professional let you assign logical names only to devices. The PRO/DATATRIEVE logical name facility lets you assign logical names to entire or partial file specifications, with only one restriction: a logical name, if used, must be the first field in the file specification. You cannot, for example, specify a physical device name, like DZ1:, and a logical directory name, like MAIL\$DIR in the same file specification.

A PRO/DATATRIEVE logical name may replace any of the following:

- A device name (BIGVOLUME:)
- A device and directory name (BIGVOLUME:[ZZDTR])
- A device, directory, and file name ((BIGVOLUME:[ZZDTR]PRODTR)
- A full file specification (BIGVOLUME:[ZZDTR]PRODTR.TSK)

You assign logical names in the application installation file with the ASSIGN LOGICAL command, as described in the section entitled "Writing the Application Installation File."

In addition to logical names you assign, the PRO/Dispatcher assigns the logical name APPL\$DIR to the directory on the hard disk where your application files reside. The P/OS Install program creates application directories in the form [ZZAPnnnnn], where an integer value in the form nnnnn is assigned. This integer value varies from one Professional to another, so you can use the logical name APPL\$DIR to identify files that reside in your application's ZZAP directory.

Because application files reside in one directory and the PRO/DATATRIEVE task image and message file reside in another directory ([ZZDTR]), the PRO/DATATRIEVE application uses logical names to locate the files it needs to operate:

- **DTR\$DIC** Identifies a PRO/DATATRIEVE application dictionary file. You assign this logical name to your application's dictionary file.
- **DTR\$INI** Identifies an optional initialization file that contains PRO/DATATRIEVE commands and statements. PRO/DATATRIEVE accesses the file identified by this logical name when it starts, before processing any terminal input.

You can assign logical names to application files and to the Professional video screen in the application installation file in this way:

```
Assign Logical DTR$DIC "APPL$DIR:INVENTORY.DIC"  
Assign Logical DTR$INI "APPL$DIR:INVENTORY.CMD"  
Assign Logical SCREEN "TI:"
```

Notice that the first two lines in the example assign the logical names DTR\$DIC and DTR\$INI in terms of the logical name assigned to the application directory, APPL\$DIR. The first line assigns the logical name DTR\$DIC to an application dictionary, INVENTORY.DIC. This causes PRO/DATATRIEVE to use that dictionary instead of the standard dictionary, PRODTR.DIC. The second line in the example assigns the logical name DTR\$INI to an application command file, INVENTORY.CMD. PRO/DATATRIEVE executes this command file before displaying the first DTR> prompt.

The third line in the example assigns the logical name SCREEN to TI:, the device name of the Professional video screen. Your application could use this logical name in statements like the following:

```
REPORT CURRENT SORTED BY SALESMAN ON *,"File name or SCREEN"
```

Because SCREEN is defined as TI:, PRO/DATATRIEVE displays the report on the video screen when the user types SCREEN in response to the prompting value expression.

In addition to using logical names in the application installation file, you can also use APPL\$DIR when defining application data files:

```
DEFINE DOMAIN WIDGETS USING WIDGETS-REC ON APPL$DIR:WIDGETS.DAT;  
DEFINE DOMAIN SALESMEN USING SALESMEN-REC ON APPL$DIR:SALESMEN.DAT;
```

These domain definitions make it possible for PRO/DATATRIEVE to access the application data files, WIDGETS.DAT and SALESMEN.DAT, from the directory where the end user installed your application. Note, however, that the domains can be accessed only from the application.

Writing the Application Installation File

The application installation file:

- Identifies all your application files
- Identifies the PRO/DATATRIEVE dictionary
- Installs the PRO/DATATRIEVE task image
- Runs PRO/DATATRIEVE

If you plan to use the Fast Install utility to install your application without first building application diskettes, the installation file must have the same name as the directory in which it resides. It must also have a file type of .INS. See your Tool Kit user's guide for more information about Fast Install requirements.

A PRO/DATATRIEVE application installation file needs only the following elements:

- **NAME "menu name"**

You can specify only one NAME command.

- **FILE filename/option**

You must identify each application file your application uses in a FILE command and specify an option (KEEP or DELETE) that determines whether the file is kept or deleted when the application is removed. See your Tool Kit documentation for information on KEEP and DELETE.

Do not include a FILE command for the PRO/DATATRIEVE task image, the PRO/DATATRIEVE message file, or the application installation file.

- **INSTALL [ZZDTR]PRODTR.TSK/TASK**

This line installs the PRO/DATATRIEVE task image. You must include the /TASK value to indicate that the file is an executable task image.

- **ASSIGN LOGICAL logical name "string"**

You must include ASSIGN LOGICAL commands to assign the logical name DTR\$DIC to your application dictionary or the default PRO/DATATRIEVE dictionary, [ZZDTR]PRODTR.DIC. You may include as many more ASSIGN LOGICAL commands as your application needs. See the section on logical names for information on assigning and using logical names.

- **RUN DTR**

You can specify only one RUN command. For applications layered on PRO/DATATRIEVE, RUN DTR tells the PRO/Dispatcher to execute PRODTR.TSK when the user selects the application from a P/OS menu.

All lines must appear in the order shown in the previous list and in the following sample installation file:

```
! This is INVENTORY.INS, the installation
! file for the INVENTORY application
Name "INVENTORY"
File INVENTORY.DIC/DELETE
File INVENTORY.CMD/DELETE
File WIDGETS.DAT/DELETE
File SALESMEN.DAT/DELETE
Install [ZZDTR]PRODTR.TSK/TASK
Assign Logical DTR$DIC "APPL$DIR:INVENTORY.DIC"
Assign Logical DTR$INI "APPL$DIR:INVENTORY.CMD"
Assign Logical SCREEN "TI:"
RUN DTR
```

This application installation file lists four application files and marks all four for deletion upon removal of the application. The INSTALL line installs the PRO/DATATRIEVE task image. The ASSIGN LOGICAL lines assign logical names to the application dictionary, an initialization command file, and the video screen. The RUN DTR line tells the PRO/Dispatcher to execute the PRO/DATATRIEVE task image.

Compressing the Application Dictionary

After you have thoroughly debugged an application, you should compress the application dictionary so that it takes up as little disk space as possible. Compressing a dictionary is particularly important if you have made extensive changes while refining the application.

To compress a PRO/DATATRIEVE dictionary, take the following steps:

1. Select the application or PRO/DATATRIEVE from the Main Menu.
2. Set your current dictionary to the application dictionary.
3. Use the SHOW ALL command to see the names of all definitions stored in the application dictionary.
4. Use the EXTRACT command to write all dictionary definitions to a command file:

```
DTR> SET DICTIONARY INVENTORY
DTR> SHOW ALL
Domains:
      CUSTOMERS      CUST_VIEW      EMPLOYEES      INVENTORY
      SALES_VIEW     SUPPLIERS      TRANSACTIONS   TRANS_BY_ITEM
Records:
      CUSTOMERS_REC  EMPLOYEES_REC  INVENTORY_REC  SUPPLIERS_REC
      TRANS_REC
Procedures:
      CUR_REPORT     CUST_REP       INV_STORE       MAIL_FORMAT
      MAIL_REPORT    TRANS_WRITE    UPDATE_MASTER   WRITE_PRICE
Tables:
      AREA_TABLE     COMPANIES
The current dictionary is DW1:[INVENTORY]INVENTORY.DIC;1
No established collections
No ready domains
```

(continued on next page)

```
DTR> EXTRACT ON NEWDIC.CMD CUSTOMERS, CUST_VIEW, EMPLOYEES,  
CON> INVENTORY, SALES_VIEW, SUPPLIERS, TRANSACTIONS, TRANS_BY_ITEM,  
CON> CUSTOMERS_REC, EMPLOYEES_REC, INVENTORY_REC, SUPPLIERS_REC,  
CON> TRANS_REC, CUR_REPORT, CUST_REP, INV_STORE, MAIL_FORMAT,  
CON> MAIL_REPORT, TRANS_WRITE, UPDATE_MASTER, WRITE_PRICE,  
CON> AREA_TABLE, COMPANIES  
DTR>
```

5. Use the **DEFINE DICTIONARY** command to define the application dictionary again:

```
DTR> DEFINE DICTIONARY [INVENTORY]INVENTORY.DIC  
DTR> SHOW DICTIONARY  
The current dictionary is DW1:[INVENTORY]INVENTORY.DIC;2  
DTR>
```

6. Invoke the command file **NEWDIC**, created with the **EXTRACT** command in Step 3, to store all your application definitions in the new dictionary:

```
DTR> @NEWDIC
```

7. Type **SHOW ALL** to confirm the successful execution of **NEWDIC**. Then exit from **PRO/DATATRIEVE** and delete the old version of **INVENTORY.DIC**.

Using the PRO/DATATRIEVE Call Interfaces **3**

This chapter supplements Chapter 3, “Running Programs that Call DATATRIEVE,” in the *DATATRIEVE-11 Call Interface Manual* and is intended for programmers experienced in writing Tool Kit language programs. It describes the steps involved in creating applications that use the callable components of Tool Kit PRO/DATATRIEVE.

Creating an application that calls PRO/DATATRIEVE consists of the following steps:

1. Writing the source language code
2. Compiling the source program
3. Task building the P/OS task image
4. Writing the application installation file
5. Running and debugging the application on a Professional
6. Creating distribution kits

This chapter explains aspects of this development process that are unique to applications that call PRO/DATATRIEVE. For a complete description of each step in the application development process, refer to your Tool Kit user’s guide and the appropriate language documentation.

Writing the Source Program

As explained in Chapter 4 of the *DATATRIEVE-11 Call Interface Manual*, you call PRO/DATATRIEVE from a Tool Kit language by taking the following actions:

- Declaring a PRO/DATATRIEVE Access Block (DAB)
- Initializing the PRO/DATATRIEVE interface
- Checking the PRO/DATATRIEVE state to see which routine PRO/DATATRIEVE expects you to call next
- Calling PRO/DATATRIEVE routines

- Handling errors and displaying messages
- Terminating the call interface

See Chapters 4 through 8 and Appendix A in the *DATATRIEVE-11 Call Interface Manual* for information on writing programs that call PRO/DATATRIEVE and for sample BASIC, COBOL, and FORTRAN programs and DAB inclusion files. Appendix C in this manual contains a sample PASCAL DAB inclusion file and a sample PASCAL program.

Compiling the Source Program

Compile your program using the appropriate Tool Kit language compiler or assembler, as described in your language documentation.

Task Building the Application

Before you can task build your application with the Professional Application Builder (PAB), you must:

1. Create a Task Builder command file (.CMD) and overlay descriptor file (.ODL). See the appropriate language documentation and your Tool Kit user's guide for information on creating .CMD and .ODL files.
2. Edit the Task Builder command file to specify the logical unit numbers (LUNs) your task image will use. The next section in this chapter explains how to customize .CMD files for applications that call PRO/DATATRIEVE. See Chapter 3 in the *DATATRIEVE-11 Call Interface Manual* and your Tool Kit user's guide for more general information on logical unit numbers.
3. Edit the overlay descriptor file to include references to the Call Interface library, DTCLIB, so that your program can be linked with Call Interface routines. Another section in this chapter describes the Call Interface Library and routines you must reference in the .ODL file.
4. Task build your application using the Professional Application Builder. See your Tool Kit user's guide for complete information on Professional Application Builder (PAB) commands.

Editing the Task Builder Command File

The Remote Call Interface uses logical units to establish links between DECnet and the calling program. For this reason, the .CMD file for a task that calls DATATRIEVE on a remote node must specify how many logical units the Remote Call Interface needs and exactly which logical unit numbers (LUNs) it can use. Otherwise, the Remote Call Interface may try to use the same LUNs used by the language processor and P/OS user interface services.

The number of logical units the Remote Call Interface requires depends on the number of logical links your program will have active at the same time and is calculated by adding one to the number of DATATRIEVE Access Blocks (DABs) your program uses. Thus, if your program has only one DAB, you need to make sure two logical units are available and to identify which two LUNs the Remote Call Interface will use. See Chapter 4 in the *DATATRIEVE-11 Call Interface Manual* for more information on the DATATRIEVE Access Block.

The Remote Call Interface has a two-word global storage area called LUNMAP. By adding a GBLPAT line to the CMD file, you identify the LUNs available to the Remote Call Interface via LUNMAP. LUNs allocated to P/OS user interface services and the programming language are not available to the Remote Call Interface and should be specified as zeros in the LUNMAP.

For example, this is a .CMD file for a program named SAMPLE that was generated by the BASIC-PLUS-2 BUILD command:

```

SY: SAMPLE/CP/FP=SY: SAMPLE/MP
TASK = SAMPLE
UNITS = 18 ; decimal _____ ①
ASG = TI:13:15 ; decimal _____ ②
ASG = SY:5:6:7:8:9:10:11:12 ; decimal _____ ②
EXTTSK = 952
CLSTR = PBFSML,POSRES,RMSRES:RO
EXTSCT = DM$BUF:4540 ; octal
EXTSCT = FL$BUF:4310 ; octal
EXTSCT = HL$BUF:3410 ; octal
EXTSCT = MM$BUF:1000 ; octal
EXTSCT = MN$BUF:4540 ; octal
GBLDEF = HL$LUN:21 ; octal _____
GBLDEF = MN$LUN:20 ; octal _____
GBLDEF = MS$LUN:16 ; octal _____ ②
GBLDEF = TT$EFN:1 ; octal
GBLDEF = TT$LUN:15 ; octal
GBLDEF = WC$LUN:22 ; octal
//

```

Comments have been added to the .CMD file to clarify the lines.

The UNITS line (1) allocates a total of 18 (decimal) logical units to the program.

The ASG and GBLDEF lines (2) indicate that P/OS services and BASIC-PLUS-2 use all but three of these logical units, leaving LUNs 2, 3, and 4, available for the Remote Call Interface.

If your program requires more than three LUNs for the Remote Call Interface, you will have to allocate more logical units for the program by changing the UNITS line. If three LUNs are sufficient, you can simply add the following GBLPAT line to the CMD file:

```

GBLPAT = MAIN:LUNMAP:000016:0; LUNs 2, 3, 4 for the CI

```

This line specifies LUNs 2, 3, and 4 for use by the Remote Call Interface and maps LUNMAP as shown in Figure 3-1.

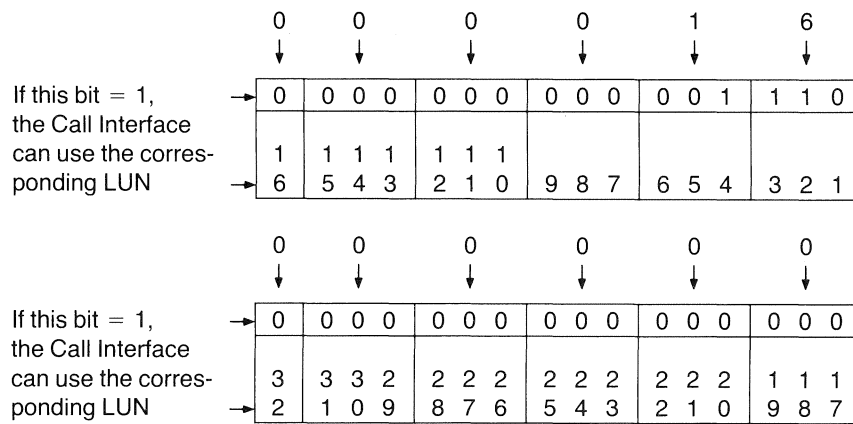


Figure 3-1: LUNMAP Bit Settings

Note

Note that ASG numbers are interpreted as decimal values, while GBLDEF and GBLPAT numbers are interpreted as octal values.

See Chapter 3 in the *DATATRIEVE-11 Call Interface Manual* for more information on LUNMAP.

Editing the Overlay Descriptor (.ODL) File

The .ODL file must reference the Call Interface object module library, DTCLIB.OLB. DTCLIB.OLB is shipped with your Tool Kit software and contains the modules needed to use the Remote and Local Call Interfaces.

The .ODL file must include the following references to DTCLIB.OLB:

- A reference to a particular language module:
 - CIBAS – For programs written in BASIC-PLUS-2
 - CICOB – For programs written in COBOL
 - CIFOR – For programs written in FORTRAN or any other Tool Kit language, such as PASCAL, that uses the FORTRAN method of passing arguments
- A reference to either the remote or local interface module:
 - LCPOS – For programs that call PRO/DATATRIEVE on your Professional
 - NCPOS – For programs that call PRO/DATATRIEVE, VAX DATATRIEVE, DATATRIEVE-11, or DATATRIEVE-20 on a remote system
- A reference to DTCLIB.OLB as a whole

To link your program with DTCLIB routines, you need to reference DTCLIB in the ROOT line and define it in a .FCTR line in the .ODL file. The general format for a .FCTR line that identifies DTCLIB routines is:

$$\text{.FCTR LB:[1,5]DTCLIB/LB: } \left\{ \begin{array}{l} \text{CIBAS} \\ \text{CICOB} \\ \text{CIFOR} \end{array} \right\} : \left\{ \begin{array}{l} \text{LCPOS} \\ \text{NCPOS} \end{array} \right\} \text{-LB:[1,5]DTCLIB/LB}$$

An .ODL file generated by the BASIC-PLUS-2 BUILD command, for example, looks like this:

```
.ROOT BASIC2-RMSROT-USER,RMSALL
USER: .FCTR SY:SAMPLE-LIBR
LIBR: .FCTR LB:[1,5]PBFOFS/LB
@LB:[1,5]PBFIC1
@LB:[1,5]RMSRLX
.END
```

If the program uses the Remote Call Interface, you would edit the .ODL file as follows:

```
USER: .FCTR SY:SAMPLE-LIBR-DTRRCI
DTRRCI: .FCTR LB:[1,5]DTCLIB/LB:CIBAS:NCPOS-LB:[1,5]DTCLIB/LB
```

CIBAS identifies the language-specific module for BASIC, NCPOS identifies the Remote Call Interface routine, and DTCLIB/LB causes the Task Builder to search DTCLIB for any other modules that are needed.

