

DATATRIEVE-11

digital

Summary Description

DECLIT
AA
CROSS
U052C

Order Number: AA-U052C-TC

DATATRIEVE-11

Summary Description

Order Number: AA-U052C-TC

July 1989

This manual provides an overview of DATATRIEVE-11. It contains a glossary of DATATRIEVE concepts and terms, and a master index to the documentation set.

Operating Systems:

RSX-11M/M-PLUS
RSTS/E
Micro/RSX
Micro/RSTS
VMS with VAX-11 RSX

Software Version:

DATATRIEVE-11 Version 3.3

digital equipment corporation
maynard, massachusetts

First Printing, September 1983
Revised, November 1987
Revised, July 1989

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

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

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

© Digital Equipment Corporation 1983, 1987, 1989.

All Rights Reserved.
Printed in U.S.A.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DATATRIEVE	Rdb/VMS	VAX Information Architecture
DATATRIEVE-11	ReGIS	VAX Rdb/ELN
DEC	RSTS	VAXcluster
DECnet	RSTS/E	VAXinfo
DECUS	RSX	VAX/VMS
Micro/RSTS	RSX-11M	VAX-11 RSX
Micro/RSX	RSX-11M-PLUS	VMS
MicroVAX	UNIBUS	VT
MicroVMS	VAX	
PDP	VAX CDD	
PDP-11	VAX DATATRIEVE	

CID # 76315

ZK5061

Contents

Preface	v
----------------------	---

Chapter 1	DATATRIEVE-11 Overview	
1.1	What Is DATATRIEVE?	1-1
1.2	Who Is DATATRIEVE For?	1-2
1.3	How Is DATATRIEVE Different from COBOL or BASIC?	1-2
1.4	Information Management with DATATRIEVE	1-3
1.4.1	Data Definition	1-3
1.4.1.1	Simple Domains	1-4
1.4.1.2	View Domains	1-4
1.4.1.3	The Application Design Tool	1-5
1.4.2	Data Storage	1-5
1.4.3	Data Retrieval	1-5
1.4.4	Procedures	1-6
1.4.5	The Report Writer Facility	1-6
1.4.6	Guide Mode	1-7
1.4.7	The EDIT Command	1-7
1.4.8	Distributed Data Access and Call Interface	1-7
1.4.8.1	Remote Terminal Interface	1-9
1.4.8.2	Call Interface	1-9

Chapter 2	DATATRIEVE–11 Documentation Directory	
2.1	DATATRIEVE–11 Installation Guide	2–1
2.2	DATATRIEVE–11 Online Release Notes	2–1
2.3	DATATRIEVE–11 Summary Description	2–2
2.4	Introduction to DATATRIEVE–11	2–2
2.5	DATATRIEVE–11 User’s Guide	2–3
2.6	DATATRIEVE–11 Guide to Writing Reports	2–3
2.7	DATATRIEVE–11 Reference Manual	2–4
2.8	DATATRIEVE–11 Call Interface Manual	2–4
2.9	DATATRIEVE–11 Quick Reference Guide	2–5

Glossary

Index

Figures

1–1	Distributed Data Access and Call Interface	1–8
-----	--	-----

Preface

This manual provides an overview of DATATRIEVE–11. It contains a documentation directory, a glossary of DATATRIEVE concepts and terms, and a master index to the documentation set.

Intended Audience

This book is for people who want to know:

- What DATATRIEVE is
 - What DATATRIEVE does
 - What software works with DATATRIEVE
 - Where to find information about DATATRIEVE
-

Structure

This book is divided into two chapters, a glossary, and a master index:

Chapter 1	Provides an overview of DATATRIEVE–11.
Chapter 2	Outlines the DATATRIEVE–11 documentation set.
Glossary	Defines terms used throughout DATATRIEVE–11 documentation.
Master Index	Contains references to subjects in all DATATRIEVE–11 documentation.

DATATRIEVE-11 Overview

This chapter gives you an overview of DATATRIEVE-11. It explains when to use DATATRIEVE, how to use DATATRIEVE to define and manipulate your data, and how the special features of DATATRIEVE allow you to manage data. If any of the terms are unfamiliar to you, refer to the glossary. If you want detailed information about subjects discussed here, consult the master index.

This chapter is not intended to serve as an interactive session with DATATRIEVE. Examples in this chapter include only the text of DATATRIEVE commands, statements, and output, without system prompts and other information that appears on your screen when you actually use DATATRIEVE. See the *Introduction to DATATRIEVE-11* or the *DATATRIEVE-11 User's Guide* for more information on using DATATRIEVE.

In this manual, the DATATRIEVE-11 software is referred to as DATATRIEVE.

1.1 What Is DATATRIEVE?

DATATRIEVE is a comprehensive data management tool. It provides the following capabilities:

- A query language for defining, storing, updating, and displaying data
- A facility for writing reports
- Interactive and program-callable access to data
- Access to data distributed on a network connected by DECnet

1.2 Who Is DATATRIEVE For?

DATATRIEVE is designed for use by people with little or no computer experience and by people who are experienced users of computers.

Typical tasks DATATRIEVE-11 performs range from answering a casual question to creating a complex report. For example, using DATATRIEVE, you can query a personnel data file to determine which employees work in a particular department. You can use the same personnel file to produce a report with a statistical analysis of employee compensation by experience level.

You can also use DATATRIEVE-11 in a distribution facility with an order processing system. In this setting, you might want to extract sales data by territory and print a report using that data.

1.3 How Is DATATRIEVE Different from COBOL or BASIC?

DATATRIEVE is a “fourth-generation” language. This means that its syntax is more like English than that of BASIC or COBOL. DATATRIEVE executes commands and statements as you type them. You can tell DATATRIEVE what information you want by name, instead of specifying how to obtain the information as you would in COBOL or BASIC.

In COBOL or BASIC, each program describes the structure of the data it uses. DATATRIEVE lets you define data structures and stores those definitions separately. You can retrieve the data and use it in any number of ways without redefining the data each time.

DATATRIEVE also handles other common language functions automatically, without the need for language statements. For instance, DATATRIEVE:

- Finds data files and opens them
- Performs input and output operations
- Formats data for output
- Converts data types

As a result, you save many lines of code and applications run faster. In addition, the code is much more readable than COBOL or BASIC. For example, a typical programming language might retrieve the records of all employees named Perry like the following:

```
LOOP:   READ EMPLOYEE-FILE
        AT END EXIT
IF LAST_NAME NOT = "PERRY"
    GO TO LOOP
PRINT FIRST_NAME, LAST_NAME, ADDRESS    ...
GO TO LOOP
```

In DATATRIEVE, this becomes:

```
PRINT EMPLOYEES WITH LAST_NAME = "PERRY"
```

1.4 Information Management with DATATRIEVE

DATATRIEVE is a tool for managing data. Using DATATRIEVE, you can do the following:

- Define data in a way that fits your information management needs
- Store and modify data
- Retrieve data and display it on a terminal, record it in a file, or print it on paper
- Format data in reports
- Use a text editor to correct typing and syntax errors
- Access data files that are distributed across a network
- Call any of the information services of DATATRIEVE from a program written in a high-level programming language such as COBOL or BASIC

1.4.1 Data Definition

To create a DATATRIEVE-11 information management application, follow these two steps:

1. Define the data that you will use with DATATRIEVE commands. You need to define data only once to establish the foundation on which to build your applications.
2. Use DATATRIEVE statements to process the data associated with these definitions.

The data definition process involves creating DATATRIEVE domains. Domains represent relationships between actual physical data and descriptions of data. DATATRIEVE performs all data management in terms of domains.

1.4.1.1 Simple Domains

In its simplest form, a DATATRIEVE domain definition consists of the name of the domain and the names of the record definition and data file associated with it. Following is an example of a domain definition:

```
DEFINE DOMAIN PERSONNEL USING PERSONNEL_RECORD ON PERSON.DAT;
```

The record definition, `PERSONNEL_RECORD`, describes the data you want to use. The data file, `PERSON.DAT`, contains the data. The domain you created, `PERSONNEL`, connects the description with the data.

To use a domain, first get access to it with the `READY` command:

```
READY PERSONNEL
```

After you ready a domain, you can instruct DATATRIEVE to display data with a statement such as the following:

```
PRINT FIRST 2 PERSONNEL
```

In response to this statement, DATATRIEVE checks the record definition, gets the data requested from the file, and displays the following lines on your terminal:

ID	STATUS	FIRST NAME	LAST NAME	DEPT	START DATE	SALARY	SUP ID
00012	EXPERIENCED	CHARLOTTE	SPIVA	TOP	12-Sep-1972	\$75,892	00012
00891	EXPERIENCED	FRED	HOWL	F11	9-Apr-1976	\$59,594	00012

If you want to put this information in a file, you can specify an output file:

```
PRINT FIRST 2 PERSONNEL ON FILE.DAT
```

You can send the information to a line printer using the following statement:

```
PRINT FIRST 2 PERSONNEL ON LP:
```

1.4.1.2 View Domains

A view domain is a special type of domain that allows you to include any combination of fields and records from one or more domains. This means that you can define a view that selects information from a number of domains. View domains provide a way to reorganize data without changing the files storing the information. You can use a view domain just as you use a simple domain.

1.4.1.3 The Application Design Tool

The Application Design Tool (ADT) is a DATATRIEVE utility that simplifies the process of defining domains. ADT operates interactively and queries you for information. Your responses to the questions enable ADT to define a domain, define a record format, and create a data file. To start the Application Design Tool, type ADT.

1.4.2 Data Storage

Information management applications require that files be filled with data. Use the DATATRIEVE STORE statement for this purpose. When you enter a STORE statement, DATATRIEVE prompts you for field values. Before accepting input, DATATRIEVE also performs any validation checks specified by the record format.

1.4.3 Data Retrieval

You use data to make decisions, generate reports, and make your work easier. DATATRIEVE allows you to retrieve stored data with a set of statements. You need not be concerned with the underlying data structure or with the physical location of the data.

Data retrieval statements in DATATRIEVE consist of verbs modified by record selection expressions (RSEs). An RSE is a phrase that allows you to specify only the records you want from one or more domains. One DATATRIEVE statement can get the answer to a casual query or produce a detailed report.

Following is a typical data retrieval statement:

```
FIND PERSONNEL WITH START_DATE GT "01-Jan-1982"
```

This statement establishes a collection of records. It might yield a response such as the following:

```
[50 records found.]
```

Subsequent FIND statements can narrow down this CURRENT collection of 50 records. For example:

```
FIND CURRENT WITH DEPARTMENT EQUAL "SALES" OR "MARKETING" AND  
ZIP_CODE EQUAL 02138
```

DATATRIEVE may respond with the following:

```
[4 records found.]
```

You can then use the PRINT statement to display the collected data on the terminal screen, record it in a file, or print it on paper. For example, the following PRINT statement displays the data on your terminal screen:

```
PRINT ALL NAME, ADDRESS, PHONE
```

1.4.4 Procedures

With the DEFINE PROCEDURE command, you can specify sequences of DATATRIEVE-11 commands and statements and store them in a data dictionary for later use. You can invoke procedures by themselves and embed them in other procedures. You can use a procedure interactively or have an applications program invoke the procedure.

1.4.5 The Report Writer Facility

The DATATRIEVE Report Writer provides a set of formatting options for producing printed reports with page and column headings, page numbers, totals, subtotals, and other summary information.

You can tell the Report Writer what data and records to include in the report, what to title the report, what headings to print at the top of each page, and what summary lines to include. You can also tell the Report Writer to perform mathematical operations and can include the results of these operations in your report.

The following example shows how the Report Writer formats a page. The page has the date and page number in the upper right corner. In the following example, the Report Writer automatically sets the page width to 80 columns, spacing the fields of each record for you. Use the REPORT statement to invoke the Report Writer:

```
REPORT PERSONNEL WITH DEPT = "F11"  
SET REPORT_NAME = "DEPARTMENT F11: EMPLOYEES"  
PRINT ID, STATUS, NAME, START_DATE  
END_REPORT
```

The Report Writer formats your report as follows:

DEPARTMENT F11: EMPLOYEES

23-Jul-89

Page 1

ID	STATUS	FIRST NAME	LAST NAME	START DATE
00891	EXPERIENCED	FRED	HOWL	9-Apr-76
32432	TRAINEE	THOMAS	SCHWEIK	7-Nov-81
78923	EXPERIENCED	LYDIA	HARRISON	19-Jun-79
87701	TRAINEE	NATHANIEL	CHONTZ	28-Jan-82

1.4.6 Guide Mode

DATATRIEVE provides a self-teaching facility named Guide Mode. To invoke Guide Mode, issue the SET GUIDE command. Guide Mode helps you through a DATATRIEVE session with a series of questions. At any time in this mode of operation, you can request a list of commands, statements, names, or value expressions that you can enter.

Guide Mode is particularly useful to the inexperienced user who wants assistance during a DATATRIEVE session.

1.4.7 The EDIT Command

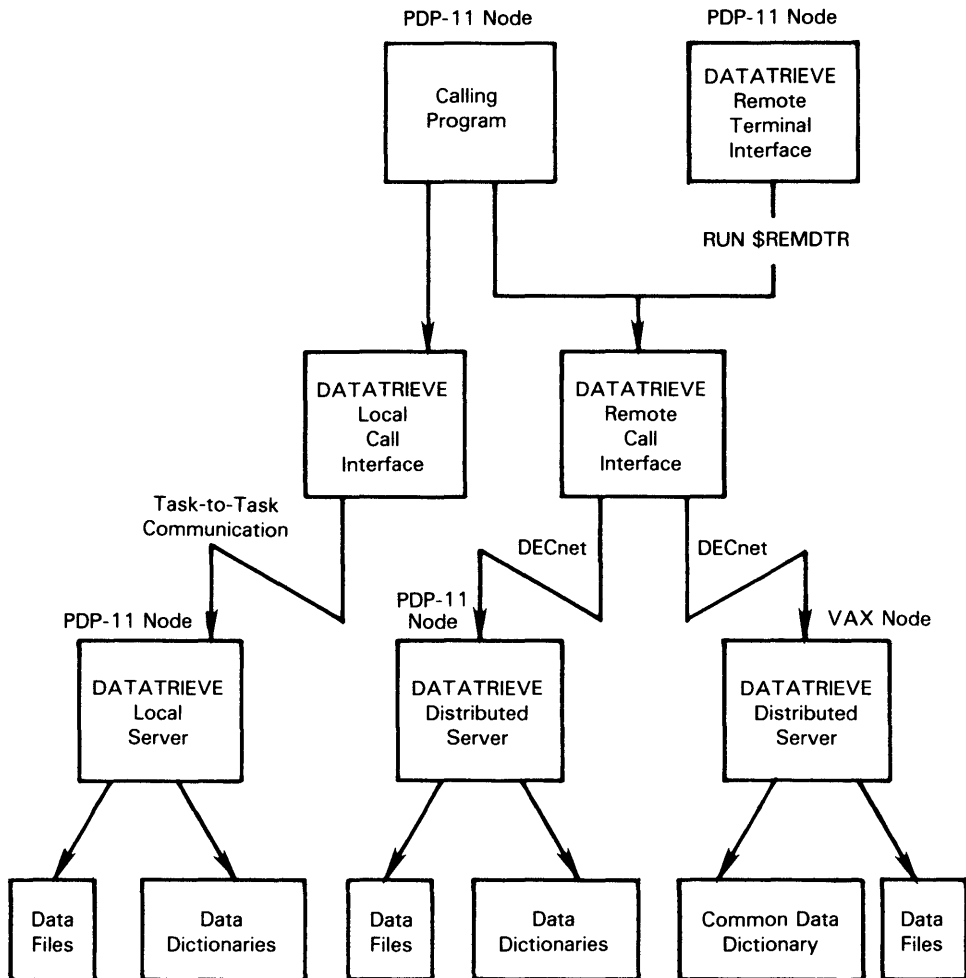
The EDIT command invokes the EDT text editor and allows you to modify existing domain, record, procedure, and table definitions stored in your current data dictionary. You can substitute characters and insert, replace, delete, and display lines in a dictionary definition.

Any mistakes you make while modifying definitions are not checked by the DATATRIEVE Editor, however.

1.4.8 Distributed Data Access and Call Interface

The DATATRIEVE Remote Terminal Interface and Call Interface give you access to distributed data and allow you to call DATATRIEVE from programs in high-level languages. The Call Interface includes a Local Call Interface, for calls to DATATRIEVE on your local PDP-11 node and a Remote Call Interface, for calls to DATATRIEVE on another node connected to your system. Figure 1-1 shows this relationship.

Figure 1-1: Distributed Data Access and Call Interface



ZK-6305-HC

1.4.8.1 Remote Terminal Interface

The DATATRIEVE Remote Terminal Interface (REMDTR) gives you access to distributed data, enabling you to run DATATRIEVE interactively on another computer connected to yours by DECnet. If you are logged on to a PDP-11 system, you can run DATATRIEVE on a PDP-11 or VAX system by typing the following:

```
RUN $REMDTR
```

REMDTR prompts you for a node name. Enter the name of a VAX or PDP-11 system with DATATRIEVE installed. REMDTR logs you on to that system, and you can have an interactive DATATRIEVE session at the remote system.