Creating the Application Installation File

As with all Professional applications, you must write an installation file (filename.INS) before you can install the application. The installation file for a COBOL application that uses the Remote Call Interface looks like this:

```
! Installation file for REMSAMP
Name "PRO/REMSAMP"
File REMSAMP.TSK/Delete
File REMSAMP.MNU/Delete
File REMSAMP.HLP/Delete
Install REMSAMP.TSK/Task
Install [ZZSYS]CBILIB.TSK/Library
Assign MENU REMSAMP.MNU
Assign HELP REMSAMP.HLP RH01
Run REMSAMP
```

Installation files for applications that use the Local Call Interface require an additional line to install the callable PRO/DATATRIEVE task image, PROCALDTR.TSK:

```
Install [ZZDTR]PROCALDTR.TSK/Task
```

Installation files for applications using the Local Call Interface may also assign the logical name CALDTR\$INI to an initialization file for the application:

```
Assign Logical CALDTR$INI "REMSAMP.INI"
```

An initialization file for calling PRO/DATATRIEVE locally is optional. There is no default for an initialization file.

Some P/OS services require additional lines in the .INS file. See your Tool Kit user's guide for information specific to the service your program uses and for a complete description of .INS file lines.

Running and Debugging the Application

Use the P/OS menu structure and Fast Install to install your application on a Professional. Fast Install reports the success or failure of the installation process. If the installation process fails, check your .INS file for errors. Note that the .INS file name must be the same as the directory name you give to Fast Install. Correct any errors in the .INS file, rename the file if necessary, and run Fast Install again. Refer to your Tool Kit user's guide for more information on Fast Install.

Once the application has been installed successfully, run and debug it as you would any other Tool Kit language application. Your language documentation describes debugging commands and their use. For information on DATATRIEVE errors, see the *DATATRIEVE-11 Call Interface Manual*.

To help in the debugging process, you can define a log file for PROCALDTR with the DCL ASSIGN command or with the Assign Logical command in the installation file:

```
$ ASSIGN filespec CALDTR$LOG
```

or

```
Assign Logical CALDTR$LOG "filespec"
```

Each time the PROCALDTR task image is invoked, a new version of the log file is created, and you can consult it for information about how the program works.

If you are creating and debugging programs that use the Remote Call Interface to access another Professional, you can define DDMF\$LOG on the remote Professional. The log file is created on the remote Professional, and you can consult that file for information about the program. You can also define DDMF\$LOG on your own Professional to track remote access by calling programs and PRO/RDT.

Creating Distribution Kits

When your application is error-free, run the Application Diskette Builder (ADB) to create diskettes for distribution. See your Tool Kit user's guide for complete information on ADB commands.

Documentation Directory **A**

This appendix describes the manuals in the Tool Kit PRO/DATATRIEVE, PRO/RDT, and PRO/DATATRIEVE documentation sets.

PRO/DATATRIEVE Documentation

The PRO/DATATRIEVE documentation set consists of the following manuals and cards:

- *PRO/DATATRIEVE for Beginners*

Objective

To introduce PRO/DATATRIEVE and data management concepts to new users.

Audience

End users who are not familiar with PRO/DATATRIEVE or with data management.

Summary

The manual provides an overview of information management terms and concepts, and then takes the user step-by-step through the process of using PRO/DATATRIEVE to manage customer information. Tasks introduced in this manual include defining, storing, retrieving, and modifying data, using procedures and command files, and creating reports.

- *PRO/DATATRIEVE Handbook*

Objective

To show how to use PRO/DATATRIEVE to perform advanced data management tasks and to provide practical, task-oriented usage and reference information about PRO/DATATRIEVE.

Audience

People familiar with PRO/DATATRIEVE, the basic concepts of data management, the Professional hard disk computer system, and the PROSE editor.

Summary

The first part of this manual contains information on how you can use PRO/DATATRIEVE to perform advanced data management tasks such as establishing context for PRO/DATATRIEVE statements, retrieving information from list fields, and restructuring a data base. The second part contains reference information on PRO/DATATRIEVE commands, statements, and field definition clauses and many explanatory examples. Language elements are grouped in chapters by function.

- *Distributed PRO/DATATRIEVE Reference Card*

Objective

To describe the distributed components of PRO/DATATRIEVE Version 2.0 and related products.

Audience

People who need to know about PRO/DATATRIEVE's distributed capabilities.

Summary

This card summarizes the distributed components of PRO/DATATRIEVE and provides information about the related products and documentation known collectively as Tool Kit PRO/DATATRIEVE.

- *PRO/DATATRIEVE Summary Card*

Objective

To provide quick reference information on the format of PRO/DATATRIEVE commands, statements, clauses, operators, and RSEs.

Audience

Users familiar with PRO/DATATRIEVE who need information on syntax formats.

Summary

This card shows the formats of PRO/DATATRIEVE commands, statements, and clauses, lists and shows examples of PRO/DATATRIEVE operators, and shows the format of an RSE.

PRO/RDT Documentation

PRO/RDT documentation consists of the *PRO/RDT Handbook* and extensive online help.

Objective

To introduce PRO/RDT concepts to new users.

Audience

End users who want to retrieve data from DATATRIEVE domains on a remote computer system.

Summary

The manual explains how to use PRO/RDT to extract DATATRIEVE data from a Professional hard disk computer system, a VAX/VMS system, a PDP-11 system, or a DECSYSTEM-20. Online help provides information on all PRO/RDT menus, menu options, and forms.

Tool Kit PRO/DATATRIEVE Documentation

The Tool Kit PRO/DATATRIEVE documentation set consists of the following manuals:

- *Tool Kit PRO/DATATRIEVE Documentation Supplement*

Objective

To explain how to use Tool Kit PRO/DATATRIEVE components to create PRO/DATATRIEVE applications for the Professional hard disk system.

Audience

Programmers familiar with PRO/DATATRIEVE or DATATRIEVE-11, the Professional or Host Tool Kit, a Tool Kit programming language, and the Professional P/OS operating system and user interface.

Summary

This manual describes the components of Tool Kit PRO/DATATRIEVE, differences between Tool Kit PRO/DATATRIEVE and DATATRIEVE-11, and the Tool Kit PRO/DATATRIEVE development process, and explains how to use the callable components of Tool Kit PRO/DATATRIEVE.

- *Introduction to DATATRIEVE-11*

Objective

To introduce DATATRIEVE-11 to new users.

Audience

People who are unfamiliar with DATATRIEVE-11 or PRO/DATATRIEVE.

Summary

The manual begins with a brief overview of information management and then, using examples, guides you through basic DATATRIEVE-11 tasks. These include defining, storing, retrieving, and modifying data and using ADT and Guide Mode.

- *DATATRIEVE-11 User's Guide*

Objective

To describe the interactive use of DATATRIEVE-11.

Audience

People who:

- Are familiar with the material covered in *PRO/DATATRIEVE for Beginners* or the *Introduction to DATATRIEVE-11*
- Have previous experience using DATATRIEVE-11 or PRO/DATATRIEVE
- Have experience in application programming but are unfamiliar with DATATRIEVE-11 or PRO/DATATRIEVE

Summary

The first part of the manual explains how to set up DATATRIEVE-11 and is not germane to the use of PRO/DATATRIEVE or the Layered Application Facility. The manual then explains how to use DATATRIEVE-11's data dictionaries and how to use DATATRIEVE-11 to define, retrieve, modify, and restructure data.

- *DATATRIEVE-11 Guide to Writing Reports*

Objective

To explain how to produce reports with either DATATRIEVE-11 or PRO/DATATRIEVE.

Audience

Users who want to produce reports based on data managed by either DATATRIEVE-11 or PRO/DATATRIEVE. The reader should be familiar with the material covered in *PRO/DATATRIEVE for Beginners* or the *Introduction to DATATRIEVE* and with some topics discussed in the *PRO/DATATRIEVE Handbook* or *DATATRIEVE-11 User's Guide*.

Summary

The manual begins with examples of DATATRIEVE-11 reports, and then describes how to write the instructions that specify the format and content of a report. Subsequent chapters discuss more advanced techniques of report writing.

- *DATATRIEVE-11 Reference Manual*

Objective

To provide comprehensive reference information about DATATRIEVE-11.

Audience

Users who have a working knowledge of DATATRIEVE-11 or PRO/DATATRIEVE and who understand the basic concepts of data processing.

Summary

The manual explains in detail the rules governing the use of all the features of DATATRIEVE-11. It describes value expressions, Boolean expressions, functions, record selection expressions, and record and field definition clauses. It presents full descriptions of the commands and statements that comprise the DATATRIEVE-11 language.

- *DATATRIEVE-11 Call Interface Manual*

Objective

To show how to write high-level language programs that call DATATRIEVE.

Audience

Programmers familiar with DATATRIEVE.

Summary

The manual explains how to call DATATRIEVE-11 from within programs written in high-level programming languages such as FORTRAN, COBOL, and BASIC. It also explains how to use DATATRIEVE to access data on remote PDP-11 and VAX-11 systems.

- *DATATRIEVE-11 Pocket Guide*

Objective

To provide quick reference information on the syntax and elements of the DATATRIEVE-11 language.

Audience

Experienced users of DATATRIEVE.

Summary

The manual lists the formats of DATATRIEVE-11 commands, statements, clauses, and value expressions and contains basic information about their use.

PRO/DATATRIEVE Sample Application

B

This sample application, NORMAL MAILER, uses procedures to create a mailing list and print address labels. The NORMAL MAILER application installation file, MAILLIST.INS, looks like this:

```
! Installation file for NORMAL MAILER application
Name "Normal Mailer"
File MAILLIST.DIC/Keep
File MAILLIST.INI/Keep
File MAILLIST.DAT/Keep
Install [ZZDTR]PRODTR.TSK/Task
Assign Logical DTR$DIC "APPL$DIR:MAILLIST.DIC"
Assign Logical DTR$INI "APPL$DIR:MAILLIST.INI"
Assign Logical SCREEN "TI:"
Assign Logical PRINTER "LP:"
RUN DTR
```

The initialization file contains only one line:

```
:INITIALIZE
```

The INITIALIZE procedure contains the following lines:

```
PROCEDURE INITIALIZE
PRINT "Type :INFO for instructions"
DECLARE X_LIST PIC X(20).
DECLARE X_LAST PIC X(15).
DECLARE X_FIRST PIC X(12).
DECLARE X_ANSWER PIC XXX.
READY MAIL_LIST WRITE
END_PROCEDURE
```

When the user selects NORMAL MAILER from a menu, PRO/DATATRIEVE executes the INITIALIZE procedure before displaying the DTR> prompt, so the user sees the following display:

```
PRO/Datatrieve, DEC Query and Report System
Version: V2.0 28-Sep-84
Copyright © 1984 Digital Equipment Corporation
Type HELP for help
Type :INFO for instructions

DTR>
```

The user sees the following when PRO/DATATRIEVE executes the INFO procedure:

*** Normal Mailer ***

The following procedures make up the Normal Mailer application:

:ADD-NAMES Lets you add new people to a list.
:PICK-NAMES Selects a set of people for subsequent procedures.
:PRINT-LABELS Prints out names and addresses in a form suitable for mailing labels. You must run the PICK-NAMES procedure first.

Start a procedure by typing a colon and the procedure name in response to the DTR> prompt. For example:

DTR> :ADD-NAMES

Some procedures (like ADD-NAMES) will cycle indefinitely. To end the procedure, press the <EXIT> key. For example, when you have finished adding names to your list, press <EXIT> when you are asked for another name.

DTR>

The NORMAL MAILER application dictionary, MAILLIST.DIC, contains the procedures identified to the user. It also contains other definitions that the user does not need to know about:

DTR> SHOW ALL
Domains:
MAIL_LIST
Records:
MAIL_LIST_RECORD
Procedures:
ADD_NAMES INFO INITIALIZE PICK_NAMES
PRINT_LABELS
Tables:
The current dictionary is DW1:[ZAP00012]MAILLIST.DIC;1
No established collections
Ready domains:
MAIL_LIST: RMS INDEXED, PROTECTED WRITE
DTR>

These definitions look like this:

```
DTR> SHOW MAIL-LIST
DOMAIN MAIL_LIST
  USING MAIL_LIST_RECORD
ON APPL$DIR:MAILLIST.DAT;
DTR> SHOW MAIL-LIST-RECORD
RECORD MAIL_LIST_RECORD
  USING
  01 MAIL_LIST_RECORD.
    03 LIST-NAME PIC X(20) QUERY-NAME IS LIST.
    03 NAME-FIELDS.
      05 LAST_NAME PIC X(15).
      05 FIRST_NAME PIC X(12).
    03 NAME_FL COMPUTED BY FIRST_NAME!!" "LAST_NAME.
    03 NAME_LF COMPUTED BY LAST_NAME!!" "FIRST_NAME.
    03 STREET PIC X(25).
    03 CITY PIC X(15).
    03 STATE PIC XX.
    03 ZIP PIC X(9).
    03 LOCALE COMPUTED BY CITY!!" "STATE!" "ZIP.
    03 PHONE_NO PIC X(10) EDIT_STRING IS XXX-XXX-XXXX.
    03 BIRTHDAY USAGE IS DATE EDIT_STRING IS DD-MMM-YY.
    03 FILLER PIC X(50).
;
```

(continued on next page)

```

DTR> SHOW ADD-NAMES
PROCEDURE ADD_NAMES
REPEAT 1000
  BEGIN
  PRINT " "
  X_LIST = **,"Which list do you want to add to"
  PRINT " "
  X_LAST = *,"Last Name"
  X_FIRST = *,"First Name"
  IF ANY MAIL-LIST with NAME-FIELDS = X_LAST!X_FIRST THEN
    PRINT " That person is already on a list" ELSE
    STORE MAIL-LIST USING
      BEGIN
      LIST = X_LIST
      LAST-NAME = X_LAST
      FIRST-NAME = X_FIRST
      STREET = *,"Street"
      CITY = *,"City or Town"
      STATE = *,"State (abbreviation)"
      ZIP = *,"Zip Code"
      END
    END
  END_PROCEDURE
DTR> SHOW INFO
PROCEDURE INFO
BEGIN
PRINT "          *** Normal Mailer ***"
PRINT " "
PRINT "The following procedures make up the Normal Mailer application:"
PRINT " "
PRINT " :ADD-NAMES      Lets you add new people to a list."
PRINT " "
PRINT " :PICK-NAMES     Selects a set of people for subsequent procedures."
PRINT " "
PRINT " :PRINT-LABELS   Prints out names and addresses in a form suitable"
PRINT "                for mailing labels. You must run the PICK-NAMES"
PRINT "                procedure first."
PRINT " "
PRINT "Start a procedure by typing a colon and the procedure name in"
PRINT "response to the DTR> prompt. For example:"
PRINT " "
PRINT "      DTR> :ADD-NAMES"
PRINT " "
PRINT "Some procedures (like ADD-NAMES) will cycle indefinitely. To"
PRINT "end the procedure, press the <EXIT> key. For example, when"
PRINT "you have finished adding names to your list, press <EXIT> when you"
PRINT "are asked for another name."
END
END_PROCEDURE
DTR> SHOW PICK-NAMES
PROCEDURE PICK_NAMES
X_LIST = *,"LIST name or blank for all lists"
FIND MAIL_LIST WITH X_LIST = LIST OR X_LIST = " "
END_PROCEDURE
DTR> SHOW PRINT-LABELS
PROCEDURE PRINT_LABELS
FOR CURRENT SORTED BY ZIP, NAME-FIELDS
  BEGIN
  PRINT SKIP 3,
    COL 10, NAME_FL(-),
    COL 10, STREET(-),
    COL 10, LOCALE(-) ON *,"SCREEN or PRINTER"
  END
END_PROCEDURE
DTR>

```


Sample PASCAL DAB **C**

This appendix contains a DATATRIEVE Access Block (DAB) definition for PASCAL programs.

```
const LONG_STRING_LEN = 80;
      SHORT_STRING_LEN = 20;

      STATE_INIT = 0;           { values returned in DAB.STATE }
      STATE_CMD  = 1;
      STATE_PVAL = 2;
      STATE_LINE = 3;
      STATE_MSG  = 4;
      STATE_GETP = 5;
      STATE_PUTP = 6;
      STATE_FREQ = 7;

      ES_WARNING      = 0;      { values returned in DAB.ERR_SEV }
      ES_SUCCESS      = 1;
      ES_ERROR        = 2;
      ES_INFORMATION  = 3;
      ES_SEVERE_ERROR = 4;

type
  LONG_STRING = packed array[1..LONG_STRING_LEN] of char;
  SHORT_STRING = packed array[1..SHORT_STRING_LEN] of char;

  DAB_TYPE = record
    IDI,
    STATE,
    ERR_CODE,
    ERR_SEV : integer;
    FLAGS : packed record
      DAB_ACTIVE,
      DTR11,
      PW_PROMPT,
      STR_OVERFLOW,
      BUF_OVERFLOW : boolean;
    end;
    STR_LEN : integer;
    STRING : packed array[1..DTR$STRING_LEN] of char;
    BUFFER : packed array[1..DTR$BUFFER_LEN] of char;
  end;

var DAB : DAB_TYPE;
    DTR_MSG_BUFFER : LONG_STRING;
    DTR_ACTUAL_LEN : integer;
```

(continued on next page)

```

[external(DTCMD)] Procedure DTR$COMMAND (var DAB : DAB_TYPE;
    var CMD_STR : [unsafe, readonly] LONG_STRING;
    var CSTR_LEN : [readonly] integer);
    seq11;

[external(DTCMD)] Procedure DTR$COMMAND_1ARG (var DAB : DAB_TYPE;
    var CMD_STR : [unsafe, readonly] LONG_STRING;
    var CSTR_LEN : [readonly] integer;
    var ARG1 : [unsafe, readonly] LONG_STRING;
    var ARG1_LEN : [readonly] integer);
    seq11;

[external(DTCMD)] Procedure DTR$COMMAND_2ARG (var DAB : DAB_TYPE;
    var CMD_STR : [unsafe, readonly] LONG_STRING;
    var CSTR_LEN : [readonly] integer;
    var ARG1 : [unsafe, readonly] LONG_STRING;
    var ARG1_LEN : [readonly] integer;
    var ARG2 : [unsafe, readonly] LONG_STRING;
    var ARG2_LEN : [readonly] integer);
    seq11;

[external(DTCMD)] Procedure DTR$COMMAND_3ARG (var DAB : DAB_TYPE;
    var CMD_STR : [unsafe, readonly] LONG_STRING;
    var CSTR_LEN : [readonly] integer;
    var ARG1 : [unsafe, readonly] LONG_STRING;
    var ARG1_LEN : [readonly] integer;
    var ARG2 : [unsafe, readonly] LONG_STRING;
    var ARG2_LEN : [readonly] integer;
    var ARG3 : [unsafe, readonly] LONG_STRING;
    var ARG3_LEN : [readonly] integer);
    seq11;

[external(DTCONT)] Procedure DTR$CONTINUE (var DAB : DAB_TYPE);
    seq11;

[external(DTFINI)] Procedure DTR$FINISH (var DAB : DAB_TYPE);
    seq11;

[external(DTFREQ)] Procedure DTR$FIELD_REQUEST (var DAB : DAB_TYPE;
    var DOMAIN_NAME : [unsafe, readonly] LONG_STRING;
    var DOM_NAME_LEN : [readonly] integer;
    var INFO_BUFFER : [unsafe] LONG_STRING;
    var BUFFER_LEN : integer);
    seq11;

[external(DTGETP)] Procedure DTR$GET_PORT (var DAB : DAB_TYPE;
    var RECORD_BUF : [unsafe] LONG_STRING;
    var RBUF_LEN : [readonly] integer;
    var RECORD_LEN : integer);
    seq11;

[external(DTINIT)] Procedure DTR$INITIALIZE (var DAB : DAB_TYPE;
    var STR_LEN,
    var BUFF_LEN : [readonly] integer;
    var NODE : [unsafe, readonly] LONG_STRING;
    var N_LEN,
    var OPTIONS : [readonly] integer);
    seq11;

[external(DTLINE)] Procedure DTR$GET_PRINT_LINE (var DAB : DAB_TYPE;
    var RECORD_BUF : [unsafe, readonly] LONG_STRING;
    var RBUF_LEN : [readonly] integer;
    var RECORD_LEN : integer);
    seq11;

```

(continued on next page)


```
[external(DTMSG)] procedure DTR$GET_MSG (var DAB : DAB_TYPE;  
    var RECORD_BUF : [unsafe, readonly] LONG_STRING;  
    var RBUF_LEN : [readonly] integer;  
    var RECORD_LEN : integer);  
  
    seq11;  
  
[external(DTPEOF)] procedure DTR$PUT_EOF (var DAB : DAB_TYPE);  
    seq11;  
  
[external(DTPUTP)] procedure DTR$PUT_PORT (var DAB : DAB_TYPE;  
    var RECORD_BUF : [unsafe, readonly] LONG_STRING;  
    var RBUF_LEN : [readonly] integer);  
  
    seq11;  
  
[external(DTPVAL)] procedure DTR$PUT_VALUE (var DAB : DAB_TYPE;  
    var RECORD_BUF : [unsafe, readonly] LONG_STRING;  
    var RBUF_LEN : [readonly] integer);  
  
    seq11;  
  
[external(DTUNWD)] procedure DTR$UNWIND (var DAB : DAB_TYPE);  
    seq11;
```


Index

<>

See Angle brackets

@

See AT sign

□

See Square brackets

A

ADB

See Application Diskette Builder

ALLOCATION clause

LEFT_RIGHT, 1-6

MAJOR_MINOR, 1-6

Angle brackets (<>)

in directory names, 2-2

APPL\$DIR logical name, 2-3, 2-4

Application directories, 2-3

Application Diskette Builder (ADB), 2-1, 3-6

Application files, 2-1

dictionary, 2-4

file types of, 2-3

format for installation, 2-5

initialization, 2-4

installation, 2-1, 2-3, 2-5 to 2-6, 3-5

minimum number, 2-2

naming, 2-2, 2-5

sample installation, 2-6

ASG line

in CMD files, 3-3

ASSIGN LOGICAL command line, 2-3, 2-5

Assigning logical names, 2-4

for DDMF log files, 3-6

for PROCALDTR log files, 3-6

At (@) sign

using to invoke NEWDIC, 2-7

B

BASIC-PLUS-2, 3-4

C

CALDTR\$INI logical name, 3-5

CALDTR\$LOG logical name, 3-6

Call Interface, 1-3, 1-7

Local, 1-1, 1-2, 1-3, 1-4

Object Module Library, 1-1, 1-3, 3-4

Remote, 1-1, 1-3, 1-4

Callable PRO/DATATRIEVE, 1-1, 1-4

installing, 3-5

Calling

a remote DATATRIEVE, 1-2, 1-3

local PRO/DATATRIEVE, 1-2, 1-3

CANCEL key, 1-6

CIBAS object module, 3-4

CICOB object module, 3-4

CIFOR object module, 3-4

CMD file type, 2-3, 3-2

COBOL, 3-4

Command files, 2-3, 2-7

Compiling the source program, 3-2

Compressing data dictionaries, 2-1, 2-6

CTRL/C

interrupt key, 1-7

CTRL/Z

exit key, 1-7

Customizing applications

with logical names, 1-2

D

DAB

See DATATRIEVE Access Block

DAT file type, 2-3
 Data alignment defaults, 1-6
 DATATRIEVE Access Block (DAB), 3-1, 3-3
 DDMF
 See Distributed Data Manipulation Facility
 DDMF\$LOG logical name, 3-6
 Defaults
 data alignment, 1-6
 file type, 2-3
 DEFINE DOMAIN command, 2-3, 2-4
 DELETE option
 with FILE command, 2-5
 DIC file type, 2-3
 Dictionary files, 2-3, 2-4
 compressing, 2-1, 2-6
 default, 2-3
 logical name for, 2-4
 Directories, 2-2
 application, 2-3
 for PRO/DATATRIEVE, 2-4
 Distributed Data Manipulation Facility
 (DDMF)
 creating log files, 3-6
 on remote nodes, 1-4
 on the Professional, 1-2
 DO key, 1-6
 DTCLIB.OLB
 Call Interface Library, 1-1, 1-3, 3-4
 call interface modules, 3-4
 language modules, 3-4
 specifying modules in, 3-5
 DTR\$DIC logical name, 2-4, 2-5
 DTR\$INI logical name, 2-4

E

Editing
 overlay descriptor files, 3-2, 3-4, 3-5
 task builder command files, 3-2 to 3-4
 EXIT key, 1-7
 EXTRACT command, 2-3

F

Fast Install, 2-5, 3-6
 FCTR line
 in ODL files, 3-5
 FILE command line
 /DELETE option, 2-5
 /KEEP option, 2-5
 File types
 CMD, 3-2
 default, 2-3

File types (cont.)
 INS, 3-5
 ODL, 3-2
 of application files, 2-3
 of installation files, 2-5
 Files
 application, 2-1, 2-2
 compressing dictionary, 2-6
 data, 2-2, 2-3
 dictionary, 2-4
 format for installation, 2-5
 initialization, 2-4
 invoking command, 2-7
 naming application, 2-2
 naming installation, 2-5, 3-6
 PRO/DATATRIEVE, 2-2
 PRO/DATATRIEVE message, 2-5
 PRO/DATATRIEVE task image, 2-5
 sample CMD, 3-3
 sample installation, 3-5
 writing installation, 2-5 to 2-6, 3-5
 FORTRAN, 3-4
 Function keys, 1-5, 1-6
 CANCEL, 1-6
 DO, 1-6
 EXIT, 1-7
 HELP, 1-6
 INTERRUPT/DO, 1-7
 MAIN SCREEN, 1-6

G

GBLDEF line
 in CMD files, 3-3
 GBLPAT line
 in CMD files, 3-3

H

HELP key, 1-6

I

INI file type, 2-3
 Initialization file, 2-3
 logical name for, 2-4
 Initialization files
 Local Call Interface, 3-5
 INS file type, 2-5, 3-5
 INSTALL command line
 /TASK option, 2-5

Installing

- application files, 3-5
- PROCALDTR.TSK, 3-5

Interactive PRO/DATATRIEVE, 1-1, 1-2

INTERRUPT/DO keys, 1-7

K

KEEP option

- with FILE command, 2-5

L

LCPOS object module, 1-3, 1-7, 3-4

LEFT_RIGHT data alignment, 1-6

Linking

- programs, 3-5
- with Call Interface modules, 1-3
- with LCPOS, 1-3
- with NCPOS, 1-3

Local Call Interface, 1-1, 1-2

initializing, 3-5

installation file for, 3-5

LCPOS module, 1-3, 1-7, 3-4

requirements for using, 1-3

using, 1-4

Log files

DDMF, 3-6

PROCALDTR, 3-6

Logical names, 1-2, 1-5, 1-6, 2-1, 2-3 to 2-4

APPL\$DIR, 2-3, 2-4

assigning, 2-4

CALDTR\$INI, 3-5

CALDTR\$LOG, 3-6

DDMF\$LOG, 3-6

DTR\$DIC, 2-4, 2-5

DTR\$INI, 2-4

in DEFINE DOMAIN command, 2-4

rules for using, 2-3

using, 2-4

Logical Unit Numbers

allocating, 3-2

Logical unit numbers

allocating, 3-2

LUN

See Logical Unit Numbers

LUNMAP

storage area, 3-3, 3-4

M

MAIN SCREEN key, 1-6

MAJOR_MINOR data alignment, 1-6

N

NAME command line, 2-5

Names

APPL\$DIR logical, 2-3

CALDTR\$INI logical, 3-5

CALDTR\$LOG logical, 3-6

DDMF\$LOG logical, 3-6

DTR\$DIC logical, 2-4

DTR\$INI logical, 2-4

logical, 2-1, 2-3 to 2-4

PRO/DATATRIEVE file, 2-2

reserved, 2-2

Naming

application files, 2-2

application installation files, 2-5, 3-6

directories, 2-2

volumes, 2-2

NC11M

object module, 1-7

NCPOS object module, 1-3, 1-7, 3-4

NCRSTS

object module, 1-7

O

ODL file type, 3-2

Overlay descriptor files

creating, 3-2

editing, 3-2, 3-4 to 3-5

sample, 3-5

P

PAB

See Professional Application Builder

PASCAL, 3-4

PRO/DATATRIEVE

file names, 2-2

message file, 2-5

task image, 2-5

PRO/DDMF

Distributed Data Manipulation Facility, 1-2

PRO/DECnet

on a Professional, 1-2, 1-3

PRO/RDT

Remote Data Transfer, 1-1, 1-2

PROCALDTR.TSK

Callable PRO/DATATRIEVE, 1-1, 1-4

PRODTR.DIC
dictionary file, 2-4
PRODTR.TSK
interactive PRO/DATATRIEVE, 1-1, 1-2, 2-5
Professional Application Builder (PAB), 3-2

R

Remote Call Interface, 1-1
installation file for, 3-5
NCPOS module, 1-3, 1-7, 3-4
requirements for using, 1-3
specifying LUNs for, 3-2
using, 1-4
Reserved names, 2-2
ROOT line
in ODL files, 3-5
RUN command line, 2-5

S

Square brackets ([])
in directory names, 2-2

T

Task Builder command file
creating, 3-2
editing, 3-2
sample, 3-3

Task Builder command files
editing, 3-2 to 3-4
Task building applications, 3-2
TASK option
with TASK command, 2-5
TI:
video screen, 2-4

U

UNITS line
in CMD files, 3-3

V

Volumes, 2-2

W

Writing
application installation files, 3-5
source programs, 3-1

Z

ZZDTR directory, 2-4

Master Index

This index contains entries for the entire Tool Kit PRO/DATATRIEVE documentation set. Manuals are indicated by the following abbreviations:

CALL	Call Interface Manual
DOCSUP	Tool Kit PRO/DATATRIEVE Documentation Supplement
INTRO	Introduction to DATATRIEVE-11
REF	Reference Manual
REPORT	Guide to Writing Reports
UG	User's Guide

Page numbers followed by a "t" contain tables; those followed by "f" contain figures.

Master Index

:
 See Colon

<
 See LESS_THAN relational operator

<>
 See Angle brackets

>
 See GREATER_THAN relational operator

@
 See AT sign

[]
 See Square brackets

A

ABORT statement, *REF* 5-15 to 5-18
 effect of SET ABORT, *REF* 5-15
 in command file, *UG* 10-5
 in STORE statement, *REF* 5-217

Aborting commands, DTUNWD, *CALL* 4-34, 8-29

Aborting procedures, *UG* 9-12

Access Block, DATATRIEVE
 See DATATRIEVE Access Block

Access codes, *REF* 5-67t

Access control list
 access codes, *REF* 5-67t
 adding entries to, *UG* 20-11
 contents, *UG* 20-1 to 20-6
 control (C) privilege, *UG* 21-2
 creating, *UG* 20-6 to 20-7
 creating for ports, *REF* 5-55
 creating for procedures, *REF* 5-57
 creating for records, *REF* 5-61
 creating for RMS domains, *REF* 5-46
 creating for tables, *REF* 5-64
 creating for view domains, *REF* 5-48
 defining entries, *REF* 5-66 to 5-68

Access control list (cont.)
 deleting entries, *REF* 5-71 to 5-72, *UG* 20-11
 displaying, *REF* 5-209 to 5-210, *UG* 20-10, 21-2
 execute (E) privilege, *UG* 21-2
 key, *UG* 20-1 to 20-6
 lock type, *UG* 20-1 to 20-6
 maintaining, *UG* 20-9 to 20-11, 21-6
 modify (M) privilege, *UG* 21-2
 privileges, *UG* 20-4 to 20-6
 processing, *UG* 20-7 to 20-8
 protecting data, *UG* 20-1
 read (R) privilege, *UG* 21-2
 sample, *UG* 20-1F
 sequence number, *UG* 20-1 to 20-6
 SHOWP command, *REF* 5-209 to 5-210
 write (W) privilege, *UG* 21-2

Access modes, *REF* 5-170t
 EXTEND, *REF* 5-170
 MODIFY, *REF* 5-170
 READ, *REF* 5-170
 required by DATATRIEVE statements, *REF* 5-172t
 WRITE, *REF* 5-170

Access options, *REF* 5-169t

Accessing domains, *INTRO* 2-4 to 2-5
 EXCLUSIVE, *REF* 5-169
 PROTECTED, *REF* 5-169
 READY command, *REF* 5-169 to 5-174
 restrictions, *REF* 5-170
 results, *REF* 5-171
 SHARED, *REF* 5-169

Accounts
 default DECnet, *CALL* 2-1
 specifying to the Remote Terminal Interface, *CALL* 2-1

ACL
 See Access control list

ADB
 See Application Diskette Builder

ADT
 See Application Design Tool

ADT command, *INTRO* 3-6, *REF* 5-19, *UG* 10-2
 See Also Application Design Tool

ADVANCED HELP command, *UG* 2-7

Alias
 domain name, *REF* 5-169

ALIGNED_MAJOR_MINOR allocation, *REF* 5-20

ALL
 in RSE, *REF* 3-4
 in SHOW command, *REF* 5-206
 print list element, *REF* 5-158, 5-160

Allocating LUNs, *CALL* 3-7

ALLOCATION clause, *REF* 5-20 to 5-22, *UG* 6-6
 ALIGNED_MAJOR_MINOR, *REF* 5-20
 in DEFINE FILE command, *REF* 5-50
 in DEFINE RECORD command, *REF* 5-60
 LEFT_RIGHT, *REF* 5-20, *DOCSUP* 1-6
 MAJOR_MINOR, *REF* 5-20, *DOCSUP* 1-6

Alphanumeric fields, *UG* 5-10
 edit string characters, *REF* 5-104 to 5-106
 picture string characters, *REF* 5-156

AND Boolean operator, *INTRO* 6-5, *REF* 2-20, *UG* 7-9

Angle brackets (<>)
 in directory names, *DOCSUP* 2-2

ANY relational operator, *REF* 2-19

APPL\$DIR logical name, *DOCSUP* 2-3, 2-4

Application Design Tool, *REF* 4-1, *UG* 4-1
 defining data with, *INTRO* 3-1 to 3-9
 defining records, *UG* 5-1
 exiting, *REF* 5-19
 invoking, *REF* 5-19
 sample session, *INTRO* 3-4 to 3-8
 use in definitions, *REF* 1-10

Application directories, *DOCSUP* 2-3

Application Diskette Builder (ADB), *DOCSUP* 2-1, 3-6

Application files, *DOCSUP* 2-1
 dictionary, *DOCSUP* 2-4
 file types of, *DOCSUP* 2-3
 format for installation, *DOCSUP* 2-5
 initialization, *DOCSUP* 2-4
 installation, *DOCSUP* 2-1, 2-3, 2-5 to 2-6, 3-5
 minimum number, *DOCSUP* 2-2
 naming, *DOCSUP* 2-2, 2-5

Application files, (cont.)
 sample installation, *DOCSUP* 2-6

Argument list, *CALL* 8-8f

Arguments
 in procedures, *UG* 9-4 to 9-5

Arithmetic expressions, *REF* 2-14 to 2-15
 evaluating, *REF* 2-14
 minus sign in, *REF* 2-14
 parentheses in, *REF* 2-14
 valid operators, *REF* 2-14t

Arithmetic operators, *REF* 2-14t

ASCENDING
 in SORT statement, *REF* 5-213, 5-223
 sort key, *INTRO* 6-8, *REF* 3-7

ASCII collating sequence, *REF* 2-18, 3-8

ASG line
 in CMD files, *DOCSUP* 3-3

ASSIGN LOGICAL command line,
 DOCSUP 2-3, 2-5

Assigning logical names, *DOCSUP* 2-4
 for DDMF log files, *DOCSUP* 3-6
 for PROCALDTR log files, *DOCSUP* 3-6

Assigning query names
 using the ADT command, *INTRO* 3-2

Assignment statement, *REF* 5-23 to 5-30, *UG* 11-2
 for elementary fields, *REF* 5-23 to 5-25
 for group fields, *REF* 5-25 to 5-27
 for variables, *REF* 5-27 to 5-30

Asterisk (*)
 edit string character, *REF* 5-107

AT (@) sign
 executing command files, *UG* 5-3
 using to invoke command files, *REF* 5-13 to 5-14

At (@) sign
 using to invoke NEWDIC, *DOCSUP* 2-7

AT BOTTOM statement (Report Writer), *REF* 5-186 to 5-188
 OF field-name, *REPORT* 2-17, 3-3, 3-8 to 3-9
 OF PAGE, *REPORT* 2-17
 OF REPORT, *REPORT* 2-17, 2-18, 3-14
 summary elements, *REF* 5-187t

AT TOP statement (Report Writer), *REF* 5-188 to 5-191
 header and summary elements, *REF* 5-189t
 OF field-name, *REPORT* 3-3
 OF PAGE, *REPORT* 2-17, 3-10 to 3-13
 OF REPORT, *REPORT* 2-17

AVERAGE statistical function, *REPORT* 1-9, 2-17 to 2-18, *REF* 2-11
 in AT BOTTOM statement (RW), *REF* 5-187
 in AT TOP statement (RW), *REF* 5-189