1.4.8.2 Call Interface

The DATATRIEVE-11 Call Interface allows you to call the data management services of DATATRIEVE-11 from programs in high-level languages such as BASIC, COBOL, and FORTRAN. You can make calls from a high-level language program to DATATRIEVE services on your local node or on a remote node or computer connected to your system by DECnet. You do not need DECnet to make a local call.

DATATRIEVE–11 Documentation Directory

This chapter describes the elements of the DATATRIEVE–11 documentation set and summarizes the objective, intended audience, and contents of each element.

2.1 DATATRIEVE–11 Installation Guide

Objective

To show how to install DATATRIEVE–11.

Audience

Those responsible for maintaining the operating system or installing DATATRIEVE–11.

Summary

The Installation Guide explains how to install DATATRIEVE–11 on different PDP–11 operating systems.

2.2 DATATRIEVE–11 Online Release Notes

Objective

To provide supplemental information about DATATRIEVE–11.

Audience

All DATATRIEVE–11 users.

Summary

The Online Release Notes that come with your installation kit provide information about the current version of DATATRIEVE-11 not included in the printed documentation set.

2.3 DATATRIEVE-11 Summary Description

Objective

To provide an overview of DATATRIEVE-11, to briefly explain the utilities for data management, and to guide users through the documentation set.

Audience

All users of DATATRIEVE-11.

Summary

This book offers an overview of DATATRIEVE-11. A documentation directory describes the manuals in the set, a glossary defines terms used in the documentation, and a master index facilitates access to specific concepts and features throughout the documentation.

2.4 Introduction to DATATRIEVE-11

Objective

To introduce DATATRIEVE to new users.

Audience

People who are unfamiliar with DATATRIEVE. Readers should be able to log in to a PDP-11 computer and display the contents of the default directory. If you need to review how to do that, consult the user's guide for your operating system.

Summary

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

2.5 DATATRIEVE–11 User's Guide

Objective

To describe the interactive use of DATATRIEVE–11.

Audience

People who:

- Are thoroughly familiar with the material covered in the *Introduction to DATATRIEVE–11*
- Have previous experience using DATATRIEVE
- Have experience in applications programming but are unfamiliar with DATATRIEVE

Summary

The manual begins by explaining how to set up your DATATRIEVE environment. It then explains how to use the DATATRIEVE–11 data dictionaries and how to use DATATRIEVE to define, retrieve, modify, and restructure data.

2.6 DATATRIEVE–11 Guide to Writing Reports

Objective

To explain how to produce reports using DATATRIEVE–11.

Audience

Users who want to produce reports based on data managed by DATATRIEVE. The reader should be familiar with the material covered in the *Introduction to DATATRIEVE–11* and with some topics discussed in the *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.

2.7 DATATRIEVE-11 Reference Manual

Objective

To provide comprehensive reference information about DATATRIEVE-11.

Audience

Users who have a working knowledge of DATATRIEVE or know the basic concepts of data processing and are familiar with their operating systems.

Summary

The manual explains in detail the rules governing the use 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 compose the DATATRIEVE-11 language.

2.8 DATATRIEVE-11 Call Interface Manual

Objective

To explain how to write high-level language programs that call DATATRIEVE, and how to use the DATATRIEVE-11 Remote Terminal Interface and Call Interface to access data on remote systems as well as the local system.

Audience

Applications programmers who are 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 systems.

2.9 DATATRIEVE–11 Quick Reference Guide

Objective

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

Audience

Experienced users of DATATRIEVE.

Summary

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

Glossary

Access control list (ACL)

A table that lists which users are allowed access to the domain, record, procedure, or table definitions stored in data dictionaries. Each definition in a data dictionary has an associated ACL. An entry in an ACL consists of the following:

- A way to identify users
- An access key
- A list of access rights

Application Design Tool (ADT)

A utility that aids you in creating domains, record definitions, and files by asking you questions and prompting you for the information.

Arithmetic operator

A symbol used to perform arithmetic operations. DATATRIEVE-11 operators include addition (+), subtraction (−), multiplication (*), and division (/).

Boolean expression

An expression that tells DATATRIEVE how to compare records and what values to use in the comparison. DATATRIEVE evaluates Boolean expressions as either true or false. In the statement PRINT PERSONNEL WITH STATUS = "TRAINEE", the expression STATUS = "TRAINEE" is a Boolean expression.

Boolean operator

A keyword that enables you to combine or negate Boolean expressions. Boolean operators are AND, OR, NOT, and BUT. For example, the expression STATUS = "TRAINEE" AND SALARY > 20000 contains the Boolean operator AND.

Call Interface

A DATATRIEVE-11 facility that allows you to call the data management services of DATATRIEVE from within a high-level language.

See also Local Call Interface and Remote Call Interface.

Calling program

A program written in a high-level programming language that uses the data management capabilities of DATATRIEVE.

Collection

A group of records gathered together from a larger group of records with the FIND statement. You can name a collection and can have several named collections available at once.

Column header

The heading that labels a column of data in a report or in the output of a PRINT statement. DATATRIEVE uses the field name as a column header unless the field definition includes a QUERY_HEADER clause.

Command

A valid combination of words that starts with a DATATRIEVE verb and that deals with a dictionary or its contents. Examples of commands are: DEFINE, SET, SHOW, FINISH, RELEASE, and READY. You cannot join commands or include them in statements.

See also Statement.

Command file

A file with a default file type .CMD that contains DATATRIEVE commands and statements. You tell DATATRIEVE to execute the commands in the file by typing the at sign (@) and the file name. DATATRIEVE then displays and executes each statement and command in the file as though you were entering them.

CURRENT

The keyword that identifies the most recently formed collection.

See also Collection.

Condition clause

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

Data dictionary

A file that contains information about your data and its organization. DATATRIEVE automatically connects you to your default dictionary when you enter DATATRIEVE. To display the name of the dictionary you are using, type SHOW DICTIONARY. To create a data dictionary, use the DEFINE DICTIONARY command. To change your current dictionary, use the SET DICTIONARY command.

Data file

A set of records stored as a unit on a storage medium, such as a disk. To create a data file, use the DEFINE FILE command.

See also Indexed file and Sequential file.

Database

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

DECnet

The Digital software facility that enables a user to access information on a remote computer through telecommunications lines. DECnet enables the operating system to function as a network node.

Detail lines

The formatted data lines that are the output of a Report Writer PRINT statement. Detail lines contain information from individual records in a report.

Domain

A named data structure that associates a data file and a record definition. Use the domain name to gain access to information in the data file as interpreted by the record definition. For example, the domain PERSONNEL associates the file PERSON.DAT and the record definition PERSONNEL_REC.

Edit string

A character or group of characters that instructs DATATRIEVE to use a specified format to display and print field values.

Elementary field

A record segment containing one item of information. Elementary fields do not contain other fields. For example, in an employee record, an elementary field might be the employee's last name.

See also Field and Group field.

Field

A segment of a record that contains data. A field can consist of a single data item such as a state abbreviation, or a combination of data items such as a street number and name. Related fields comprise a record.

See also Elementary field and Group field.

Field definition clause

A phrase that describes the data stored in a record field. The field definition clauses provided by ADT include PIC, USAGE IS, EDIT_STRING, and QUERY_NAME. You can specify other field definition clauses by using the DEFINE RECORD command.

Field tree

A hierarchical model of the fields in a record, based on the record definition in the data dictionary. Related fields are stored on the same branch of the tree to make data retrieval efficient.

File

See Data file.

Group field

A record segment containing one or more elementary or group fields. In the record definition PERSONNEL_REC, the group field NAME contains two elementary fields, LAST_NAME and FIRST_NAME.

See also Field and Elementary field.

Guide Mode

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

Index key

A field that DATATRIEVE uses to locate records in an indexed file. You define the primary index key and any alternate keys in the DEFINE FILE command. You cannot change the value of a primary index key; however, you can change the values of alternate index keys. For example, a good primary key for the records in a personnel file is an employee ID number because it uniquely identifies each employee. An example of an alternate key for the same group of records might be department number.

Indexed file

A file of records in which each record is identified by at least one index key. You can delete records from an indexed file, but you cannot change the primary index key.

Information management

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

Interactive processing

A mode of computer operation in which you control the actions of the computer by entering commands and data at a terminal rather than by using a program. When you communicate with DATATRIEVE you are using interactive processing.

Keyword

A word that has a specific meaning for DATATRIEVE. Examples of keywords are DEFINE, SET, and REPORT. The names that you define for domains and tables cannot exactly match a DATATRIEVE keyword.

Literal

A value expression that is either a character string enclosed in quotation marks or a number.

Local Call Interface

A Call Interface that allows you to access DATATRIEVE data management services from within a high-level language program on the same node without using DECnet.

See also Call Interface and Calling program.