B

- B (blank)
 - edit string character, *REF* 5-105
- BASIC-PLUS-2, *DOCSUP* 3-4
- BEGIN-END statement, *REF* 5-31 to 5-34, *UG* 8-2 to 8-6
 - declaring variables in, *REF* 5-39
 - initializing variables, *REF* 5-32
 - invoking procedures from, *REF* 5-10
 - restrictions, *REF* 5-31
 - results, *REF* 5-31
 - usage notes, *REF* 5-32
 - with REPEAT statement, *REF* 5-32
- BETWEEN relational operator, *INTRO* 6-4, *REF* 2-17, *UG* 7-6
- Boolean expressions, *REF* 2-16 to 2-21
 - Boolean operators, *REF* 2-20 to 2-21
 - commas in, *REF* 3-7
 - compound, *REF* 2-21, 2-21t, *UG* 7-9
 - defined, *REF* 2-1
 - evaluating, *REF* 2-21
 - grouping with parentheses, *REF* 2-21
 - in RSE, *REF* 3-6
 - relational operators, *REF* 2-17 to 2-20, 2-20t
 - using, *REF* 2-16
- Boolean operators, *REF* 2-20 to 2-21, *UG* 7-9
- Brackets []
 - used as syntax prompts, *UG* 2-5
- BT
 - See* BETWEEN relational operator
- BUFLN, parameter to DTINIT, *CALL* 4-11
- BUT Boolean operator, *INTRO* 6-5, *REF* 2-20, *UG* 7-9

C

- CALDTR\$INI logical name, *DOCSUP* 3-5
- CALDTR\$LOG logical name, *DOCSUP* 3-6
- Call Interface, *REF* 1-12, *DOCSUP* 1-3, 1-7, *UG* 1-6, *CALL* 1-4, 1-5f
 - closing, *CALL* 4-34
 - creating menus, *CALL* 1-5
 - initializing, *CALL* 4-13, 8-16
 - Local, *DOCSUP* 1-1, 1-2, 1-3, 1-4
 - Object Module Library, *DOCSUP* 1-1, 1-3, 3-4
 - overview, *CALL* 4-1
 - procedures used with, *CALL* 1-5
 - Remote, *DOCSUP* 1-1, 1-3, 1-4
 - tables used with, *CALL* 1-5
 - writing programs that use, *CALL* 4-1 to 4-34

- Callable PRO/DATATRIEVE, *DOCSUP* 1-1, 1-4
 - installing, *DOCSUP* 3-5
- Calling
 - a remote DATATRIEVE, *DOCSUP* 1-2, 1-3
 - local PRO/DATATRIEVE, *DOCSUP* 1-2, 1-3
- Calls to DATATRIEVE
 - DTCMD, *CALL* 4-14, 8-9
 - DTCONT, *CALL* 4-27, 4-28, 8-11
 - DTFINI, *CALL* 4-34, 8-13
 - DTGETP, *CALL* 4-26, 8-14
 - DTINIT, *CALL* 4-13, 8-16
 - DTLINE, *CALL* 4-17, 8-20
 - DTMSG, *CALL* 4-17, 8-21
 - DTPEOF, *CALL* 4-28, 8-23
 - DTPUTP, *CALL* 4-28, 8-25
 - DTPVAL, *CALL* 4-22, 8-27
 - DTUNWD, *CALL* 4-34, 8-29
- CANCEL key, *DOCSUP* 1-6
- Case sensitivity
 - with relational operators, *UG* 7-5
- CHANGE, *UG* 6-7
 - in DEFINE FILE command, *REF* 5-51
- Character set, *REF* 1-4
- Character string literals
 - See* Literals
- Characters
 - edit string, *REF* 5-102
 - floating edit string, *REF* 5-111
 - picture string, *REF* 5-156
 - printing, *REF* 2-2
- CIBAS object module, *DOCSUP* 3-4
- CICOB object module, *DOCSUP* 3-4
- CIFOR object module, *DOCSUP* 3-4
- Clauses
 - in procedures, *UG* 9-4 to 9-5
- CLOSE command, *REF* 5-35
- Closing the Call Interface, *CALL* 4-34
- CMD file type, *DOCSUP* 2-3, 3-2
- !CMD, substitution directive, *CALL* 4-15
- COBOL, *DOCSUP* 3-4
- COL
 - in AT BOTTOM statement (RW), *REF* 5-187
 - in AT TOP statement (RW), *REF* 5-189
 - print list element, *REF* 5-160
 - RW print list element, *REF* 5-193
- Collections, *INTRO* 6-10 to 6-21
 - as record source, *REF* 3-2
 - changing content of, *INTRO* 6-19 to 6-20
- CURRENT, *REF* 3-2
- displaying, *INTRO* 6-11 to 6-12
- erasing, *INTRO* 8-2
- establishing with FIND, *REF* 3-2

- Collections, (cont.)
 - forming, *INTRO* 6-11, *REF* 5-121 to 5-122
 - making new CURRENT from old, *INTRO* 6-14 to 6-15
 - naming, *INTRO* 6-15 to 6-17
 - releasing, *INTRO* 6-20 to 6-21, *REF* 5-177 to 5-179
 - removing records from, *REF* 5-75 to 5-79
 - renaming, *INTRO* 6-18 to 6-19
 - SHOW command, *INTRO* 6-14
 - showing, *REF* 5-206
 - showing names of, *INTRO* 6-17 to 6-18
 - sorting, *INTRO* 6-12 to 6-14, *REF* 5-213 to 5-214
- Colon (:)
 - using to invoke procedures, *REF* 5-10 to 5-12, *UG* 1-4
- Column headers, *REPORT* 2-10, 2-15 to 2-17
 - in AT TOP statement (RW), *REF* 5-189
 - printing, *REF* 5-161, 5-162, 5-193
 - specifying, *REPORT* 2-16 to 2-17
 - suppressing, *REPORT* 2-15, *REF* 5-161, 5-162, 5-193
 - wrapping, *REPORT* 2-13
- Column-page setting
 - See *SET COLUMNS_PAGE* command
- COLUMN_HEADER, *REPORT* 3-12
- Comma (,)
 - edit string character, *REF* 5-110
- Command files, *REPORT* 2-18 to 2-19, *DOCSUP* 2-3, 2-7, *UG* 1-4, 10-1 to 10-9
 - aborting, *UG* 10-5
 - comments, *REF* 5-14, *UG* 10-3
 - compared with procedures, *UG* 10-1
 - contents, *UG* 10-2 to 10-3
 - creating, *UG* 10-2
 - creating with EXTRACT, *REF* 5-118
 - default file type, *REF* 5-14
 - editing, *UG* 10-5
 - editing with, *UG* 17-1
 - in FOR and REPEAT statements, *UG* 10-8
 - in procedures, *REF* 5-13
 - invocation command lines, *UG* 10-3 to 10-4
 - invoking, *REF* 5-13 to 5-14, *UG* 10-3 to 10-5
 - maintaining, *UG* 10-8 to 10-9
 - nesting, *UG* 10-7 to 10-8
 - restrictions on invoking, *UG* 10-4 to 10-5
 - sample, *UG* 10-5 to 10-7
 - SET ABORT/SET NO ABORT, *UG* 10-5
 - stopping execution of, *REF* 5-15
 - uses, *UG* 10-1
- Commands and statements, *REF* 1-2 to 1-7, *UG* 1-3 to 1-4
 - alphabetic summary, *REF* 5-3t
- Commands and statements, (cont.)
 - complex statements, *REF* 1-3
 - compound, *REF* 1-3
 - continuing, *REF* 1-6 to 1-7
 - differences between, *REF* 1-2
 - elements of, *REF* 1-5
 - functional summary, *REF* 5-6t
 - functions of, *REF* 1-2
 - in procedures, *UG* 9-3 to 9-4
 - in QUERY.INI file, *UG* 2-2
 - nested, *REF* 1-3
 - structure of, *REF* 1-2
 - syntax, *REF* 5-1
 - terminating, *REF* 1-6 to 1-7
- Commands, passing, using DTCMD, *CALL* 4-14, 8-9
- Comments, *REF* 1-7
 - in command files, *REF* 5-14
- COMP data type, *REF* 5-229
- COMP-1 data type, *REF* 5-229
- COMP-2 data type, *REF* 5-229
- COMP-3 data type, *REF* 5-229
- COMP-5 data type, *REF* 5-230
- Compiling programs that call DATATRIEVE, *CALL* 3-1
- Compiling the source program, *DOCSUP* 3-2
- Components of DATATRIEVE-11, *CALL* 1-1 to 1-2
- Compound Boolean expressions, *REF* 2-21
 - evaluating, *REF* 2-21t
- Compound statements, *REF* 1-3, *UG* 1-3, 8-1 to 8-6
 - BEGIN-END, *REF* 5-31 to 5-34, *UG* 8-4 to 8-6
 - FOR in, *UG* 8-3 to 8-4
 - REPEAT in, *UG* 8-1 to 8-3
 - THEN, *REF* 5-226 to 5-227
 - variables in, *REF* 2-8
- Compressing data dictionaries, *DOCSUP* 2-1, 2-6
- Compressing data dictionary, *UG* 21-4 to 21-5
- COMPUTED BY clause, *REF* 5-36 to 5-37, *UG* 5-1, 5-11
 - creating virtual fields, *REF* 2-6, 5-36
 - in DECLARE statement, *REF* 5-38
 - in field definition, *REF* 4-7
- CON> prompt, *REF* 1-8, *UG* 2-4
- Concatenated expressions, *REPORT* 3-4, *REF* 2-2, 2-15 to 2-16
 - formatting output, *REF* 2-16
- Conditional transfers
 - with IF-THEN-ELSE statement, *REF* 1-3
- Conserving memory
 - See *Optimizing workspace*

CONT

- See CONTAINING relational operator*
- CONTAINING relational operator, *INTRO* 6-4, *REF* 2-17, *UG* 7-5
 - case-sensitivity, *REF* 2-18
- Context, *UG* 12-1
 - establishing, *UG* 12-1 to 12-13
- Context block
 - content of, *UG* 12-2 to 12-4
 - existing collections, *UG* 12-4 to 12-6
 - global variables, *UG* 12-4
 - record streams, *UG* 12-6 to 12-7
- Context stack, *UG* 12-2 to 12-8
 - lasting changes, *UG* 12-6
 - left, *UG*
- Context variables, *REF* 2-7, *UG* 12-8 to 12-11
 - field name qualifiers, *UG* 12-8
 - in FIND statement, *REF* 3-6
 - in RSE, *REF* 3-6
 - qualifying field names, *REF* 2-6
 - with FOR loops, *REF* 3-6
 - with MODIFY statement, *UG* 12-13
 - with STORE statement, *REF* 5-219, *UG* 12-13
- Continuation character, *REF* 1-6 to 1-7
 - in character string literals, *REF* 2-2
- Continuation prompt, *REF* 1-8
- Continuing, DTCONT, *CALL* 4-27, 4-28, 8-11
- Control groups, *REPORT* 3-2 to 3-9
 - levels of, *REPORT* 3-4 to 3-8
 - sort keys, *REPORT* 3-2 to 3-8
 - without sort keys, *REPORT* 3-8
- Controlling output, *UG* 19-1 to 19-5
 - column width, *UG* 19-1 to 19-4
- Copying domains, *CALL* 2-3 to 2-4
- Corruption of data
 - protection against, *UG* 20-1
- COUNT statistical function, *REPORT* 1-9, 2-17 to 2-18, *REF* 2-11
 - in AT BOTTOM statement (RW), *REF* 5-187
 - in AT TOP statement (RW), *REF* 5-189
- CR (credit)
 - edit string character, *REF* 5-109
- CREATE DICTIONARY command, *UG* 21-1
- Cross tabulations, *REPORT* 3-13 to 3-14
- CTRL/C, *UG* 2-6
 - interrupt key, *DOCSUP* 1-7
 - with BEGIN-END block, *REF* 5-32
- CTRL/Z, *INTRO* 2-5, 5-3
 - exit key, *DOCSUP* 1-7
 - exiting from DATATRIEVE, *REF* 5-116, *UG* 2-6

CURRENT

- as target record stream, *UG* 12-20 to 12-21
- in record selection expressions, *INTRO* 6-15
- in SHOW command, *REF* 5-206
- Customizing applications
 - with logical names, *DOCSUP* 1-2

D

- DAB
 - See DATATRIEVE Access Block*
- DAB fields, *CALL* 8-6t
 - DAB\$_BUFFER, *CALL* 8-16
 - DAB\$_STRING, *CALL* 8-7t, 8-9
 - DAB\$_ERR_CODE, *CALL* 4-18, 8-5, 8-21
 - DAB\$_ERR_SEV, *CALL* 4-18, 8-5, 8-21
 - DAB\$_STR_LEN, *CALL* 8-7, 8-16
- DAB\$_M_BUF_OVERFLOW flag, *CALL* 8-20
- DAB\$_M_STR_OVERFLOW flag, *CALL* 8-7
- DAB\$_V_BUFFER, *CALL* 8-16
- DAB\$_V_STRING, *CALL* 8-7t, 8-9
- DAB\$_W_ERR_CODE, *CALL* 4-18, 8-5
- DAB\$_W_ERR_SEV, *CALL* 4-18, 8-5, 8-21
- DAB\$_W_FLAGS, *CALL* 8-6t
- DAB\$_W_STR_LEN, *CALL* 8-7, 8-16
- DAT file type, *DOCSUP* 2-3
- Data alignment defaults, *DOCSUP* 1-6
- Data dictionary, *UG* 1-2
 - access privileges, *UG* 21-2
 - changing, *UG* 3-2 to 3-3
 - contents, *UG* 3-1 to 3-2
 - creating, *INTRO* 3-9, *UG* 2-2, 3-2
 - default type (.DIC), *UG* 3-2
 - defining, *REF* 5-43 to 5-44
 - deleting, *REF* 5-69
 - deleting definitions in, *UG* 21-3 to 21-4
 - determining size of, *UG* 21-4
 - displaying, *UG* 3-3, 21-2
 - editing definitions, *UG* 21-1
 - extracting from, *UG* 21-6 to 21-7
 - function, *UG* 3-1
 - maintaining, *UG* 21-1 to 21-7
 - modifying, *UG* 21-2 to 21-3
 - objects, *UG* 21-1
 - optimizing disk storage, *UG* 21-4 to 21-5
 - placing definitions into, *INTRO* 3-8 to 3-9
 - QCPRS utility, *UG* 21-4
 - security, *UG* 20-1 to 20-11
 - setting, *UG* 3-3
 - showing current, *REF* 5-44, 5-206
 - transferring definitions, *UG* 21-1
- Data files
 - See Files*

- Data types, *INTRO* 3-2 to 3-3
 - COMP (INTEGER), *REF* 5-229
 - COMP-1 (REAL), *REF* 5-229
 - COMP-2 (DOUBLE), *REF* 5-229
 - COMP-3 (PACKED), *REF* 5-229
 - COMP-5 (ZONED), *REF* 5-230
 - DATE, *REF* 5-230
- DATATRIEVE
 - components, *UG* 1-4 to 1-7
 - concepts and terms, *UG* 1-1 to 1-4
- DATATRIEVE Access Block (DAB), *DOCSUP*
 - 3-1, 3-3, *CALL* 4-1, 4-10t, 8-2, 8-2t
 - declaring, *CALL* 4-9
- DATATRIEVE states, *CALL* 4-1, 4-8 to 4-9,
 - 8-3t
 - list of, *CALL* 4-8
- DATATRIEVE-11, components of, *CALL* 1-1 to
 - 1-2
- Date
 - arithmetic, *REF* 2-9
 - edit string, *REF* 2-9
- DATE data type, *REF* 5-230
- Date fields, *UG* 5-10
 - edit string characters, *REF* 5-111
- Dates
 - suppressing, *REPORT* 2-10
- DB (debit)
 - edit string character, *REF* 5-109
- DDMF
 - See Distributed Data Manipulation Facility*
 - See Distributed Server*
- DDMF\$LOG logical name, *DOCSUP* 3-6
- DDMF.TSK, *CALL* 1-1
- Decimal point (.)
 - edit string character, *REF* 5-110
- DECLARE PORT statement, *REF* 5-41 to
 - 5-42, *CALL* 4-25
- DECLARE statement, *REF* 5-38 to 5-40, *UG*
 - 1-3
 - in BEGIN-END block, *REF* 5-31, 5-39
 - variable-name, *UG* 11-1
- Declaring variables, *UG* 11-1
- DECnet
 - default accounts, *CALL* 2-1
 - node specification, *CALL* 2-1, 8-17
 - using the Remote Call Interface, *CALL* 1-4
 - using the Remote Terminal Interface, *CALL*
 - 1-3
- DECREASING
 - in SORT statement, *REF* 5-213, 5-223
 - sort key, *REF* 3-7
- Default dictionary
 - QUERY.DIC, *UG* 21-1
- Defaults
 - data alignment, *DOCSUP* 1-6
 - file type, *DOCSUP* 2-3
- DEFINE command, *UG* 1-3
 - used in data dictionary, *UG* 21-3
- DEFINE DICTIONARY command, *INTRO*
 - 3-9, *REF* 5-43 to 5-44, *UG* 3-2 to 3-3
- DEFINE DOMAIN command, *INTRO* 3-9,
 - REF* 5-45 to 5-49, *DOCSUP* 2-3, 2-4, *UG*
 - 1-2, 4-1 to 4-2
 - entered interactively, *UG* 5-2
 - for RMS domains, *REF* 5-45 to 5-47
 - for view domains, *REF* 5-47 to 5-49
 - optional password, *UG* 4-2
 - syntax, *UG* 4-2
 - terminated by semicolon, *UG* 4-2
 - usage rules, *UG* 4-2
- DEFINE FILE command, *INTRO* 3-9, *REF*
 - 5-50 to 5-54, 5-173, *UG* 1-2, 6-3 to 6-8
 - optional clauses, *UG* 6-6 to 6-8
 - syntax, *UG* 6-3
 - used with sequential files, *UG* 6-4
- DEFINE PORT command, *REF* 5-55 to 5-56,
 - CALL* 4-25
- DEFINE PROCEDURE command, *REPORT*
 - 1-6, *REF* 5-57 to 5-59, *UG* 1-4, 9-2
- DEFINE RECORD command, *INTRO* 3-9,
 - REF* 4-1, 5-60 to 5-62, 5-173
 - advantage over ADT, *UG* 5-1
- DEFINE TABLE command, *REF* 5-63 to 5-65,
 - UG* 13-2
- DEFINEP command, *REF* 5-66 to 5-68, *UG* 1-3
- Defining
 - alternate keys, *UG* 6-5
 - data files, *REF* 5-50 to 5-54
 - data outside the Application Design Tool,
 - INTRO* 3-9 to 3-10
 - data with Application Design Tool, *INTRO*
 - 3-1 to 3-9
 - dictionaries, *REF* 5-43 to 5-44
 - domains, *REF* 1-10, 5-45 to 5-49, *UG* 1-2, 4-1
 - to 4-3
 - files, *REF* 1-10
 - global variables, *REF* 2-8
 - local variables, *REF* 2-8
 - ports, *REF* 5-55 to 5-56
 - privileges, *REF* 5-66 to 5-68
 - procedures, *REF* 1-7, 5-57 to 5-59
 - records, *REF* 1-10, 4-1 to 4-9, 5-60 to 5-62,
 - UG* 5-1 to 5-18
 - tables, *REF* 1-11, 5-63 to 5-65
- Definition prompt, *REF* 1-8

DELETE command, *REF* 5-69 to 5-70, *UG* 1-3, 17-5 to 17-6
 removing data dictionary definition, *UG* 21-3
 terminated by semicolon, *UG* 21-3
 used in data dictionary, *UG* 21-3

DELETE command (Editor), *REF* 5-86 to 5-88

DELETE option
 with FILE command, *DOCSUP* 2-5

DELETEP command, *REF* 5-71 to 5-72, *UG* 1-3

Deleting
 dictionaries, *REF* 5-69
 dictionary objects, *REF* 5-69 to 5-70
 domains, *REF* 5-69
 files with SUPERSEDE clause, *REF* 5-50
 procedures, *REF* 5-69
 records, *REF* 5-69
 records in indexed files, *REF* 5-53
 tables, *REF* 5-69

DESCENDING
 in SORT statement, *REF* 5-213, 5-223
 sort key, *INTRO* 6-8, *REF* 3-7

Detail lines, *REPORT* 2-10 to 2-17
 content, *REPORT* 2-11 to 2-12
 format, *REPORT* 2-12 to 2-15

DFN> prompt, *REF* 1-8, *UG* 2-4

DIC file type, *DOCSUP* 2-3

DICTIONARY
 in SHOW command, *REF* 5-206

Dictionary
See Data dictionary

Dictionary files, *DOCSUP* 2-3, 2-4
 compressing, *DOCSUP* 2-1, 2-6
 default, *DOCSUP* 2-3
 logical name for, *DOCSUP* 2-4

Dictionary objects
 controlling access to, *UG* 20-1 to 20-11
 editing, *REF* 1-11
 extracting definitions, *REF* 5-117
 extracting with QXTR, *REF* 5-119

Dictionary tables
See Tables

Directories, *DOCSUP* 2-2
 application, *DOCSUP* 2-3
 for PRO/DATATRIEVE, *DOCSUP* 2-4

Disk space
 conserving with QCPRS utility, *UG* 21-4 to 21-5

DISPLAY statement, *REF* 5-73 to 5-74
 with FILLER fields, *REF* 4-5

Displaying
 DATATRIEVE objects, *INTRO* 2-4
 dictionary name, *INTRO* 2-2

Displaying (cont.)
 established collections, *UG* 21-2
 readied domains, *UG* 21-2

Displaying reports, *REPORT* 1-4

Displaying variables, *REF* 5-40

Distributed Data Manipulation Facility
See Distributed Server

Distributed Data Manipulation Facility (DDMF)
 creating log files, *DOCSUP* 3-6
 on remote nodes, *DOCSUP* 1-4
 on the Professional, *DOCSUP* 1-2

Distributed Server, *REF* 1-12, *CALL* 1-1, 1-2, 1-5f
 function and use of, *UG* 1-5 to 1-7
 using with the Remote Call Interface, *CALL* 1-4

DO key, *DOCSUP* 1-6

Dollar sign (\$) edit string character, *REF* 5-110
 floating edit string character, *REF* 5-111

Domains
 access all records, *UG* 7-2
 accessing, *REF* 5-169 to 5-174
 alternative names, *REF* 5-169
 as record source, *REF* 3-2
 copying, *CALL* 2-3 to 2-4
 defining, *REF* 1-10, 5-45 to 5-49, *UG* 1-2, 3-1, 4-1 to 4-3
 deleting, *REF* 5-69
 editing, *REF* 5-80 to 5-85
 hierarchical, *REF* 5-148 to 5-151
 modifying, *REF* 5-46
 planning, *INTRO* 3-2 to 3-4
 restructuring, *UG* 16-1 to 16-9
 rules for naming, *UG* 4-1
 sample, *INTRO* 1-3, *UG* 2-2
 showing, *REF* 5-207
 views, *UG* 14-1 to 14-7

DOUBLE data type
See COMP-2 data type

DROP statement, *REF* 5-75 to 5-79, *UG* 1-3

DTCLIB.OLB
 Call Interface Library, *DOCSUP* 1-1, 1-3, 3-4
 call interface modules, *DOCSUP* 3-4
 language modules, *DOCSUP* 3-4
 specifying modules in, *DOCSUP* 3-5

DTCLIB.OLB library, *CALL* 1-2, 1-4

DTCLIB.OLB, DATATRIEVE object module library, *CALL* 3-7
 specifying modules in, *CALL* 3-7t

DTCMD routine, *CALL* 8-9
 format, *CALL* 4-14
 substituting variables with, *CALL* 4-15

DTCONT routine, *CALL* 4-27, 4-28, 8-11
DTFINI routine, *CALL* 4-34, 8-13
DTGETP routine, *CALL* 4-26, 8-14
DTINIT routine, *CALL* 4-11, 4-13, 8-16
 options, *CALL* 8-17t
DTLINE routine, *CALL* 4-17, 8-20
DTMSG routine, *CALL* 4-17, 8-21
DTPEOF routine, *CALL* 4-28, 8-23
DTPUTP routine, *CALL* 4-28, 8-25
DTPVAL routine, *CALL* 4-22, 8-27
DTR\$DIC logical name, *DOCSUP* 2-4, 2-5
DTR\$INI logical name, *DOCSUP* 2-4
DTR.TSK, *UG* 1-5, *CALL* 1-2
DTR> prompt, *REF* 1-8, *UG* 2-4
DTUNWD routine, *CALL* 4-34, 8-29
DUP, *UG* 6-8
 in DEFINE FILE command, *REF* 5-51

E

EDIT command, *REF* 5-80 to 5-85, *UG* 1-3, 10-2
Edit string characters, *REF* 5-102t
 alphanumeric insertion, *REF* 5-105
 alphanumeric replacement, *REF* 5-104
 asterisk (*), *REF* 5-107
 B (space), *REF* 5-105
 comma (,), *REF* 5-110
 CR (credit), *REF* 5-109
 date fields, *REF* 5-111 to 5-113
 DB (debit), *REF* 5-109
 decimal point (.), *REF* 5-110
 dollar sign (\$), *REF* 5-110
 floating, *REF* 5-111
 floating dollar sign, *REF* 5-111
 floating minus sign, *REF* 5-111
 floating plus sign, *REF* 5-111
 for alphanumeric fields, *REF* 5-104 to 5-106
 hyphen (-), *REF* 5-105
 J (Julian date), *REF* 5-102
 M (month letter), *REF* 5-102
 minus sign (-), *REF* 5-109
 N (month number), *REF* 5-102
 9 (numeric), *REF* 5-107
 numeric fields, *REF* 5-107 to 5-113
 numeric insertion, *REF* 5-109
 numeric replacement, *REF* 5-107
 percent sign (%), *REF* 5-110
 plus sign (+), *REF* 5-109
 slash (/), *REF* 5-105, 5-110
 T (text), *REF* 5-105
 W (day letter), *REF* 5-102
 X (alphanumeric), *REF* 5-104

Edit string characters, (cont.)
 Y (year), *REF* 5-102
 Z, *REF* 5-107
 zero (0), *REF* 5-110
Edit strings, *REPORT* 2-14 to 2-15
 date, *REF* 2-9
 for concatenated expressions, *REF* 2-16
 in PRINT statement, *REF* 5-161
 in RW PRINT statement, *REF* 5-193
 text edit strings, *REPORT* 2-15
EDIT_STRING clause, *REF* 5-101 to 5-113
 in field definition, *REF* 4-7, *UG* 5-4
Editing
 overlay descriptor files, *DOCSUP* 3-2, 3-4 to 3-5
 task builder command files, *DOCSUP* 3-2 to 3-4
Editing dictionary objects, *REF* 1-11, 5-80 to 5-85, 5-118
Editor, *UG* 17-1 to 17-14
 changing record definitions, *UG* 5-2
 command mode, *REF* 5-82
 commands, *UG* 17-4 to 17-13
 DELETE command, *UG* 17-5 to 17-6
 editing procedures, *UG* 9-13
 ending an editing session, *REF* 5-80
 EXIT command, *UG* 17-6
 IN> prompt, *REF* 1-8, 5-82, 5-90
 INSERT command, *UG* 17-6 to 17-8
 insert mode, *REF* 5-82
 invoking, *UG* 17-1 to 17-2
 line pointer, *UG* 17-2
 modes, *UG* 17-2
 prompts, *REF* 1-11
 QED> prompt, *REF* 1-8, 5-80, *UG* 17-3 to 17-14
 QUIT command, *UG* 17-8
 range specifiers, *UG* 17-3 to 17-4
 REPLACE command, *UG* 17-8 to 17-10
 sample editing session, *UG* 17-13 to 17-14
 SUBSTITUTE command, *UG* 17-10 to 17-11
 summary, *REF* 5-81t
 TYPE command, *UG* 17-11 to 17-13
 use of, *REF* 1-11
 using edited definitions, *REF* 5-84
Editor commands, *REF* 5-80 to 5-100
 DELETE, *REF* 5-86 to 5-88
 EXIT, *REF* 5-88 to 5-89
 INSERT, *REF* 5-90 to 5-92
 QUIT, *REF* 5-93 to 5-94
 range specifiers, *REF* 5-83t
 REPLACE, *REF* 5-94 to 5-96
 SUBSTITUTE, *REF* 5-96 to 5-98
 TYPE, *REF* 5-98 to 5-100

Elementary fields, *REPORT* 2-11
 alphanumeric, *REF* 4-6
 assigning values to, *REF* 5-23 to 5-25
 COMPUTED BY, *REF* 4-6
 DATE, *REF* 4-6
 defined, *REF* 1-9, *UG* 5-4 to 5-5
 in field definition, *REF* 4-2
 names, *REF* 2-3
 numeric, *REF* 4-6
 ELSE clause, *UG* 13-3
 in DEFINE TABLE command, *REF* 5-63
 EMPLOYEE_REC
 valid field names, *UG* 5-6F
 End-of-file marker, passing, DTPEOF, *CALL*
 4-28, 8-23
 END_PROCEDURE clause, *REF* 5-57, *UG* 9-2
 END_REPORT statement (Report Writer),
REPORT 1-4, 1-5, 1-7, *REF* 5-191, *UG* 2-4
 END_TABLE clause, *UG* 13-3
 in DEFINE TABLE command, *REF* 5-64
 Ending a DATATRIVE session, *INTRO* 2-5
 Ending the DATATRIVE session, DTFINI,
CALL 4-34, 8-13
 EQUAL relational operator, *INTRO* 6-4, *REF*
 2-17, *UG* 7-4
 Equal sign
 in assignment statements, *REF* 5-23, 5-26,
 5-28
 ERASE statement, *REF* 5-114 to 5-115, *UG*
 1-3
 access mode required, *REF* 5-172
 restrictions, *UG* 6-2
 Erasing data, *INTRO* 8-1 to 8-3
 Error messages, *UG* 2-5 to 2-6
 incorrectly named fields, *UG* 5-6
 located by a procedure, *UG* 9-5 to 9-6
 Error messages, retrieving, DTMSG, *CALL*
 4-17, 8-21
 Error severity codes, DAB\$W_ERR_SEV,
CALL 4-18, 8-5
 Error status codes, DAB\$W_ERR_CODE,
CALL 4-18, 8-5
 Evaluating
 arithmetic expressions, *REF* 2-14
 Boolean expressions, *REF* 2-21, 2-21t
 Event Flag Numbers, *CALL* 3-7
 Exclamation point (!), *REF* 1-7
 in command files, *REF* 5-14
 EXCLUSIVE access option, *REF* 5-169
 EXIT command, *INTRO* 2-5, *REF* 5-116, *UG*
 1-3, 2-6, 17-6, *CALL* 2-2
 EXIT command (Editor), *REF* 5-88 to 5-89
 EXIT key, *DOCSUP* 1-7

Exiting
 an ADT session, *INTRO* 3-5
 DATATRIVE, *INTRO* 2-5
 Exiting DATATRIVE, *UG* 2-6
 Expressions
 arithmetic, *REF* 2-14 to 2-15
 Boolean, *REF* 2-1, 2-16 to 2-21
 concatenated, *REF* 2-2, 2-15 to 2-16
 DATE value, *REF* 2-9
 evaluating arithmetic, *REF* 2-14
 evaluating Boolean, *REF* 2-21, 2-21t
 prompting value, *REF* 2-10 to 2-11
 statistical, *REF* 2-11 to 2-13
 table values in, *REF* 2-11
 value, *REF* 2-1, 2-1 to 2-16
 EXTEND access mode, *REF* 5-170
 EXTRACT command, *REF* 5-117 to 5-120,
DOCSUP 2-3, *UG* 1-3, 9-13, 21-3, *CALL*
 2-4
 copying record definitions, *UG* 5-3

F

FAMILIES sample domain, *UG* 2-2
 using list field, *UG* 5-13
 using SHOWP, *UG* 21-2
 Fast Install, *DOCSUP* 2-5, 3-6
 FCTR line
 in ODL files, *DOCSUP* 3-5
 Field classes, *REF* 4-6, 4-6t
 Field definitions, *UG* 5-3 to 5-11
 clauses, *UG* 5-9 to 5-11
 clauses in, *REF* 4-7 to 4-9, 4-7t, *UG* 1-2
 COMP (INTEGER), *REF* 5-229
 COMP-1 (REAL), *REF* 5-229
 COMP-2 (DOUBLE), *REF* 5-229
 COMP-3 (PACKED), *REF* 5-229
 COMP-5 (ZONED), *REF* 5-230
 components of, *REF* 4-1
 EDIT_STRING clause, *UG* 5-4
 elementary fields, *UG* 5-4 to 5-5
 group fields, *UG* 5-4 to 5-5
 level numbers, *UG* 5-4 to 5-5
 level numbers in, *REF* 4-2 to 4-3
 naming fields, *UG* 5-6
 PICTURE clause, *UG* 5-4
 rules for writing, *REF* 4-8, *UG* 5-3 to 5-11
 specifying field characteristics, *REF* 4-7
 terminated by period (), *UG* 5-4
 valid field names, *UG* 5-6F
 word boundary alignment, *REF* 5-21
 Field names
 as value expressions, *REF* 2-3
 COMPUTED BY clause, *REF* 2-6

Field names (cont.)

duplicate, *UG* 5-7 to 5-8
elementary, *REF* 2-3
FILLER, *REF* 4-4 to 4-5, *UG* 5-8 to 5-9
group, *REF* 2-4 to 2-5
in AT BOTTOM statement (RW), *REF* 5-187
in AT TOP statement (RW), *REF* 5-189
in field definition, *REF* 4-1, 4-3 to 4-5
qualified, *REF* 2-3 to 2-7
qualifying, *REF* 3-6
REDEFINES, *REF* 2-3
restrictions, *UG* 5-6
rules for, *REF* 4-3
use of hyphens or underscores, *UG* 1-2
using as value expressions, *REF* 2-3
using query names, *REF* 2-6

Field values

in detail lines, *REPORT* 2-11 to 2-12

Fields

alphanumeric, *UG* 5-10
COMPUTED BY, *REF* 2-6, 5-23, 5-36 to 5-37, *UG* 5-11
data types, *INTRO* 3-2 to 3-3, *REF* 5-229 to 5-230
date, *REF* 5-230, *UG* 5-10
definition clauses, *UG* 5-9 to 5-11
elementary, *REF* 1-9, 2-3, 4-2
elementary and group, *UG* 1-2
formatting for output, *REF* 5-101 to 5-113
formatting storage, *REF* 5-155 to 5-157, 5-228 to 5-230
group, *REF* 1-9, 4-2
key, *REF* 5-51
KEY attributes, *REF* 5-52t
KEY defaults, *REF* 5-51t
level numbers of, *REF* 4-3
list, *UG* 5-13
masking, *REF* 4-4
multiple occurrences, *REF* 5-148 to 5-151
naming, *UG* 5-6
numeric, *UG* 5-10
query names, *INTRO* 3-2
redefining, *REF* 5-175 to 5-176
showing, *REF* 5-207
size, *INTRO* 3-3
specifying types of data, *UG* 5-9 to 5-11

FILE command line

/DELETE option, *DOCSUP* 2-5
/KEEP option, *DOCSUP* 2-5

File types

CMD, *DOCSUP* 3-2
default, *DOCSUP* 2-3
INS, *DOCSUP* 3-5
ODL, *DOCSUP* 3-2

File types (cont.)

of application files, *DOCSUP* 2-3
of installation files, *DOCSUP* 2-5

Files

allocating storage, *REF* 5-20 to 5-22
application, *DOCSUP* 2-1, 2-2
changing structure of, *UG* 16-1 to 16-9
closing log, *REF* 5-35, 5-153
comparison of sequential and indexed, *UG* 6-2T
compressing dictionary, *DOCSUP* 2-6
creating, *REF* 5-50 to 5-54
data, *DOCSUP* 2-2, 2-3
defining, *REF* 1-10, 5-50 to 5-54, *UG* 1-2, 6-1 to 6-8
dictionary, *DOCSUP* 2-4
format for installation, *DOCSUP* 2-5
indexed, *INTRO* 3-3 to 3-4, 7-7, *REF* 1-10, *UG* 6-1 to 6-8
initialization, *DOCSUP* 2-4
invoking command, *DOCSUP* 2-7
naming application, *DOCSUP* 2-2
naming installation, *DOCSUP* 2-5, 3-6
opening log, *REF* 5-152
organizing, *REF* 1-10
PRO/DATATRIEVE, *DOCSUP* 2-2
PRO/DATATRIEVE message, *DOCSUP* 2-5
PRO/DATATRIEVE task image, *DOCSUP* 2-5
sample CMD, *DOCSUP* 3-3
sample installation, *DOCSUP* 3-5
sequential, *INTRO* 3-3 to 3-4, *REF* 1-10, *UG* 6-1 to 6-7
with fixed-length records, *REF* 5-51
writing installation, *DOCSUP* 2-5 to 2-6, 3-5
FILLER fields, *REF* 4-4 to 4-5
accessing data in, *REF* 4-5
as group field name, *UG* 5-8
defining, *REF* 5-60
displaying with DISPLAY, *REF* 4-5
FIND statement, *INTRO* 6-11, *REPORT* 2-2, *REF* 5-121 to 5-122, *UG* 1-3
access mode required, *REF* 5-172
context variable in, *REF* 3-6
FINISH command, *INTRO* 6-21, *REPORT* 1-7, *REF* 5-123 to 5-125, *UG* 1-3
releasing domains, *REF* 5-173
FIRST clause
in SELECT, *REF* 5-197
FIRST n clause
in RSE, *REF* 3-4, *UG* 7-3
Flat records, *UG* 5-13F
FOR statement, *REPORT* 3-16, *REF* 5-126 to 5-129, *UG* 8-3