Name clause

A component of a record selection expression (RSE) that tells DATATRIEVE what to name a record stream.

See also Record selection expression (RSE).

Primary key

The index key whose value determines the order of records in an indexed file. You cannot modify or erase the value in a primary key field.

See also Index key.

Print list

Value expressions such as keywords, symbols, and field names that are used to display and format record fields. Some Report Writer print list items include SKIP, COL, SPACE, and (-); these items are used to format detail and summary lines in a report.

Privilege

Access to data and data definitions for a specific purpose. There are four privileges to a definition in a data dictionary: R (read), W (write), M (modify), and E (execute or extend).

Privileges are defined in the access control list (ACL) for a dictionary definition. Users must have C (control) privilege to access an ACL.

See also Access control list (ACL).

Procedure

A named sequence of statements and commands that you store in a data dictionary to perform certain functions. To create a procedure, use the DEFINE PROCEDURE command. To execute a procedure, type a colon (:) and the procedure name. DATATRIEVE executes a procedure as though you had just typed the commands and statements it contains.

Prompt

A character or phrase that DATATRIEVE displays on your terminal to let you know it is waiting for input. The DTR> prompt indicates that you should type in commands and statements that tell DATATRIEVE what to do next. Other types of prompts, such as CON>, DFN>, RW>, and Enter, help you know what DATATRIEVE expects from you.

Prompting expression

An expression you can specify to prompt for input when a statement executes.

Query header

A name that DATATRIEVE uses as a column header when displaying or printing values. You define a column header with the QUERY_HEADER clause. For example, STATUS is the column header for values from the field EMPLOYEE_STATUS in the record PERSONNEL_REC.

If you do not use the QUERY_HEADER clause to specify a column header, DATATRIEVE uses the field name.

Query name

A synonym for a field name, usually a shorter word. You define a field name synonym with the `QUERY_NAME` clause and can use the synonym anywhere you can use the field name.

Record

A collection of related fields stored as a unit. For example, one employee record may include an employee's name, address, department, salary, and starting date.

See also Field.

Record definition

A description of record fields that is stored in a data dictionary. A record definition tells `DATATRIEVE` how to identify and store information when it accesses a file.

Record selection expression (RSE)

A phrase defining conditions that individual records must meet before they are retrieved from data files. For example, the statement `PRINT PERSONNEL WITH STATUS = "TRAINEE"` contains the record selection expression `PERSONNEL WITH STATUS = "TRAINEE"`.

Record source

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

Record stream

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

See also Record selection expression (RSE).

Relational operators

Symbols used to compare values. For example, the `DATATRIEVE` statement `FIND PERSONNEL WITH SALARY > 10000` contains the relational operator `>` (greater than).

Remote Call Interface

A Call Interface that allows you to access `DATATRIEVE` data management services on another node from within a high-level language program. The other node must be connected to your system by `DECnet`.

See also Call Interface and Calling program.

Remote Terminal Interface (REMDTR)

A program that enables you to invoke DATATRIEVE on remote PDP-11 and VAX systems through DECnet.

Report header

The heading of a Report Writer report. Some report header options are page number and date in the top-right corner of the report and a centered title.

Report specification

A series of Report Writer statements that creates and formats a report.

Report Writer

A component of DATATRIEVE-11 that allows you to create, format, and print reports. The Report Writer can also include the results of mathematical operations in the report.

Restriction clause

A phrase in a record selection expression that tells DATATRIEVE which records to include in a record stream.

RSE

See Record selection expression (RSE).

Selected record

The one record in a collection that is available for display or modification without specifying a record selection expression. Use the SELECT statement to indicate a selected record.

Sequential file

A file whose records are stored and accessed in the order they are written to the file. A sequential file does not have an index. You cannot delete records from a sequential file.

See also Data file and Indexed file.

Sort key

A field that forms the basis for sorting. For example, in the statement PRINT PERSONNEL SORTED BY START_DATE, START_DATE is the sort key.

Statement

A valid combination of words starting with a DATATRIEVE verb that performs query, report, or data manipulation functions. A statement can contain other statements, but it cannot contain commands.

See also Command.

Statistical functions

Keywords, such as COUNT, AVERAGE, and TOTAL, that tell DATATRIEVE or its Report Writer to perform a particular calculation.

Substitution directive

An expression in a command or statement passed to DATATRIEVE-11 from a calling program. The substitution directive is replaced by parameters given in the program. For example, the program can prompt for the name of a domain and read it in:

```
          READ (5, 40) DOMAIN
40      FORMAT (A)
```

You pass the name of the domain with the !CMD substitution directive:

```
CALL DTR$COMMAND (DAB, (READY !CMD), DOMAIN)
```

Summary line

A line in a report that summarizes information included in the report. Use the AT TOP and AT BOTTOM statements to create summary lines.

Top-level field

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

Value expression

A string of symbols that specifies a value for DATATRIEVE to use when executing a statement. Character-string and numeric literals, arithmetic expressions, qualified field names, and variables are some of the DATATRIEVE value expressions you can use.

Variable

A value expression created by a DECLARE statement. For example, the following statement creates a variable, X, which can be assigned any 1- or 2-digit numerical value:

```
DECLARE X    PIC 99.
```


View domain

A special type of domain that allows you to include any combination of fields and records from one or more domains. To users accessing data with a view domain, the results are the same as if they used a simple domain. View domains are most useful for the following:

- Restricting access to sensitive parts of the database
- Making the results of frequently used selections and joins permanent

Virtual field

A field that appears in the record definition but not in the physical record. In DATATRIEVE, you create virtual fields with the **COMPUTED BY** clause. For example, you can add the field called **WEEKLY_SALARY** to **PERSONNEL_REC** by adding the following to the record definition:

```
05 WEEKLY_SALARY    COMPUTED BY SALARY/52  
                     EDIT_STRING IS $$,$$$.
```

Because the value of a **COMPUTED BY** field is computed when a statement executes, it occupies no space in the record.

Workspace

The maximum amount of memory you can use during a DATATRIEVE session. Use the **SHOW SPACE** command to display information about your DATATRIEVE workspace.

Index

This index contains entries for the entire DATATRIEVE-11 document set. Manuals are indicated by the following abbreviations:

INTRO	Introduction to DATATRIEVE-11
REPORT	Guide to Writing Reports
UG	User's Guide
REF	Reference Manual
CALL	Call Interface Manual

Page numbers followed by a "t" refer to tables; those followed by "f" refer to figures. For example, a reference to *REF* 5-3t would indicate a table on page 5-3 of the *DATATRIEVE-11 Reference Manual*.

:
 See Colon
 <
 See LESS_THAN relational operator
 >
 See GREATER_THAN relational operator
 Edit string characters
 9 (numeric), *REF*, 5-109
 Picture string characters
 9 (numeric), *REF*, 5-175
 @
 See AT sign

A

Aborting commands, *DTUNWD*, *CALL*, 4-31, 8-30
 Aborting procedures, *UG*, 9-13 to 9-14
 ABORT statement, *REF*, 5-14 to 5-18
 effect of SET ABORT, *REF*, 5-15
 in command file, *UG*, 10-6
 in STORE statement, *REF*, 5-259
 Absolute value function (FN\$ABS), *REF*, 6-4
 Access Block, DATATRIEVE
 See DATATRIEVE Access Block
 Access codes, *REF*, 5-83t
 Access control list
 access codes, *REF*, 5-83t
 adding entries to, *UG*, 19-12 to 19-13
 contents, *UG*, 19-1 to 19-7
 control (C) privilege, *UG*, 20-2
 creating, *UG*, 19-7 to 19-8
 creating for ports, *REF*, 5-70
 creating for procedures, *REF*, 5-72
 creating for records, *REF*, 5-76
 creating for RMS domains, *REF*, 5-58

Access control list (cont'd.)
 creating for tables, *REF*, 5-79
 creating for view domains, *REF*, 5-62
 defining entries, *REF*, 5-82 to 5-85
 deleting entries, *UG*, 19-13 to 19-14; *REF*, 5-89 to 5-90
 displaying, *UG*, 19-12, 20-2; *REF*, 5-250 to 5-251
 execute (E) privilege, *UG*, 20-2
 key, *UG*, 19-1 to 19-7
 lock type, *UG*, 19-1 to 19-7
 maintaining, *UG*, 19-11 to 19-14, 20-7
 modify (M) privilege, *UG*, 20-2
 privileges, *UG*, 19-4 to 19-7
 processing, *UG*, 19-8 to 19-11
 protecting data, *UG*, 19-1
 read (R) privilege, *UG*, 20-2
 sample, *UG*, 19-1F
 sequence number, *UG*, 19-1 to 19-7
 SHOWP command, *REF*, 5-250 to 5-251
 write (W) privilege, *UG*, 20-2
 Accessing domains, *INTRO*, 2-4
 EXCLUSIVE, *REF*, 5-195
 PROTECTED, *REF*, 5-195
 READY command, *REF*, 5-194 to 5-201
 restrictions, *REF*, 5-196
 results, *REF*, 5-197
 SHARED, *REF*, 5-195
 Access modes, *REF*, 5-196t
 EXTEND, *REF*, 5-196
 MODIFY, *REF*, 5-196
 READ, *REF*, 5-196
 required by DATATRIEVE statements, *REF*, 5-198t
 WRITE, *REF*, 5-196
 Access options, *REF*, 5-195t