For statement. (cont.)
controlling PRINT output, *REF* 5-161
creating target record streams, *UG* 12-22 to 12-24
declaring variables in, *REF* 5-39
invoking procedures with, *REF* 5-10
to form record streams, *REF* 3-3
with STORE statement, *REF* 5-218

Form feed
trailing, *REPORT* 2-3

Formatting output
See also Edit string characters
date fields, *REF* 5-111
default, *REPORT* 2-2, 2-4
fields, *REF* 5-101 to 5-113
report header, *REPORT* 2-7
reports, *REPORT* 1-4
with PRINT, *REF* 5-158 to 5-165
with the Report Writer, *REF* 1-11, 5-193

FORTTRAN, *DOCSUP* 3-4

FROM clause, *UG* 14-2
in DEFINE DOMAIN command, *REF* 5-48

Function keys, *DOCSUP* 1-5, 1-6
CANCEL, *DOCSUP* 1-6
DO, *DOCSUP* 1-6
EXIT, *DOCSUP* 1-7
HELP, *DOCSUP* 1-6
INTERRUPT/DO, *DOCSUP* 1-7
MAIN SCREEN, *DOCSUP* 1-6

Functions
format of statistical, *REF* 2-12
statistical, *REF* 2-11 to 2-13
using statistical, *REF* 2-11
values derived with statistical, *REF* 2-11t

G

GBLDEF line
in CMD files, *DOCSUP* 3-3

GBLPAT line
in CMD files, *DOCSUP* 3-3

GE
See GREATER_EQUAL relational operator

Global variables, *UG* 11-4
named in context block, *UG* 12-4

GREATER_EQUAL relational operator,
INTRO 6-4, *REF* 2-17, *UG* 7-6

GREATER_THAN relational operator,
INTRO 6-4, *REF* 2-17, *UG* 7-6

Group fields, *REPORT* 2-11
alphanumeric, *REF* 4-6
as primary key, *UG* 6-4 to 6-5
assigning values to, *REF* 5-25 to 5-27

Group fields (cont.)
defined, *REF* 1-9, *UG* 5-4 to 5-5
in assignment statements, *REF* 2-4 to 2-5
in field definition, *REF* 4-2
qualifying, *REF* 2-6
using FILLER field, *UG* 5-8

GT

See GREATER_THAN relational operator

Guide Mode, *INTRO* 4-3, *UG* 2-3, 2-7 to 2-8
invoking, *UG* 2-8
using a question mark (?), *UG* 2-8

H

Halting execution

with ABORT, *REF* 5-15 to 5-18

HELP command, *INTRO* 4-1 to 4-2, *REF* 5-130
to 5-132, *UG* 1-3, 2-6 to 2-7

ADVANCED HELP, *UG* 2-7

HELP key, *DOCSUP* 1-6

Hierarchical domains, *REF* 5-148 to 5-151

Hierarchical records, *REPORT* 3-15 to 3-18

Hierarchies, *UG* 15-1 to 15-11

defining, *UG* 5-13 to 5-18

eliminating empty print lines, *UG* 15-8

retrieving values, *UG*

saving space, *UG* 5-13

using, *UG* 5-14 to 5-18

Hyphen (-), *REPORT* 2-15, 2-16

as a minus sign, *REF* 1-4

continuation character, *REF* 1-6, *UG* 5-6

conversion to underscore (_), *REF* 1-4, *UG*
1-2, 4-2

edit string character, *REF* 5-105

in record name, *UG* 5-3

in string literals, *REF* 2-2

I

IF-THEN-ELSE statement, *REF* 5-133 to
5-135, *UG* 8-5

using with ABORT statement, *REF* 5-16

IN relational operator, *REF* 2-17, 2-19, *UG*
13-5

IN> prompt, *REF* 1-8, 1-12, 5-82, 5-90

Inclusion files, DAB, *CALL* 4-11

INCREASING

in SORT statement, *REF* 5-213, 5-223

sort key, *REF* 3-7

Indexed files, *INTRO* 3-3 to 3-4, *UG* 6-1 to 6-8

compared with sequential, *UG* 6-2T

compressing, *UG* 21-6

- Indexed files, (cont.)
 - creating, *REF* 5-51, 5-52
 - defining, *UG* 6-4
 - deleting records, *REF* 5-53
 - ERASE statement with, *INTRO* 8-1 to 8-3
 - KEY field attributes, *REF* 5-52t
 - KEY field defaults, *REF* 5-51t
 - keys, *REF* 1-10
 - modifying, *INTRO* 7-6 to 7-8
 - modifying records in, *REF* 5-53
 - multikey, *UG* 6-3
 - optimizing storage, *UG* 21-6
- Information management, *INTRO* 1-1
- INI file type, *DOCSUP* 2-3
- Initialization file, *DOCSUP* 2-3
 - logical name for, *DOCSUP* 2-4
- Initialization files
 - Local Call Interface, *DOCSUP* 3-5
- Initializing
 - global variables, *REF* 5-38
 - local variables, *REF* 5-32, 5-39
 - variables, *REF* 2-7
- Initializing the Call Interface, *CALL* 4-13, 8-16
- Input line prompt, *UG* 2-4
- INS file type, *DOCSUP* 2-5, 3-5
- INSERT command, *UG* 17-6 to 17-8
- INSERT command (Editor), *REF* 5-90 to 5-92
- INSTALL command line
 - /TASK option, *DOCSUP* 2-5
- Installation kit, *UG* 2-2
- Installing
 - application files, *DOCSUP* 3-5
 - PROCALDTR.TSK, *DOCSUP* 3-5
- INTEGER data type
 - See COMP data type*
- Interactive DATATRIEVE, *UG* 1-5, *CALL* 1-2
- Interactive PRO/DATATRIEVE, *DOCSUP* 1-1, 1-2
- Internal storage, *REF* 5-228 to 5-230
- INTERRUPT/DO keys, *DOCSUP* 1-7
- @ (Invoke Command File) Command
 - See AT*
- : (Invoke procedure)
 - See Colon*
- Invoking
 - an ADT session, *INTRO* 3-6
 - command files, *REF* 5-13 to 5-14
 - DATATRIEVE, *INTRO* 2-1, *UG* 2-1 to 2-2
 - procedures, *REPORT* 1-7, *REF* 1-7, 5-10 to 5-12, 5-58
 - tables, *REF* 5-64

J

- J (Julian date)
 - edit string character, *REF* 5-102

K

- KEEP option
 - with FILE command, *DOCSUP* 2-5
- KEY clause, *UG* 6-4
 - in DEFINE FILE command, *REF* 5-51
- KEY field
 - attributes, *REF* 5-52t
 - defaults, *REF* 5-51t
 - duplicate values in, *REF* 5-51
 - modifying, *REF* 5-51
 - primary, *REF* 5-51
- Keys
 - accessing dictionary objects, *UG* 20-2
 - alternate index, *INTRO* 3-4
 - defining alternate keys, *UG* 6-5
 - defining key fields, *UG*
 - duplicate values, *INTRO* 3-4
 - primary index, *INTRO* 3-3
- Keywords, *REF* 1-5

L

- LAST
 - in SELECT, *REF* 5-197
- LCPOS object module, *DOCSUP* 1-3, 1-7, 3-4
- LE
 - See LESS_EQUAL relational operator*
- LEFT_RIGHT allocation, *REF* 5-20, *CALL* 2-4
- LEFT_RIGHT data alignment, *DOCSUP* 1-6
- LESS_EQUAL relational operator, *INTRO* 6-4, *REF* 2-17, *UG* 7-6
- LESS_THAN relational operator, *INTRO* 6-4, *REF* 2-17, *UG* 7-6
- Letters
 - in string literals, *REF* 2-3
 - in table strings, *REF* 5-63
 - treatment of, *REF* 1-4
 - treatment with CONTAINING, *REF* 2-18
- Level numbers
 - in field definition, *REF* 4-1, 4-2 to 4-3
 - rules for assigning, *REF* 4-3
 - valid values, *REF* 5-61
- Line pointer, *UG* 17-2
- Linking
 - programs, *DOCSUP* 3-5
 - with Call Interface modules, *DOCSUP* 1-3
 - with LCPOS, *DOCSUP* 1-3
 - with NCPOS, *DOCSUP* 1-3
- Lists, *REF* 2-19
 - as record source, *REF* 3-3

Lists, (cont.)

- changing length of, *UG* 5-17 to 5-18
- creating with OCCURS FOR, *REF* 5-48
- defining fixed occurrences, *UG* 5-14 to 5-15
- defining variable occurrences, *UG* 5-15 to 5-17
- defining with OCCURS clause, *REF* 5-148 to 5-151, *UG* 5-13 to 5-18
- modifying, *REF* 3-4
- nesting to form sublists, *UG* 5-17
- printing, *REPORT* 3-17 to 3-18

Lists, in records

See Hierarchies

Literals, *REPORT* 2-12, *REF* 2-2 to 2-3

- character string, *REF* 2-2 to 2-3
- hyphens in, *REF* 2-2
- in detail lines, *REPORT* 2-11
- numeric, *REF* 2-3
- printing, *REF* 5-159

Local Call Interface, *DOCSUP* 1-1, 1-2

- initializing, *DOCSUP* 3-5
- installation file for, *DOCSUP* 3-5
- LCPOS module, *DOCSUP* 1-3, 1-7, 3-4
- requirements for using, *DOCSUP* 1-3
- using, *DOCSUP* 1-4

Local variables, *UG* 11-4 to 11-5

- effect on context stack, *UG* 12-7

Lock types, *UG* 20-2

Log files

- closing, *REF* 5-153
- DDMF, *DOCSUP* 3-6
- opening, *REF* 5-152
- PROCALDTR, *DOCSUP* 3-6

Logical names, *DOCSUP* 1-2, 1-5, 1-6, 2-1, 2-3 to 2-4

- APPL\$DIR, *DOCSUP* 2-3, 2-4
- assigning, *DOCSUP* 2-4
- CALDTR\$INI, *DOCSUP* 3-5
- CALDTR\$LOG, *DOCSUP* 3-6
- DDMF\$LOG, *DOCSUP* 3-6
- DTR\$DIC, *DOCSUP* 2-4, 2-5
- DTR\$INI, *DOCSUP* 2-4
- in DEFINE DOMAIN command, *DOCSUP* 2-4
- rules for using, *DOCSUP* 2-3
- using, *DOCSUP* 2-4

Logical Unit Numbers

- allocating, *DOCSUP* 3-2, *CALL* 3-7, 3-8f

Logical unit numbers

- allocating, *DOCSUP* 3-2

Loops

- ABORT statement in, *REF* 5-15
- FOR statement, *REF* 1-3, 5-126 to 5-129

Loops, (cont.)

- REPEAT statement, *REF* 1-3, 5-181
- WHILE statement, *REF* 1-3, 5-233 to 5-234

Lowercase letters

- conversion of, *REF* 1-4
- in string literals, *REF* 2-3

LT

See LESS_THAN relational operator

LUN

See Logical Unit Numbers

LUNMAP

storage area, *DOCSUP* 3-3, 3-4

LUNMAP, area for LUN specification, *CALL* 3-7

LUNs

See Logical Unit Numbers

M

M (month letter)

- edit string character, *REF* 5-102
- MACRO-11 assembly language, *CALL* 8-8
- MAIN SCREEN key, *DOCSUP* 1-6
- MAJOR_MINOR allocation, *REF* 5-20, *CALL* 2-4
- MAJOR_MINOR data alignment, *DOCSUP* 1-6
- MAX clause, *UG* 5-17, 6-7
- in DEFINE FILE command, *REF* 5-51
- MAX statistical function, *REPORT* 2-18 to 2-19, *REF* 2-11
- in AT BOTTOM statement (RW), *REF* 5-187
- in AT TOP statement (RW), *REF* 5-189

Memory, conserving

See Optimizing workspace

Menu interface, example, *CALL* 5-1

Menus

creating with the Call Interface, *CALL* 1-5

Messages, obtaining, *DTMSG*, *CALL* 4-17, 8-21

MIN statistical function, *REPORT* 2-18 to 2-19, *REF* 2-11

in AT BOTTOM statement (RW), *REF* 5-187

in AT TOP statement (RW), *REF* 5-189

Minus sign (–)

- edit string character, *REF* 5-109
- floating edit string character, *REF* 5-111
- in arithmetic expressions, *REF* 2-14

MODIFY access mode, *REF* 5-170

MODIFY statement, *REF* 5-136 to 5-147, *UG* 1-3, 2-5

access mode required, *REF* 5-172

ALL, *INTRO* 7-8 to 7-9

assigning field values with, *REF* 5-23, 5-26

MODIFY statement (cont.)
 changing fields, *UG* 6-2
 effect of ABORT statement, *REF* 5-16
 preventing, *INTRO* 7-5 to 7-6
 VERIFY clause, *REF* 5-217
 with lists, *REF* 3-4

Modifying
 access control list, *REF* 5-66 to 5-68
 data, *INTRO* 7-1 to 7-10
 domains, *REF* 5-46
 key fields, *REF* 5-51
 procedures, *REF* 5-58
 record definitions, *REF* 5-62
 records in indexed files, *REF* 5-53
 records in sequential files, *REF* 5-53
 selected record, *INTRO* 7-4 to 7-6
 tables, *REF* 5-65

N

N (month number)
 edit string character, *REF* 5-102

NAME command line, *DOCSUP* 2-5

Names, *REF* 1-5 to 1-6
 APPL\$DIR logical, *DOCSUP* 2-3
 CALDTR\$INI logical, *DOCSUP* 3-5
 CALDTR\$LOG logical, *DOCSUP* 3-6
 continuing, *REF* 1-6
 DDMF\$LOG logical, *DOCSUP* 3-6
 DTR\$DIC logical, *DOCSUP* 2-4
 DTR\$INI logical, *DOCSUP* 2-4
 logical, *DOCSUP* 2-1, 2-3 to 2-4
 PRO/DATATRIEVE file, *DOCSUP* 2-2
 reserved, *DOCSUP* 2-2
 rules for, *REF* 1-5

Naming
 application files, *DOCSUP* 2-2
 application installation files, *DOCSUP* 2-5,
 3-6
 directories, *DOCSUP* 2-2
 volumes, *DOCSUP* 2-2

Naming fields
 using the ADT command, *INTRO* 3-2

Naming reports, *REPORT* 1-4

NC11M
 object module, *DOCSUP* 1-7

NCPOS object module, *DOCSUP* 1-3, 1-7, 3-4

NCRSTS
 object module, *DOCSUP* 1-7

NE
See NOT_EQUAL relational operator

NEW_PAGE, *REPORT* 3-10, 3-13
 in AT BOTTOM statement (RW), *REF* 5-187

NEW_PAGE, (cont.)
 in AT TOP statement (RW), *REF* 5-189
 print list element, *REF* 5-160
 RW print list element, *REF* 5-193

NEW_SECTION
 in AT BOTTOM statement (RW), *REF* 5-187
 in AT TOP statement (RW), *REF* 5-189
 RW print list element, *REF* 5-193

NEXT
 in SELECT, *REF* 5-197, 5-198

9 (numeric)
 edit string character, *REF* 5-107
 picture string character, *REF* 5-156

NO CHANGE, *UG* 6-7
 in DEFINE FILE command, *REF* 5-51

NO DUP, *UG* 6-8
 in DEFINE FILE command, *REF* 5-51

Node specification, *CALL* 2-1, 8-17

NOT BETWEEN relational operator, *REF*
 2-17

NOT Boolean operator, *INTRO* 6-5, *REF* 2-20,
UG 7-9

NOT BT
See NOT BETWEEN relational operator

NOT CONT
See NOT CONTAINING relational operator

NOT CONTAINING relational operator, *REF*
 2-17

NOT EQUAL relational operator, *UG* 7-4

NOT IN relational operator, *REF* 2-17, *UG*
 13-5

NOT_EQUAL relational operator, *INTRO* 6-4,
REF 2-17

Numeric fields, *UG* 5-10
 edit string characters, *REF* 5-107 to 5-113
 picture string characters, *REF* 5-156

Numeric literals
See Literals

O

Object module libraries, *CALL* 3-7

Obtaining messages, DTMSG, *CALL* 4-17,
 8-21

Obtaining print lines, DTLINE, *CALL* 4-17,
 8-20

Obtaining records from DATATRIEVE,
 DTGETP, *CALL* 4-26, 8-14

OCCURS clause, *REPORT* 3-15, *REF* 5-148 to
 5-151, *UG* 5-13 to 5-18
 defining hierarchical records, *UG* 5-13
 fixed number of occurrences, *REF* 5-148 to
 5-149, *UG* 5-14 to 5-15
 in field definition, *REF* 4-7

OCCURS clause, (cont.)
 variable number of occurrences, *REF* 5-150
 to 5-151, *UG* 5-15 to 5-17

OCCURS DEPENDING clause
 changing list length, *UG* 5-17

OCCURS FOR clause, *UG* 14-2
 in DEFINE DOMAIN command, *REF* 5-48

ODL file type, *DOCSUP* 3-2

OF rse clause
 targeting record streams, *UG* 12-21 to 12-22

OPEN command, *REF* 5-152 to 5-154

Operating systems
 for DATATRIEVE, *UG* 1-1

Operators
 arithmetic, *REF* 2-14, 2-14t
 Boolean, *REF* 2-20 to 2-21
 relational, *REF* 2-17 to 2-20, 2-20t

Optimizing
 response time, *UG* 18-5 to 18-8
 SHOW SPACE command, *REF* 5-207
 with keyed access, *UG* 18-5 to 18-8
 workspace, *UG* 18-4 to 18-5

Options, for DTINIT, *CALL* 8-16, 8-17t

OR Boolean operator, *INTRO* 6-5, *REF* 2-20,
UG 7-9

Output
 controlling, *UG* 19-1 to 19-5
 default settings, *UG* 19-1

Overlay descriptor files
 creating, *DOCSUP* 3-2
 editing, *DOCSUP* 3-2, 3-4 to 3-5
 sample, *DOCSUP* 3-5

Overlays, *CALL* 3-9

OWNERS sample domain, *UG* 2-2

P

P (decimal scaling)
 picture string character, *REF* 5-157

PAB
See Professional Application Builder

PACKED data type
See COMP-3 data type

Page format, *REPORT* 2-4 to 2-7
 length, *REPORT* 2-6
 width, *REPORT* 1-6, 2-2, 2-5

Page numbers, *REPORT* 2-7
 suppressing, *REPORT* 2-10

Parentheses
 in arithmetic expressions, *REF* 2-14
 in Boolean expressions, *REF* 2-21
 in statistical expressions, *REF* 2-12

PASCAL, *DOCSUP* 3-4

Passing command lines to DATATRIEVE,
 DTCMD, *CALL* 4-14, 8-9

Passing end-of-file marker, DTPEOF, *CALL*
 4-28, 8-23

Passing records from DATATRIEVE, DTGETP,
CALL 8-14

Passing records to DATATRIEVE, DTPUTP,
CALL 4-28, 8-25

Passing values to DATATRIEVE, DTPVAL,
CALL 4-22, 8-27

Passwords
 keys, *UG* 20-2
 specifying to the Remote Terminal Interface,
CALL 2-1

Percent sign (%)
 edit string character, *REF* 5-110

Period
 in DECLARE statement, *REF* 5-38
 in DEFINE DOMAIN clause, *REF* 5-48
 in DEFINE RECORD command, *REF* 5-61
 in field definition, *REF* 4-1, *UG* 5-4

PERSONNEL sample domain, *UG* 2-2, 2-3
 record data items, *UG* 5-2F

PERSONNEL_REC
 record level numbers, *UG* 5-4
 sample record definition, *UG* 5-2F

PICTURE clause, *REF* 5-155 to 5-157, *UG*
 5-10
 alphanumeric fields, *REF* 5-156
 in DECLARE statement, *REF* 5-38
 in field definition, *REF* 4-7, *UG* 5-4
 numeric fields, *REF* 5-156
 restrictions, *REF* 2-3

Picture string characters
 9 (numeric), *REF* 5-156
 P (decimal scaling), *REF* 5-157
 S (sign), *REF* 5-157
 V (decimal point), *REF* 5-157
 X (alphanumeric), *REF* 5-156

Plus sign (+)
 edit string character, *REF* 5-109
 floating edit string character, *REF* 5-111

Pool space
See Workspace

Ports, *CALL* 4-24f, 8-14
 declaring temporary, *REF* 5-41 to 5-42
 defining, *REF* 5-55 to 5-56
 definition of, *CALL* 4-24
 global, *REF* 5-42
 local, *REF* 5-42

Primary index key, *INTRO* 3-3

Print items
 column position of, *REPORT* 2-13 to 2-14
 cross tabulations, *REPORT* 3-14
 edit string format, *REPORT* 2-14 to 2-15
 order of, *REPORT* 2-13

Print lines, obtaining, *DTLINE*, *CALL* 4-17, 8-20

PRINT statement, *INTRO* 6-1, *REF* 5-158 to 5-165, *UG* 1-3
 access mode required, *REF* 5-172
 creating reports with, *INTRO* 9-1
 defaults, *REF* 5-162
 displaying data with, *INTRO* 6-5 to 6-10
 lists, *UG* 15-7
 print list elements, *REF* 5-160t
 print list modifiers, *REF* 5-161t
 retrieving data, *UG* 2-3 to 2-4

PRINT statement (Report Writer), *REPORT* 1-3, 1-3, 1-4, 2-1, *REF* 5-192 to 5-194
 detail lines, *REPORT* 2-11
 order of print items, *REPORT* 2-13

Privileges
 access codes, *REF* 5-67t
 assigning, *UG* 20-10
 defining with *DEFINER*, *REF* 5-66 to 5-68
 showing, *REF* 5-209 to 5-210

PRO/DATATRIEVE
 file names, *DOCSUP* 2-2
 message file, *DOCSUP* 2-5
 task image, *DOCSUP* 2-5

PRO/DDMF
 Distributed Data Manipulation Facility, *DOCSUP* 1-2

PRO/DECnet
 on a Professional, *DOCSUP* 1-2, 1-3

PRO/RDT
 Remote Data Transfer, *DOCSUP* 1-1, 1-2

PROCALDTR.TSK
 Callable PRO/DATATRIEVE, *DOCSUP* 1-1, 1-4

Procedures, *UG* 1-4, 9-1 to 9-15
 aborting, *UG* 9-12
 comments in, *UG* 9-5
 compared with command files, *UG* 10-1
 contents, *UG* 3-1, 9-3 to 9-5
 defining, *REF* 1-7, 5-57 to 5-59, *UG* 9-1 to 9-2
 deleting, *REF* 5-69, *UG* 9-14 to 9-15
 displaying, *UG* 9-13
 editing, *REF* 5-80 to 5-85, *UG* 9-13
 for reports, *REPORT* 2-19
 in compound statements, *UG* 9-10 to 9-12

Procedures (cont.)
 including command files in, *REF* 5-13
 invoking, *REF* 1-7, 5-10 to 5-12, 5-58, *UG* 9-2 to 9-3
 locating errors, *UG* 9-5 to 9-6
 maintaining, *UG* 9-13 to 9-15
 modifying, *REF* 5-58
 nesting, *REF* 5-10, *UG* 9-8 to 9-9
 REPEAT statements in, *REF* 5-182
 samples, *UG* 9-6 to 9-8
 SET ABORT/SET NO ABORT, *UG* 9-12
 showing, *REF* 5-207
 stopping execution of, *REF* 5-15
 using with the Call Interface, *CALL* 1-5

PRODTR.DIC
 dictionary file, *DOCSUP* 2-4

PRODTR.TSK
 interactive PRO/DATATRIEVE, *DOCSUP* 1-1, 1-2, 2-5

Professional Application Builder (PAB), *DOCSUP* 3-2

Prompting expressions, *CALL* 4-22

Prompting value expressions, *REPORT* 1-6 to 1-7, 2-12, *REF* 2-10 to 2-11
 as print item, *REPORT* 2-11
 for page width, *REPORT* 2-5
 for report length, *REPORT* 2-6
 for storing and modifying values, *REF* 1-9, *UG* 2-5
 for syntax, *REF* 1-9
 forming, *REF* 2-10
 in Boolean expressions, *REF* 2-18
 in loops, *REF* 2-10

Prompts, *REF* 1-8 to 1-9
 CON>, *REF* 1-8, *UG* 2-4
 DFN>, *REF* 1-8, *UG* 2-4
 DTR>, *REF* 1-8, *UG* 2-4
 IN>, *REF* 1-8, 1-12, 5-82, 5-90
 QED>, *REF* 1-8, 1-11, 5-80
 REMDTR>, *REF* 1-12
 RW>, *REPORT* 1-5, *REF* 1-8, *UG* 2-4
 syntax, *UG* 2-5

PROTECTED access option, *REF* 5-169

Protecting dictionary tables, *UG* 13-9

Q

QCPRS utility
 conserving disk space, *UG* 21-4 to 21-5
 data dictionary compression, *UG* 21-4
 defaults, *UG* 21-5
 improving performance, *UG* 21-4
 invoked by Digital Command Language, *UG* 21-4

QED> prompt, *REF* 1-8, 1-11, 5-80, *UG* 17-3 to 17-14

Qualified field names, *REF* 2-6

Query names, *INTRO* 3-2
specifying, *UG* 5-11 to 5-12
using, *REF* 2-6

QUERY.DIC default dictionary, *UG* 21-1

QUERY.INI file, *UG* 2-2
sample, *UG* 2-3
SET DICTIONARY in, *UG* 2-3
SET GUIDE in, *UG* 2-3

QUERY_HEADER clause, *REF* 5-166 to 5-167, *UG* 5-9
in field definition, *REF* 4-4, 4-7

QUERY_NAME clause, *REF* 5-168, *UG* 5-9
in field definition, *REF* 4-4, 4-7

Question mark (?)
used in Guide Mode, *UG* 2-8

QUIT command, *UG* 17-8

QUIT command (Editor), *REF* 5-93 to 5-94

Quotation marks
around string literals, *REF* 2-2
in string literals, *REF* 2-2

QXTR utility, *REF* 5-119, *UG* 21-6 to 21-7
creating command files, *UG* 21-6
transferring data dictionary objects, *UG* 21-6

R

Range specifiers
in editing commands, *REF* 5-83t

READ access mode, *REF* 5-170

READY
in SHOW command, *REF* 5-207

READY command, *INTRO* 2-4 to 2-5,
REPORT 2-2, *REF* 5-169 to 5-174
gaining access to domains, *UG* 1-3
retrieving data, *UG* 2-3 to 2-4
with PRINT statement, *REF* 5-159

REAL data type
See COMP-1 data type

Record definition
changing, *UG* 16-1 to 16-9
contents, *UG* 3-1
field definitions within, *UG* 5-3 to 5-11
modifying, *REF* 5-62
OCCURS clause, *REF* 5-148 to 5-151
PICTURE clause, *REF* 5-155 to 5-157
QUERY_HEADER clause, *REF* 5-166 to 5-167
QUERY_NAME clause, *REF* 5-168
REDEFINES clause, *REF* 5-175 to 5-176

Record definition (cont.)
SIGN clause, *REF* 5-211 to 5-212
steps in writing, *UG* 5-1 to 5-18
using OCCURS clause, *UG* 5-13
VALID IF clause, *REF* 5-231 to 5-232, *UG* 12-8

Record names
qualifying, *REF* 2-6

Record selection expression, *INTRO* 6-2 to 6-5,
REF 2-22, *UG* 7-1 to 7-10
accessing records, *UG* 7-2
ALL clause, *REF* 3-4, *UG* 7-2
Boolean expressions, *UG* 7-4 to 7-6
context variable in, *REF* 3-6
definition, *UG* 7-1
FIRST n clause, *REF* 3-4, *UG* 7-1 to 7-3
format, *REF* 3-1
limiting the number, *UG* 7-3
record source, *REF* 3-2 to 3-4
retrieving field values, *UG* 5-8
SORTED BY clause, *REF* 3-5, 3-7, *UG* 7-1, 7-9 to 7-10
tables, *UG* 7-7
WITH clause, *REF* 3-6, *UG* 7-1, 7-4 to 7-9

record selection expression
tables, *UG* 13-5

Record stream, *INTRO* 6-1, *UG* 7-1
displaying fields, *INTRO* 6-9 to 6-10
erasing, *INTRO* 8-2, 8-3
forming, *INTRO* 6-2 to 6-5, *REF* 3-1
from collections, *REF* 3-2
from domains, *REF* 3-2
from lists, *REF* 3-3
limiting, *INTRO* 6-2 to 6-6
modifying fields, *INTRO* 7-9 to 7-10
modifying records in, *INTRO* 7-9 to 7-10
naming, *REF* 3-6
sorting, *INTRO* 6-6 to 6-9, *REF* 3-7
specifying conditions, *REF* 3-6
specifying number, *REF* 3-4
stopping display, *INTRO* 6-10

Record streams
context block, *UG* 12-6 to 12-7

Records
comparing, *UG* 7-4 to 7-6
components of, *REF* 1-9
defining, *REF* 1-10, 4-1 to 4-9, 5-60 to 5-62,
UG 1-2, 5-1 to 5-18
deleting, *REF* 5-69, *UG* 6-2
editing, *REF* 5-80 to 5-85
establishing a single record context, *REF* 5-198
example data items, *UG* 5-2F
flat, *UG* 5-13F

- Records (cont.)
 - forming streams, *REF* 3-1
 - grouping, *UG*
 - hierarchical, *REF* 2-19, *UG* 5-13 to 5-18
 - limiting access, *UG* 14-2
 - limiting fields in a, *UG* 14-3
 - logical, *REF* 4-1
 - modifying, *UG* 6-2
 - passing to DATATRIEVE, DTPUTP, *CALL* 4-28, 8-25
 - removing from collections, *REF* 5-75 to 5-79
 - retrieving from DATATRIEVE, DTGETP, *CALL* 4-26, 8-14
 - rules for naming, *UG* 5-3
 - sample definition, *UG* 5-2F
 - selecting, *REF* 5-197 to 5-201
 - showing, *REF* 5-207
 - sorting, *REF* 5-213 to 5-214
 - storing, *REF* 5-215 to 5-222
 - transferring, *CALL* 4-24 to 4-33
 - with missing values, *INTRO* 5-3
- REDEFINES clause, *REF* 5-175 to 5-176, *UG* 5-9
 - in field definition, *REF* 4-7
- Relational operators, *INTRO* 6-4t, *REF* 2-17 to 2-20, 2-20t, *UG* 7-4 to 7-8
 - summary of, *UG* 7-8T
- RELEASE command, *INTRO* 6-20, *REF* 5-177 to 5-179, *UG* 1-3
- Releasing
 - domains and collections, *REF* 5-123 to 5-125
 - global variables, *REF* 5-39, 5-40
 - local variables, *REF* 5-39
- REMDTR, *CALL* 1-1, 1-3
- Remote Access Interface, *REF* 1-12
- Remote Call Interface, *DOCSUP* 1-1
 - See Call Interface*
 - installation file for, *DOCSUP* 3-5
 - NCPOS module, *DOCSUP* 1-3, 1-7, 3-4
 - requirements for using, *DOCSUP* 1-3
 - specifying LUNs for, *DOCSUP* 3-2
 - using, *DOCSUP* 1-4
- Remote Terminal Interface, *REF* 1-12, *UG* 1-6 to 1-7, *CALL* 1-1
 - advantages of, *CALL* 1-3
 - copying domains, *CALL* 2-3 to 2-4
 - example, *CALL* 2-2, 2-3, 2-5
 - exiting, *CALL* 2-2
 - invoking, *CALL* 2-1
 - prompt, *REF* 1-12
 - specifying a node, *CALL* 2-1
 - specifying an account, *CALL* 2-1
 - testing DATATRIEVE, *CALL* 2-2 to 2-3
 - using REMDTR, *CALL* 1-3
- REPEAT statement, *REF* 5-180 to 5-182, *UG* 8-1 to 8-3
 - BEGIN-END statement in, *REF* 5-32
 - declaring variables in, *REF* 5-39
 - invoking procedures with, *REF* 5-10
 - nesting in procedures, *REF* 5-182
 - with STORE statement, *REF* 5-217
- REPLACE command, *UG* 17-8 to 17-10
- REPLACE command (Editor), *REF* 5-94 to 5-96
- Report header, *REPORT* 2-7 to 2-10
- REPORT statement, *REPORT* 1-4, 1-5, 2-1, 2-2 to 2-3, *REF* 5-183 to 5-186, *UG* 1-3
 - access mode required, *REF* 5-172
 - identifying data, *REPORT* 2-2
 - retrieving data, *UG* 2-3 to 2-4
 - selecting output medium, *REPORT* 2-19 to 2-21
- Report Writer, *REF* 1-11
 - AT BOTTOM statement, *REF* 5-186 to 5-188
 - AT BOTTOM statement summary elements, *REF* 5-187t
 - AT TOP statement, *REF* 5-188 to 5-191
 - AT TOP statement header and summary elements, *REF* 5-189t
 - invoking, *REPORT* 1-5, 2-2 to 2-3, *REF* 5-183 to 5-186
 - print list elements, *REF* 5-193t
 - print list modifiers, *REF* 5-193t
 - PRINT statement, *REF* 5-192 to 5-194
 - prompt, *REF* 1-8
 - SET statement, *REF* 5-194 to 5-196
- REPORT_HEADER, *REPORT* 3-12, 3-13
 - in AT TOP statement (RW), *REF* 5-189
- Reports, *INTRO* 9-1 to 9-6
 - cross tabulations, *REPORT* 3-13 to 3-14
 - dating, *REPORT* 2-7, 2-9
 - designing, *REPORT* 2-1 to 2-21
 - general features, *INTRO* 9-1 to 9-2
 - hierarchical records, *REPORT* 3-15 to 3-16
 - naming, *REPORT* 2-1, 2-7
 - output to disk file, *INTRO* 9-5
 - output to line printer, *INTRO* 9-5
 - page numbers, *REPORT* 2-7
 - periodic, *REPORT* 1-1
 - print list options, *INTRO* 9-5
 - query, *REPORT* 1-1
 - sample, *REPORT* 1-2
 - saving specifications, *INTRO* 9-6
 - special page headings, *REPORT* 3-12 to 3-13
 - specification, *INTRO* 9-2 to 9-5, *REPORT* 1-4 to 1-5, 2-1 to 2-2
 - title page, *REPORT* 3-10 to 3-12
 - writing, *REF* 5-183 to 5-186

Reserved names, *DOCSUP* 2-2

Response time
 reducing with QCPRS utility, *UG* 21-4

Restructuring domains
 examples, *UG* 16-1 to 16-9

Retrieving data, *INTRO* 6-1 to 6-21, *UG* 2-3 to 2-4
 displaying all records, *INTRO* 6-1

Retrieving messages, *DTMSG*, *CALL* 4-17, 8-21

Retrieving print lines, *DTLINE*, *CALL* 4-17, 8-20

Retrieving records from DATATRIEVE,
 DTGETP, *CALL* 4-26, 8-14

RETURN key, *REF* 1-6

RMS domains
 defining, *REF* 5-45 to 5-47

RMS facility
 capabilities, *UG* 6-1 to 6-3

ROOT line
 in ODL files, *DOCSUP* 3-5

Routines, *CALL* 8-8 to 8-29
 DTCMD, *CALL* 4-14, 8-9
 DTCONT, *CALL* 4-27, 4-28, 8-11
 DTFINI, *CALL* 4-34, 8-13
 DTGETP, *CALL* 4-26, 8-14
 DTINIT, *CALL* 4-13, 8-16
 DTLINE, *CALL* 4-17, 8-20
 DTMSG, *CALL* 4-17, 8-21
 DTPEOF, *CALL* 4-28, 8-23
 DTPUTP, *CALL* 4-28, 8-25
 DTPVAL, *CALL* 4-22, 8-27
 DTUNWD, *CALL* 4-34, 8-29