Accounts

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

ACL

- See Access control list*

ADT

- See Application Design Tool*

ADT command, *INTRO*, 3-6; *UG*, 10-3; *REF*, 5-19 to 5-20

- See Also Application Design Tool*

ADVANCED HELP command, *UG*, 2-8

Alias

- domain name, *REF*, 5-194

ALIGNED_MAJOR_MINOR allocation, *REF*, 5-21

ALL

- in RSE, *REF*, 3-5
- in SHOW command, *REF*, 5-246
- print list element, *REF*, 5-179, 5-181t

Allocating LUNs, *CALL*, 3-9

ALLOCATION clause, *UG*, 6-7; *REF*, 5-21 to 5-23

- ALIGNED_MAJOR_MINOR, *REF*, 5-21

- in DEFINE FILE command, *REF*, 5-63

- in DEFINE RECORD command, *REF*, 5-74

- LEFT_RIGHT, *REF*, 5-21

- MAJOR_MINOR, *REF*, 5-21

Alphanumeric fields, *UG*, 5-12

- edit string characters, *REF*, 5-108 to 5-109

- picture string characters, *REF*, 5-176

Alphanumeric functions, *REF*, 6-2

AND Boolean operator, *INTRO*, 6-6; *UG*, 7-9; *REF*, 2-25

ANY relational operator, *REF*, 2-24

Application Design Tool, *UG*, 4-1; *REF*, 4-1

- defining data with, *INTRO*, 3-1 to 3-10

- defining records, *UG*, 5-1

- exiting, *REF*, 5-20

- invoking, *REF*, 5-19 to 5-20

- sample session, *INTRO*, 3-4 to 3-9

- use in definitions, *REF*, 1-12

Argument list, *CALL*, 8-7f

Arguments

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

Arithmetic expressions, *REF*, 2-15 to 2-17

- evaluating, *REF*, 2-16

- minus sign in, *REF*, 2-16

- parentheses in, *REF*, 2-16

- valid operators, *REF*, 2-16t

Arithmetic operators, *REF*, 2-16t

ASCENDING

- in SORT statement, *REF*, 5-254

ASCENDING (cont'd.)

- in SUM statement, *REF*, 5-266

- sort key, *INTRO*, 6-10; *REF*, 3-8

ASCII collating sequence, *REF*, 2-23

Assigning query names

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

Assignment statement, *UG*, 11-2; *REF*, 5-24 to 5-35

- for elementary fields, *REF*, 5-25 to 5-28

- for group fields, *REF*, 5-29 to 5-31

- for variables, *REF*, 5-32 to 5-35

Asterisk (*)

- edit string character, *REF*, 5-109

AT (@) sign

- executing command files, *UG*, 5-3

- using to invoke command files, *REF*, 5-11 to 5-13

AT BOTTOM statement (Report Writer), *REF*, 5-220 to 5-223

- OF field-name, *REPORT*, 2-19, 3-3, 3-9 to 3-10

- OF PAGE, *REPORT*, 2-19

- OF REPORT, *REPORT*, 2-19, 2-20, 3-15

- summary elements, *REF*, 5-221t

AT TOP statement (Report Writer), *REF*, 5-224 to 5-227

- header and summary elements, *REF*, 5-225t

- OF field-name, *REPORT*, 3-3

- OF PAGE, *REPORT*, 2-19, 3-11 to 3-15

- OF REPORT, *REPORT*, 2-19

AVERAGE statistical function, *REPORT*, 1-9, 2-19 to 2-20; *REF*, 2-13

- in AT BOTTOM statement (RW), *REF*, 5-221

- in AT TOP statement (RW), *REF*, 5-225

B

B (blank)

- edit string character, *REF*, 5-109

BASIC sample programs, *CALL*, 7-1 to 7-9

BEGIN-END statement, *UG*, 8-3 to 8-7; *REF*, 5-36 to 5-41

- declaring variables in, *REF*, 5-48

- initializing variables, *REF*, 5-37

- invoking procedures from, *REF*, 5-7

- restrictions, *REF*, 5-36

- results, *REF*, 5-37

- usage notes, *REF*, 5-38

- with REPEAT statement, *REF*, 5-38

BETWEEN relational operator, *INTRO*, 6-6; *UG*, 7-6, 7-7; *REF*, 2-21

Boolean expressions, *REF*, 2-19 to 2-27

Boolean expressions (cont'd.)

- Boolean operators, *REF*, 2–25 to 2–27
- commas in, *REF*, 3–7
- compound, *UG*, 7–9 to 7–10; *REF*, 2–26, 2–27, 2–27t
- defined, *REF*, 2–1
- evaluating, *REF*, 2–26
- grouping with parentheses, *REF*, 2–26
- in RSE, *REF*, 3–7
- relational operators, *REF*, 2–19 to 2–25
- using, *REF*, 2–19

Boolean operators, *UG*, 7–9; *REF*, 2–25 to 2–27

Brackets []

- used as syntax prompts, *UG*, 2–5

BT

See *BETWEEN* relational operator

BUFLEN, parameter to DTINIT, *CALL*, 4–10

BUT Boolean operator, *INTRO*, 6–6; *UG*, 7–9; *REF*, 2–25

C

Call Interface, *UG*, 1–7; *REF*, 1–14; *CALL*, 1–5, 1–5f

- closing, *CALL*, 4–31
- creating menus, *CALL*, 1–7
- initializing, *CALL*, 4–11, 4–12, 8–17
- overview, *CALL*, 4–1
- procedures used with, *CALL*, 1–7
- tables used with, *CALL*, 1–7
- writing programs that use, *CALL*, 4–1 to 4–31

Calls to DATATRIEVE

- DTCMD, *CALL*, 4–13, 8–9
- DTCONT, *CALL*, 4–25, 8–12
- DTFINI, *CALL*, 4–31, 8–14
- DTGETP, *CALL*, 4–24, 8–15
- DTINIT, *CALL*, 4–12, 8–17
- DTLINE, *CALL*, 4–15, 8–21
- DTMSG, *CALL*, 4–16, 8–22
- DTPEOF, *CALL*, 4–26, 8–24
- DTPUTP, *CALL*, 4–26, 8–26
- DTPVAL, *CALL*, 4–20, 8–28
- DTUNWD, *CALL*, 4–31, 8–30

Case function (FN\$UPCASE), *REF*, 6–20

Case sensitivity

- with relational operators, *UG*, 7–5

CHANGE, *UG*, 6–9

- in DEFINE FILE command, *REF*, 5–64

Character change mode, *UG*, 16–5

Characters

- edit string, *REF*, 5–103

Characters (cont'd.)

- floating edit string, *REF*, 5–114
- picture string, *REF*, 5–176
- printing, *REF*, 2–2

Character set, *REF*, 1–4

Character string literals

See *Literals*

Clauses

- in procedures, *UG*, 9–4 to 9–5

CLOSE command, *REF*, 5–42 to 5–43

Closing the Call Interface, *CALL*, 4–31

ICMD, substitution directive, *CALL*, 4–13

COBOL sample programs, *CALL*, 6–1 to 6–11

COL

- in AT BOTTOM statement (RW), *REF*, 5–221

- in AT TOP statement (RW), *REF*, 5–225

print list element, *REF*, 5–181t

RW print list element, *REF*, 5–230t

Collections, *INTRO*, 6–13 to 6–24

- as record source, *REF*, 3–2

changing content of, *INTRO*, 6–22 to 6–23

CURRENT, *REF*, 3–2

displaying, *INTRO*, 6–14 to 6–15

erasing, *INTRO*, 8–2 to 8–3

establishing with FIND, *REF*, 3–2

forming, *INTRO*, 6–14; *REF*, 5–130 to 5–132

making new CURRENT from old, *INTRO*, 6–17 to 6–18

naming, *INTRO*, 6–18 to 6–20

releasing, *INTRO*, 6–23 to 6–24; *REF*, 5–207 to 5–210

removing records from, *REF*, 5–94 to 5–99

renaming, *INTRO*, 6–21 to 6–22

SHOW command, *INTRO*, 6–16 to 6–17

showing, *REF*, 5–246

showing names of, *INTRO*, 6–20 to 6–21

sorting, *INTRO*, 6–15 to 6–16; *REF*, 5–254 to 5–256

Colon (:)