RSE
See Record selection expression

RUN command line, *DOCSUP* 2-5

Running programs that call DATATRIEVE,
CALL 3-1 to 3-9

RW> prompt, *REF* 1-8, *UG* 2-4

S

S (sign)
 picture string characters, *REF* 5-157

Sample editing session, *UG* 17-13 to 17-14

Sample report, *REPORT* 1-2

Saving reports
 using command files, *REPORT* 1-7 to 1-8
 using procedures, *REPORT* 1-6 to 1-7

Security
 dictionary definitions, *UG* 20-1 to 20-11

SELECT statement, *INTRO* 7-1 to 7-5, *REF* 5-197 to 5-201
 forms of, *INTRO* 7-2 to 7-3

SELECT statement, (cont.)
 SHOW command with, *INTRO* 7-3 to 7-4
 with DROP statement, *REF* 5-198

Semicolon, *REF* 1-6
 in DECLARE PORT statement, *REF* 5-41
 in DEFINE DOMAIN clause, *REF* 5-48
 in DEFINE PORT command, *REF* 5-55
 in DEFINE RECORD command, *REF* 5-61
 in DELETE command, *REF* 5-69
 in domain definition, *REF* 5-46
 with DEFINE DOMAIN command, *UG* 4-2
 with DELETE command, *UG* 21-3

Sequence numbers, *UG* 20-2

Sequential files, *INTRO* 3-3 to 3-4, *REF* 1-10, *UG* 6-1 to 6-7
 compared with indexed, *UG* 6-2T
 creating, *REF* 5-51, 5-52
 defining, *UG* 6-3 to 6-4
 MODIFY statement with, *INTRO* 8-3
 modifying, *INTRO* 7-1 to 7-6
 modifying records in, *REF* 5-53

SET ABORT command, *REF* 5-202, 5-204, *UG* 10-5, 19-4
 effect on ABORT statement, *REF* 5-15, *UG* 19-4
 with REPEAT statement, *REF* 5-181

SET COLUMNS_PAGE command, *REF* 5-202, 5-204, *UG* 19-1 to 19-4

SET command, *REF* 5-202 to 5-205
 terminal control, *UG* 1-3

SET DICTIONARY command, *REF* 5-203, 5-204, *UG* 3-3
 in QUERY.INI file, *UG* 2-3

SET GUIDE command, *INTRO* 4-3, *REF* 5-203, *UG* 2-8, 10-2
 in QUERY.INI file, *UG* 2-3

SET NO ABORT command, *REF* 5-202, 5-204, *UG* 10-5

SET NO PROMPT command, *REPORT* 1-3, *REF* 5-203

SET PROMPT command, *REF* 5-203, *UG* 19-4 to 19-5

SET statement (Report Writer), *REPORT* 1-4, *REF* 5-194 to 5-196
 COLUMNS_PAGE, *INTRO* 9-3, *REPORT* 2-2, 2-5, *REF* 5-194
 DATE, *REPORT* 2-9, *REF* 5-196
 LINES_PAGE, *REPORT* 2-5 to 2-6, *REF* 5-194
 MAX_LINES, *REPORT* 2-6 to 2-7, *REF* 5-194
 MAX_PAGES, *REPORT* 2-6 to 2-7, *REF* 5-194
 NO DATE, *REPORT* 2-10

SET statement (Report Writer), (cont.)
 NO NUMBER, *REPORT* 2-10
 NUMBER, *REF* 5-196
 REPORT_NAME, *REPORT* 2-2, 2-7, 2-7 to 2-9, *REF* 5-196
 SET TERMINAL command, *UG* 19-2
 SETUP.DTR command
 RSTS/E systems, *INTRO* 2-2
 RSX systems, *INTRO* 2-2
 Severity codes, DAB\$W_ERR_SEV, *CALL* 4-18, 8-5
 SHARED access option, *REF* 5-169
 SHOW ALL command, *INTRO* 2-3, *REF* 5-206, *UG* 21-2
 SHOW COLLECTIONS command, *REF* 5-178, 5-206
 SHOW command, *REF* 5-199, 5-206 to 5-208, *UG* 1-3, 2-2
 with domains, *UG* 4-3
 SHOW CURRENT command, *REF* 5-206
 SHOW DICTIONARY command, *INTRO* 2-2, *REF* 5-206, *UG* 3-2 to 3-3
 SHOW DOMAINS command, *INTRO* 2-4, *REF* 5-207
 SHOW FIELDS command, *REPORT* 1-5, *REF* 5-178, 5-207
 SHOW PROCEDURES command, *REF* 5-207
 SHOW READY command, *INTRO* 2-4, *REF* 5-172, 5-178, 5-207
 SHOW RECORDS command, *INTRO* 2-4, *REF* 5-207
 SHOW SPACE command, *REF* 5-207
 SHOW TABLES command, *REF* 5-207, *UG* 13-8
 SHOWP command, *REF* 5-209 to 5-210, *UG* 1-3
 SIGN clause, *REF* 5-211 to 5-212, *UG* 5-9
 in field definition, *REF* 4-7
 LEADING, *REF* 5-211
 SEPARATE, *REF* 5-211
 TRAILING, *REF* 5-211
 Single record context, *UG* 12-14
 SKIP
 in AT BOTTOM statement (RW), *REF* 5-187
 in AT TOP statement (RW), *REF* 5-189
 print list element, *REF* 5-160
 RW print list element, *REF* 5-193
 Slash (/), *REPORT* 2-8, 2-16
 edit string character, *REF* 5-105, 5-110
 Sort keys, *INTRO* 6-7, *REPORT* 3-2 to 3-8
 ASCENDING, *REF* 3-7
 DESCENDING, *REF* 3-7
 in SORT statement, *REF* 5-213
 in SUM statement, *REF* 5-223
 multiple, *REPORT* 3-4, *REF* 3-8
 SORT statement, *REF* 5-213 to 5-214, *UG* 1-3
 access mode required, *REF* 5-172
 ASCENDING order, *REF* 5-213, 5-223
 DECREASING order, *REF* 5-213, 5-223
 DESCENDING order, *REF* 5-213, 5-223
 INCREASING order, *REF* 5-213, 5-223
 SORTED BY clause, *REPORT* 3-2 to 3-8, *UG* 7-9 to 7-10
 format, *REF* 3-7
 in RSE, *REF* 3-7
 sort keys, *REF* 3-7
 SPACE, *REPORT* 2-13, 2-14
 in AT BOTTOM statement (RW), *REF* 5-187
 in AT TOP statement (RW), *REF* 5-189
 in SHOW command, *REF* 5-207
 print list element, *REF* 5-160
 RW print list element, *REF* 5-193
 Special page headings, *REPORT* 3-12 to 3-13
 Specifying domain names
 DEFINE DOMAIN command, *UG* 4-1 to 4-2
 Specifying ranges
 in editing commands, *REF* 5-83
 Square brackets ([])
 in directory names, *DOCSUP* 2-2
 Stallpoints
 see States
 Starting
 an ADT session, *INTRO* 3-6
 DATATRIEVE, *INTRO* 2-1 to 2-4
 Startup banner, *UG* 2-1
 Statements
 See Commands and statements
 States, DATATRIEVE, *CALL* 4-1, 4-8 to 4-9, 8-3t
 Statistical expressions, *REF* 2-11 to 2-13
 RSE in, *REF* 2-13
 Statistical functions, *REPORT* 1-8 to 1-9, 2-12, 2-17 to 2-19, *REF* 2-11t
 AVERAGE, *REPORT* 1-8, 2-17
 COUNT, *REPORT* 1-8, 2-17
 MAX, *REPORT* 1-8, 2-18
 MIN, *REPORT* 1-8, 2-18
 TOTAL, *REPORT* 1-8, 2-17
 Status code, DAB\$W_ERR_CODE, *CALL* 4-18, 8-5
 Stopping command execution, DTUNWD, *CALL* 4-34, 8-29
 STORE statement, *INTRO* 5-1 to 5-5, *REF* 5-215 to 5-222, *UG* 1-3, 2-5
 access mode required, *REF* 5-172
 assigning field values with, *REF* 5-23, 5-26
 cancelling, *INTRO* 5-3
 data types, *INTRO* 5-4

STORE statement, (cont.)
 effect of ABORT statement, *REF* 5-16
 in FOR statement, *REF* 5-218
 prompts, *REF* 5-216
 REPEAT with, *INTRO* 5-3
 using, to store into a port, *CALL* 4-26
 VERIFY clause, *REF* 5-217

STRLEN, parameter to DTINIT, *CALL* 4-11

Sublists, *UG* 5-17
 retrieving values, *UG* 15-10 to 15-11

SUBSTITUTE command, *UG* 17-10 to 17-11

SUBSTITUTE command (Editor), *REF* 5-96 to 5-98

Substitution directive (!CMD), *CALL* 4-15, 8-9

SUM statement, *REPORT* 1-3, 1-3 to 1-4, *REF* 5-223 to 5-225, *UG* 1-3
 access mode required, *REF* 5-172

Summaries
 group, *REPORT* 3-8 to 3-9

Summarizing data, *REPORT* 2-17 to 2-19

Summary lines, *REPORT* 1-8 to 1-9, 2-17 to 2-19

Summary statistics, *REPORT* 1-8 to 1-9

SUPERSEDE clause, *UG* 6-6
 in DEFINE FILE command, *REF* 5-50

Syntax
 commands and statements, *REF* 5-1
 keywords in, *REF* 1-5
 symbols and conventions, *REF* 5-2, 5-2t

System security
 using access control lists, *UG* 20-1 to 20-11

T

T (text)
 edit string character, *REPORT* 2-15, *REF* 5-105

TAB
 in AT BOTTOM statement (RW), *REF* 5-187
 in AT TOP statement (RW), *REF* 5-189
 print list element, *REF* 5-160
 RW print list element, *REF* 5-193

Tables, *UG* 13-1 to 13-10
 and workspace, *UG* 13-7
 code/translation strings, *REF* 5-63, *UG* 1-4, 13-1 to 13-7
 comparing with IN, *REF* 2-19
 creating, *UG* 13-2 to 13-5
 default edit string, *REF* 5-64
 defining, *REF* 1-11, 5-63 to 5-65, *UG* 3-1
 defining and using, *UG* 13-4 to 13-5
 deleting, *REF* 5-69, *UG* 13-9
 displaying, *UG* 13-8

Tables, (cont.)
 displaying contents, *UG* 13-8
 editing, *REF* 5-80 to 5-85, *UG* 13-4, 13-8 to 13-9
 ELSE clause, *UG* 13-3
 functions and uses of, *REF* 1-11, *UG* 1-4
 IF-THEN-ELSE statement, *UG* 13-6
 in a record selection expression, *UG* 13-5
 in value expressions, *REF* 2-11
 invoking, *REF* 5-64
 maintaining, *UG* 13-9
 modifying, *REF* 5-65
 protecting, *UG* 13-9 to 13-10
 releasing, *REF* 5-177 to 5-179
 retrieving values with VIA, *REF* 2-11
 sample, *UG* 13-1 to 13-2
 showing, *REF* 5-207
 using IN relational operator, *UG* 13-5
 using with the Call Interface, *CALL* 1-5
 VALID IF clause, *UG* 13-6 to 13-7
 validating data, *UG* 13-6 to 13-7
 VIA value expression, *UG* 13-7
 WITH in, *UG* 13-5

Task Builder
 using, with Callable DATATRIEVE, *CALL* 3-4 to 3-6

Task Builder command file
 creating, *DOCSUP* 3-2
 editing, *DOCSUP* 3-2
 example of, on RSTS, *CALL* 3-4
 example of, on RSX-11M-PLUS, *CALL* 3-5
 sample, *DOCSUP* 3-3

Task Builder command files
 editing, *DOCSUP* 3-2 to 3-4

Task building applications, *DOCSUP* 3-2

TASK option
 with TASK command, *DOCSUP* 2-5

Task size, reducing
See Optimizing workspace

Terminal Interface
See Remote Terminal Interface

Termination characters, *REF* 1-6 to 1-7
 RETURN key, *REF* 1-6
 semicolon, *REF* 1-6

Terminology
 DATATRIEVE, *INTRO* 1-2

THEN statement, *REF* 5-226 to 5-227

TI
 for output to terminal, *REPORT* 2-20

TI:
 video screen, *DOCSUP* 2-4

Title page, *REPORT* 3-10

TODAY value expression, *REF* 2-9

TOTAL statistical function, *REPORT* 2-17 to 2-18, *REF* 2-11
 in AT BOTTOM statement (RW), *REF* 5-187
 in AT TOP statement (RW), *REF* 5-189
 Totals
 SUM statement, *REF* 5-224
 Transferring control
 with ABORT, *REF* 5-15
 Transferring data, *CALL* 4-16 to 4-33
 Transferring records, *CALL* 4-24 to 4-33
 Transfers
 with IF-THEN-ELSE statement, *REF* 1-3
 TYPE command, *UG* 17-11 to 17-13
 TYPE command (Editor), *REF* 5-98 to 5-100

U

UIC
 See *User identification code*
 Underscore (), *REF* 1-4, *UG* 1-2, 4-2
 in record name, *UG* 5-3
 UNITS line
 in CMD files, *DOCSUP* 3-3
 UNITS Task Builder qualifier, *CALL* 3-8
 Unwinding, DTUNWD, *CALL* 4-34, 8-29
 Uppercase letters
 in string literals, *REF* 2-3
 USAGE clause, *REF* 5-228 to 5-230, *UG* 5-9, 5-10 to 5-11
 COMP data type, *REF* 5-229
 COMP-1 data type, *REF* 5-229
 COMP-2 data type, *REF* 5-229
 COMP-3 data type, *REF* 5-229
 COMP-5 data type, *REF* 5-230
 DATE, *REF* 5-230
 in DECLARE statement, *REF* 5-38
 in field definition, *REF* 4-7
 User identification code, *UG* 3-3
 keys, *UG* 20-2 to 20-3
 USING clause, *REPORT* 2-12, 2-14 to 2-15
 in PRINT statement, *REF* 5-161
 in RW PRINT statement, *REF* 5-193
 in STORE statement, *REF* 5-215

V

V (decimal point)
 picture string character, *REF* 5-157
 VALID IF clause, *REF* 5-24, 5-28, 5-231 to 5-232, *UG* 5-9
 in field definition, *REF* 4-7
 in record definition, *UG* 12-8

Value expressions
 concatenated, *REF* 2-15 to 2-16
 DATE, *REF* 2-9 to 2-10
 defined, *REF* 2-1
 displaying, *REF* 5-73 to 5-74
 global variables in, *REF* 2-8
 in arithmetic expressions, *REF* 2-14 to 2-15
 in AT BOTTOM statement (RW), *REF* 5-187
 in AT TOP statement (RW), *REF* 5-189
 in detail lines, *REPORT* 2-11, 2-12
 local variables in, *REF* 2-8
 prompting, *REPORT* 1-6 to 1-7, *REF* 2-10 to 2-11
 statistical, *REF* 2-11 to 2-13
 table, *REF* 2-11
 TODAY, *REF* 2-9
 types of, *REF* 2-1
 use of, *REF* 2-1
 variables in, *REF* 2-7 to 2-9
 Variables, *REPORT* 2-12, *UG* 11-1 to 11-9
 as counters, *UG* 11-6 to 11-9
 assigning values to, *REF* 5-27 to 5-30, 5-39
 context, *REF* 2-6
 declaring, *REF* 2-7, 5-38 to 5-40, *UG* 11-1
 defining global, *REF* 2-8, 5-38
 defining local, *REF* 2-8, 5-31
 displaying with SHOW FIELDS, *REF* 5-40
 formatting for output, *REF* 5-101 to 5-113
 global, *UG* 11-4
 in detail lines, *REPORT* 2-11
 in value expressions, *REF* 2-7 to 2-9
 initializing, *REF* 2-7
 initializing global, *REF* 5-38
 initializing local, *REF* 5-32
 local, *UG* 11-4 to 11-5
 releasing global, *REF* 2-8, 5-40, 5-177 to 5-179
 storing values, *UG* 11-2 to 11-6
 VERIFY clause
 in MODIFY statement, *REF* 5-217
 in STORE statement, *REF* 5-217, 5-220, *UG* 12-8
 VERIFY USING clause
 in STORE statement, *REF* 5-215, 5-217
 VIA clause, *REF* 2-11, 5-64
 VIA value expression, *UG* 13-7
 View domains, *UG* 14-1 to 14-7
 combining data, *UG* 14-4 to 14-5
 containing a list, *UG* 14-6 to 14-7
 defining, *REF* 5-47 to 5-49, *UG* 14-2
 limiting record access, *UG* 14-2
 using, *UG* 14-5 to 14-7

Virtual fields
 See COMPUTED BY clause
Volumes, *DOCSUP 2-2*

W

W (day letter)
 edit string character, *REF 5-102*
WHILE statement, *REF 5-233 to 5-234, UG 11-8*
 with STORE statement, *REF 5-220*
WITH clause
 in RSE, *REF 3-6*
Workspace, *UG 13-7*
 defined, *UG 18-1*
 optimizing use of, *UG 18-4 to 18-5*
WRITE access mode, *REF 5-170*
Writing
 application installation files, *DOCSUP 3-5*
 source programs, *DOCSUP 3-1*

X

X (alphanumeric)
 edit string character, *REF 5-104*
 picture string character, *REF 5-156*

Y

Y (year)
 edit string character, *REF 5-102*
YACHTS sample domain, *UG 2-2, 2-4*
 access all records, *UG 7-2*
 comparing records, *UG 7-4 to 7-6*
 SET COLUMNS_PAGE command, *UG 19-2 to 19-4*

Z

Z
 edit string character, *REF 5-107*
Zero (0)
 edit string character, *REF 5-110*
ZONED data type
 See COMP-5 data type
ZZDTR directory, *DOCSUP 2-4*

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
and Puerto Rico
call **800-258-1710**

In Canada
call **800-267-6146**

In New Hampshire,
Alaska or Hawaii
call **603-884-6660**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

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

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809-754-7575

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 _____

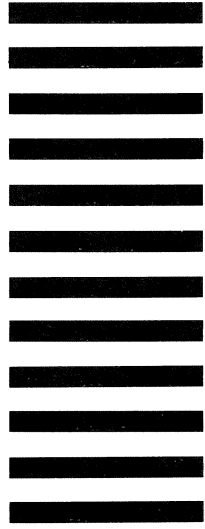
City _____ State _____ Zip Code
or
Country _____

-----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

ATTN: DISG Documentation ZKO2-2/N53
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, N.H. 03062

-----Do Not Tear - Fold Here and Tape-----

Cut Along Dotted Line