- using to invoke procedures, *UG*, 1–4; *REF*, 5–7 to 5–10

Column headers, *REPORT*, 2–12, 2–17 to 2–19

- in AT TOP statement (RW), *REF*, 5–225

printing, *REF*, 5–182, 5–184, 5–231

specifying, *REPORT*, 2–18 to 2–19

suppressing, *REPORT*, 2–17 to 2–18; *REF*, 5–182, 5–184, 5–231

wrapping, *REPORT*, 2–15

Column-page setting

See *SET COLUMNS_PAGE* command

COLUMN_HEADER, *REPORT*, 3–13

- Comma (,)
 - edit string character, *REF*, 5-113
- Command files, *UG*, 1-4, 10-1 to 10-10; *REPORT*, 2-21 to 2-22
 - aborting, *UG*, 10-6
 - comments, *UG*, 10-3; *REF*, 5-12
 - compared with procedures, *UG*, 10-1
 - contents, *UG*, 10-3
 - creating, *UG*, 10-2 to 10-3
 - creating with EXTRACT, *REF*, 5-126
 - default file type, *REF*, 5-12
 - editing, *UG*, 10-6
 - editing with, *UG*, 16-1
 - in FOR and REPEAT statements, *UG*, 10-9
 - in procedures, *REF*, 5-11
 - invocation command lines, *UG*, 10-4
 - invoking, *UG*, 10-3 to 10-6; *REF*, 5-11 to 5-13
 - maintaining, *UG*, 10-9 to 10-10
 - nesting, *UG*, 10-8 to 10-9
 - restrictions on invoking, *UG*, 10-4 to 10-6
 - sample, *UG*, 10-6 to 10-8
 - SET ABORT/SET NO ABORT, *UG*, 10-6
 - stopping execution of, *REF*, 5-14
 - uses, *UG*, 10-1
- Commands, passing, using DTCMD, *CALL*, 4-13, 8-9
- Commands and statements, *UG*, 1-3 to 1-4; *REF*, 1-2 to 1-7
 - alphabetic summary, *REF*, 5-3t to 5-6t
 - 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*, 5-3t to 5-6t
 - functions of, *REF*, 1-2
 - in procedures, *UG*, 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
- Comments, *REF*, 1-7
 - in command files, *REF*, 5-12
- COMP-1 data type, *REF*, 5-274
- COMP-2 data type, *REF*, 5-274
- COMP-3 data type, *REF*, 5-274
- COMP-5 data type, *REF*, 5-274
- COMP data type, *REF*, 5-274
- Compiling programs that call DATATRIEVE, *CALL*, 3-2

- Components of DATATRIEVE-11, *CALL*, 1-1 to 1-2
- Compound Boolean expressions, *REF*, 2-26
 - evaluating, *REF*, 2-27t
- Compound statements, *UG*, 1-4, 8-1 to 8-7; *REF*, 1-3
 - BEGIN-END, *UG*, 8-4 to 8-7; *REF*, 5-36 to 5-41
 - IF-THEN-ELSE in, *UG*, 8-5
 - in REPEAT, *UG*, 8-6 to 8-7
 - in STORE, *UG*, 8-5 to 8-6
 - FOR in, *UG*, 8-3 to 8-4
 - BEGIN-END, *UG*, 8-4
 - REPEAT in, *UG*, 8-1 to 8-3
 - THEN, *REF*, 5-270 to 5-271
 - variables in, *REF*, 2-9
- Compressing data dictionary, *UG*, 20-4 to 20-6
- Computed by clause, *UG*, 5-13
- COMPUTED BY clause, *UG*, 5-1; *REF*, 5-44 to 5-46
 - creating virtual fields, *REF*, 2-7, 5-44
 - in DECLARE statement, *REF*, 5-47
 - in field definition, *REF*, 4-8
- CON> prompt, *UG*, 2-5; *REF*, 1-9
- Concatenated expressions, *REPORT*, 3-4; *REF*, 2-2, 2-17 to 2-19
 - formatting output, *REF*, 2-18
- Conditional transfers
 - with IF-THEN-ELSE statement, *REF*, 1-3
- Conserving memory
 - See Optimizing workspace*
- CONT
 - See CONTAINING relational operator*
- CONTAINING relational operator, *INTRO*, 6-6; *UG*, 7-5; *REF*, 2-23
 - case-sensitivity, *REF*, 2-23
- Context, *UG*, A-1
 - establishing, *UG*, A-1 to A-16
- Context block
 - content of, *UG*, A-2 to A-4
 - existing collections, *UG*, A-5 to A-7
 - global variables, *UG*, A-4
 - record streams, *UG*, A-6 to A-8
- Context stack, *UG*, A-2 to A-9
 - lasting changes, *UG*, A-7
 - left
 - assignment statements, *UG*, A-13
- Context variables, *UG*, A-9 to A-13; *REF*, 2-8
 - field name qualifiers, *UG*, A-10
 - in FIND statement, *REF*, 3-6
 - in RSE, *REF*, 3-6
 - qualifying field names, *REF*, 2-7

- Context variables (cont'd.)
 - with FOR loops, *REF*, 3–6
 - with MODIFY statement, *UG*, A–14
 - with STORE statement, *UG*, A–14; *REF*, 5–261
- Continuation character, *REF*, 1–6 to 1–7
 - in character string literals, *REF*, 2–2
- Continuation prompt, *REF*, 1–9
- Continuing, DTCONT, *CALL*, 4–25, 8–12
- Control groups, *REPORT*, 3–1 to 3–10
 - levels of, *REPORT*, 3–5 to 3–8
 - sort keys, *REPORT*, 3–2 to 3–8
 - without sort keys, *REPORT*, 3–8 to 3–9
- Controlling output, *UG*, 18–1 to 18–5
 - column width, *UG*, 18–1 to 18–4
- Copying domains, *CALL*, 2–3 to 2–4
- Corruption of data
 - protection against, *UG*, 19–1
- COUNT statistical function, *REPORT*, 1–9, 2–19 to 2–20; *REF*, 2–13
 - in AT BOTTOM statement (RW), *REF*, 5–221
 - in AT TOP statement (RW), *REF*, 5–225
- CR (credit)
 - edit string character, *REF*, 5–111
- CREATE DICTIONARY command, *UG*, 20–1
- Cross tabulations, *REPORT*, 3–15 to 3–16
- CTRL/C, *UG*, 2–6
 - with BEGIN-END block, *REF*, 5–37
- CTRL/Z, *INTRO*, 2–5, 5–2
 - exiting from DATATRIEVE, *UG*, 2–6; *REF*, 5–123
- CURRENT
 - as target record stream, *UG*, A–22 to A–24
 - in record selection expressions, *INTRO*, 6–18
 - in SHOW command, *REF*, 5–247

D

DAB

- See DATATRIEVE Access Block*

DAB\$M_BUF_OVERFLOW flag, *CALL*, 8–21

DAB\$M_STR_OVERFLOW flag, *CALL*, 8–6

DAB\$V_BUFFER, *CALL*, 8–17

DAB\$V_STRING, *CALL*, 8–6t, 8–9

DAB\$W_ERR_CODE, *CALL*, 4–17, 8–4

DAB\$W_ERR_SEV, *CALL*, 4–17, 8–4, 8–22

DAB\$W_FLAGS, *CALL*, 8–5t

DAB\$W_STR_LEN, *CALL*, 8–7, 8–17

DAB fields, *CALL*, 8–5t

- DAB\$V_BUFFER, *CALL*, 8–17
- DAB\$V_STRING, *CALL*, 8–6t, 8–9
- DAB\$W_ERR_CODE, *CALL*, 4–17, 8–4, 8–22
- DAB\$W_ERR_SEV, *CALL*, 4–17, 8–4, 8–22

DAB fields (cont'd.)

- DAB\$W_STR_LEN, *CALL*, 8–7, 8–17

Data dictionary, *UG*, 1–3

- access privileges, *UG*, 20–2
- changing, *UG*, 3–2 to 3–4
- contents, *UG*, 3–1 to 3–2
- creating, *INTRO*, 3–10; *UG*, 2–2, 3–2
- default extension (.DIC), *UG*, 3–2
- defining, *REF*, 5–53 to 5–55
- deleting, *REF*, 5–86 to 5–87
- deleting definitions in, *UG*, 20–4
- determining size of, *UG*, 20–5
- displaying, *UG*, 3–3, 20–2
- editing definitions, *UG*, 20–1
- extracting from, *UG*, 20–7 to 20–8
- function, *UG*, 3–1
- maintaining, *UG*, 20–1 to 20–8
- modifying, *UG*, 20–2 to 20–4
- objects, *UG*, 20–1
- optimizing disk storage, *UG*, 20–4 to 20–6
- placing definitions into, *INTRO*, 3–9 to 3–10
- QCPRS utility, *UG*, 20–5
- QXTR utility, *UG*, 20–6
- security, *UG*, 19–1 to 19–14
- setting, *UG*, 3–3
- showing current, *REF*, 5–54, 5–247
- transferring definitions, *UG*, 20–1

Data files

- See Files*

DATATRIEVE

- components, *UG*, 1–5 to 1–8
- concepts and terms, *UG*, 1–1 to 1–5

DATATRIEVE–11, components of, *CALL*, 1–1 to 1–2

DATATRIEVE Access Block (DAB), *CALL*, 4–1, 4–8t, 8–1, 8–1t

- declaring, *CALL*, 4–8

DATATRIEVE states, *CALL*, 4–1, 4–7 to 4–8, 8–3t

- list of, *CALL*, 4–7

Data types, *INTRO*, 3–2 to 3–3

- COMP (INTEGER), *REF*, 5–274
- COMP-1 (REAL), *REF*, 5–274
- COMP-2 (DOUBLE), *REF*, 5–274
- COMP-3 (PACKED), *REF*, 5–274
- COMP-5 (ZONED), *REF*, 5–274
- DATE, *REF*, 5–273

Date

- arithmetic, *REF*, 2–10
- edit string, *REF*, 2–11

DATE data type, *REF*, 5–273

Date fields, *UG*, 5–13

- edit string characters, *REF*, 5–115

- Date functions, *REF*, 6–2
- Dates
 - suppressing, *REPORT*, 2–11
- Date string function (FN\$DATE), *REF*, 6–5
- Day function (FN\$DAY), *REF*, 6–6
- DB (debit)
 - edit string character, *REF*, 5–111
- DDMF
 - See Distributed Server*
- DDMF.TSK, *CALL*, 1–2
- Decimal point (.)
 - edit string character, *REF*, 5–112, 5–113
- DECLARE PORT statement, *REF*, 5–51 to 5–52; *CALL*, 4–23
- DECLARE statement, *UG*, 1–4; *REF*, 5–47 to 5–50
 - in BEGIN-END block, *REF*, 5–37, 5–48
 - variable-name, *UG*, 11–1
- Declaring variables, *UG*, 11–1
- DECnet
 - default accounts, *CALL*, 2–1
 - node specification, *CALL*, 2–1, 8–18
 - using the Remote Call Interface, *CALL*, 1–5
 - using the Remote Terminal Interface, *CALL*, 1–3
- DECREASING
 - in SORT statement, *REF*, 5–254
 - in SUM statement, *REF*, 5–266
 - sort key, *REF*, 3–8
- Default dictionary
 - QUERY.DIC, *UG*, 20–1
- DEFINE command, *UG*, 1–3
- DEFINE DICTIONARY command, *INTRO*, 3–10; *UG*, 3–2 to 3–4; *REF*, 5–53 to 5–55
- DEFINE DOMAIN command, *INTRO*, 3–10; *UG*, 1–3, 4–1 to 4–3; *REF*, 5–56 to 5–62
 - entered interactively, *UG*, 5–3
 - for RMS domains, *REF*, 5–57 to 5–59
 - for view domains, *REF*, 5–60 to 5–62
 - optional password, *UG*, 4–2
 - syntax, *UG*, 4–3
 - terminated by semicolon, *UG*, 4–2
 - usage rules, *UG*, 4–2
- DEFINE FILE command, *INTRO*, 3–10; *UG*, 1–2, 6–3 to 6–10; *REF*, 5–63 to 5–68, 5–200
 - optional clauses, *UG*, 6–7 to 6–10
 - syntax, *UG*, 6–3
 - used with sequential files, *UG*, 6–4
- DEFINER command, *UG*, 1–3; *REF*, 5–82 to 5–85
- DEFINE PORT command, *REF*, 5–69 to 5–70; *CALL*, 4–23
- DEFINE PROCEDURE command, *UG*, 1–4, 9–2; *REPORT*, 1–6; *REF*, 5–71 to 5–73
- DEFINE RECORD command, *INTRO*, 3–10; *REF*, 4–1, 5–74 to 5–77, 5–199
 - advantage over ADT, *UG*, 5–1
- DEFINE TABLE command, *UG*, 12–3; *REF*, 5–78 to 5–80
- Defining
 - alternate keys, *UG*, 6–6
 - data files, *REF*, 5–63 to 5–68
 - dictionaries, *REF*, 5–53 to 5–55
 - domains, *UG*, 1–3, 4–1; *REF*, 1–11, 5–56 to 5–62
 - RMS, *REF*, 5–57 to 5–59
 - view, *REF*, 5–60 to 5–62
 - files, *REF*, 1–11
 - global variables, *REF*, 2–9
 - local variables, *REF*, 2–9
 - ports, *REF*, 5–69 to 5–70
 - privileges, *REF*, 5–82 to 5–85
 - procedures, *REF*, 1–7, 5–71 to 5–73
 - records, *UG*, 5–1 to 5–22; *REF*, 1–11, 4–1 to 4–10, 5–74 to 5–77
 - tables, *REF*, 1–12, 5–78 to 5–80
- Defining data
 - outside the Application Design Tool, *INTRO*, 3–10 to 3–11
 - with Application Design Tool, *INTRO*, 3–1 to 3–10
- Definition prompt, *REF*, 1–9
- DELETE command, *UG*, 1–3; *REF*, 5–86 to 5–88
 - removing data dictionary definition, *UG*, 20–4
 - terminated by semicolon, *UG*, 20–4
- DELETEP command, *UG*, 1–3; *REF*, 5–89 to 5–90
- Deleting
 - dictionaries, *REF*, 5–86
 - dictionary objects, *REF*, 5–86 to 5–88
 - domains, *REF*, 5–86
 - files with SUPERSEDE clause, *REF*, 5–64
 - procedures, *REF*, 5–86
 - records, *REF*, 5–86
 - records in indexed files, *REF*, 5–67
 - tables, *REF*, 5–86
- DESCENDING
 - in SORT statement, *REF*, 5–254
 - in SUM statement, *REF*, 5–266
 - sort key, *INTRO*, 6–9, 6–10; *REF*, 3–8
- Detail lines, *REPORT*, 2–12 to 2–19
 - content, *REPORT*, 2–13 to 2–14
 - format, *REPORT*, 2–14 to 2–17
- DFN> prompt, *UG*, 2–5; *REF*, 1–9
- Dictionary
 - See Data dictionary*

DICTIONARY

in SHOW command, *REF*, 5-247

Dictionary objects

controlling access to, *UG*, 19-1 to 19-14

editing, *REF*, 1-13

extracting definitions, *REF*, 5-125

extracting with QXTR, *REF*, 5-127

Dictionary tables

See Tables

Disk space

conserving with QCPRS utility, *UG*, 20-4 to 20-6

Displaying

DATATRIEVE objects, *INTRO*, 2-3 to 2-4

See also SHOW commands

dictionary name, *INTRO*, 2-2

established collections, *UG*, 20-2

readied domains, *UG*, 20-2

Displaying reports, *REPORT*, 1-4

Displaying variables, *REF*, 5-50

DISPLAY statement, *REF*, 5-91 to 5-93

with FILLER fields, *REF*, 4-5

Distributed Data Manipulation Facility

See Distributed Server

Distributed Server, *REF*, 1-14; *CALL*, 1-2, 1-3, 1-5f

function and use of, *UG*, 1-6 to 1-8

using with the Remote Call Interface, *CALL*, 1-5

Dollar sign (\$)

edit string character, *REF*, 5-113

floating edit string character, *REF*, 5-114

Domains

access all records, *UG*, 7-2

accessing, *REF*, 5-194 to 5-201

alternative names, *REF*, 5-194

as record source, *REF*, 3-2

copying, *CALL*, 2-3 to 2-4

defining, *UG*, 1-2, 3-1, 4-1; *REF*, 1-11, 5-56 to 5-62

RMS, *REF*, 5-57 to 5-59

view, *REF*, 5-60 to 5-62

deleting, *REF*, 5-86

editing, *REF*, 5-100 to 5-101

hierarchical, *REF*, 5-164 to 5-169

modifying, *REF*, 5-59

planning, *INTRO*, 3-2 to 3-4

restructuring, *UG*, 15-1 to 15-10

examples, *UG*, 15-2 to 15-10

rules for naming, *UG*, 4-1

sample, *INTRO*, 1-3; *UG*, 2-2

showing, *REF*, 5-247

views, *UG*, 13-1 to 13-8

defining, *UG*, 13-2

DOUBLE data type

See COMP-2 data type

DROP statement, *UG*, 1-4; *REF*, 5-94 to 5-99

DTCLIB.OLB, DATATRIEVE object module library,
CALL, 3-9

DTCLIB.OLB library, *CALL*, 1-2, 1-5

DTCMD routine, *CALL*, 8-9

format, *CALL*, 4-13

substituting variables with, *CALL*, 4-13

DTCONT routine, *CALL*, 4-25, 8-12

DTFINI routine, *CALL*, 4-31, 8-14

DTGETP routine, *CALL*, 4-24, 8-15

DTINIT routine, *CALL*, 4-10, 4-12, 8-17

options, *CALL*, 8-18t

DTLINE routine, *CALL*, 4-15, 8-21

DTMSG routine, *CALL*, 4-16, 8-22

DTPEOF routine, *CALL*, 4-26, 8-24

DTPUTP routine, *CALL*, 4-26, 8-26

DTPVAL routine, *CALL*, 8-28

DTR.TSK, *UG*, 1-6; *CALL*, 1-2

DTR> prompt, *UG*, 2-5; *REF*, 1-9

DTUNWD routine, *CALL*, 4-31, 8-30

DUP, *UG*, 6-9

in DEFINE FILE command, *REF*, 5-65

E

EDIT

CHANGE command, *UG*, 16-4

EDIT command, *UG*, 1-3, 10-3; *REF*, 5-100 to 5-101

Editing dictionary objects, *REF*, 1-13, 5-100 to 5-101, 5-126

Editor, *UG*, 16-1 to 16-5

changing record definitions, *UG*, 5-3

editing procedures, *UG*, 9-15

EDT

prompt, *REF*, 5-101

ending an editing session, *REF*, 5-101

invoking, *UG*, 16-2 to 16-4

line pointer, *UG*, 16-4

modes, *UG*, 16-4 to 16-5

range specifiers, *UG*, 16-5

summary, *REF*, 5-101

use of, *REF*, 1-13

Edit string characters, *REF*, 5-104t

alphanumeric insertion, *REF*, 5-108

alphanumeric replacement, *REF*, 5-108

asterisk (*), *REF*, 5-109

B (space), *REF*, 5-109

comma (,), *REF*, 5-113

Edit string characters (cont'd.)

- CR (credit), *REF*, 5-111
- date fields, *REF*, 5-114 to 5-119
- DB (debit), *REF*, 5-111
- decimal point (.), *REF*, 5-112, 5-113
- dollar sign (\$), *REF*, 5-113
- floating, *REF*, 5-114
- floating dollar sign, *REF*, 5-114
- floating minus sign, *REF*, 5-114
- floating plus sign, *REF*, 5-114
- for alphanumeric fields, *REF*, 5-108 to 5-109
- hyphen (-), *REF*, 5-109
- J (Julian date), *REF*, 5-107
- M (month letter), *REF*, 5-106
- minus sign (-), *REF*, 5-105
- N (month number), *REF*, 5-107
- numeric fields, *REF*, 5-109 to 5-119
- numeric insertion, *REF*, 5-111
- numeric replacement, *REF*, 5-109
- percent sign (%), *REF*, 5-113
- plus sign (+), *REF*, 5-111
- slash (/), *REF*, 5-107, 5-113
- T (text), *REF*, 5-108
- W (day letter), *REF*, 5-107
- X (alphanumeric), *REF*, 5-108
- Y (year), *REF*, 5-107
- Z, *REF*, 5-109
- zero (0), *REF*, 5-113

Edit strings, *REPORT*, 2-16 to 2-17

- date, *REF*, 2-11
- for concatenated expressions, *REF*, 2-18
- in PRINT statement, *REF*, 5-183
- in RW PRINT statement, *REF*, 5-231
- text edit strings, *REPORT*, 2-17

EDIT_STRING clause, *REF*, 5-102 to 5-119

- in field definition, *UG*, 5-5; *REF*, 4-8

EDT

- Entering and exiting, *REF*, 1-13
- prompt, *REF*, 5-101

EDT prompt, *REF*, 1-10

Elementary fields, *REPORT*, 2-13

- alphanumeric, *REF*, 4-7
- assigning values to, *REF*, 5-24 to 5-28
- COMPUTED BY, *REF*, 4-7
- DATE, *REF*, 4-7
- defined, *UG*, 5-6 to 5-7; *REF*, 1-11
- in field definition, *REF*, 4-2
- names, *REF*, 2-4
- numeric, *REF*, 4-7

ELSE clause

- in DEFINE TABLE command, *REF*, 5-79

EMPLOYEE_REC

- valid field names, *UG*, 5-8F

Ending a DATATRIEVE session, *INTRO*, 2-5

Ending the DATATRIEVE session, DTFINI, *CALL*, 4-31, 8-14

End-of-file marker, passing, DTPEOF, *CALL*, 4-26, 8-24

END_PROCEDURE clause, *UG*, 9-2; *REF*, 5-71

END_REPORT statement (Report Writer), *UG*, 2-4; *REPORT*, 1-4, 1-5, 1-7; *REF*, 5-228

END_TABLE clause, *UG*, 12-3

- in DEFINE TABLE command, *REF*, 5-79

EQUAL relational operator, *INTRO*, 6-6; *UG*, 7-5; *REF*, 2-23

Equal sign

- in assignment statements, *REF*, 5-25, 5-29, 5-32

ERASE statement, *UG*, 1-4; *REF*, 5-120 to 5-122

- access mode required, *REF*, 5-198t
- restrictions, *UG*, 6-2

Erasing data, *INTRO*, 8-1 to 8-4

Error messages, *UG*, 2-6

- incorrectly named fields, *UG*, 5-8
- located by a procedure, *UG*, 9-6 to 9-7

Error messages, retrieving, DTMSG, *CALL*, 4-16, 8-22

Error severity codes, DAB\$W_ERR_SEV, *CALL*, 4-17, 8-4

Error status codes, DAB\$W_ERR_CODE, *CALL*, 4-17, 8-4

Evaluating

- arithmetic expressions, *REF*, 2-16
- Boolean expressions, *REF*, 2-26, 2-27t

Event Flag Numbers, *CALL*, 3-9

Exclamation point (!), *REF*, 1-7

- in command files, *REF*, 5-12

EXCLUSIVE access option, *REF*, 5-195

Executing DCL command from DATATRIEVE (FN\$DCL), *REF*, 6-7

EXIT command, *INTRO*, 2-5; *UG*, 1-3, 2-6; *REF*, 5-123; *CALL*, 2-2

Exiting

- an ADT session, *INTRO*, 3-6
- DATATRIEVE, *INTRO*, 2-5

Exiting DATATRIEVE, *UG*, 2-6 to 2-7

Expressions

- arithmetic, *REF*, 2-15 to 2-17
- Boolean, *REF*, 2-1, 2-19 to 2-27
- concatenated, *REF*, 2-2, 2-17 to 2-19
- DATE value, *REF*, 2-10
- evaluating arithmetic, *REF*, 2-16
- evaluating Boolean, *REF*, 2-26, 2-27t

Expressions (cont'd.)

- prompting value, *REF*, 2-11 to 2-12
- statistical, *REF*, 2-13 to 2-15
- table values in, *REF*, 2-12 to 2-13
- value, *REF*, 2-1, 2-1 to 2-19

EXTEND access mode, *REF*, 5-196

EXTRACT command, *UG*, 1-3, 9-15, 20-3 to 20-4;
REF, 5-125 to 5-129; *CALL*, 2-4

- copying record definitions, *UG*, 5-3

Extracting

- day part of input, *REF*, 6-6
- hour part of input, *REF*, 6-9
- hundredth-of-a-second part of input, *REF*, 6-10
- minute part of input, *REF*, 6-12
- month part of input, *REF*, 6-14
- second (time) part of input, *REF*, 6-15
- substring from input, *REF*, 6-17
- time part of input, *REF*, 6-19
- year part of input, *REF*, 6-21

Extracting data dictionary information, *UG*, 20-6

F

FAMILIES sample domain, *UG*, 2-2

- using list field, *UG*, 5-16
- using SHOWP, *UG*, 20-2

Field classes, *REF*, 4-7, 4-7t

Field definitions, *UG*, 5-4 to 5-14

- clauses, *UG*, 5-12 to 5-14
- clauses in, *UG*, 1-2; *REF*, 4-8 to 4-10
- COMP (INTEGER), *REF*, 5-274
- COMP-1 (REAL), *REF*, 5-274
- COMP-2 (DOUBLE), *REF*, 5-274
- COMP-3 (PACKED), *REF*, 5-274
- COMP-5 (ZONED), *REF*, 5-274
- components of, *REF*, 4-1
- EDIT_STRING clause, *UG*, 5-5
- elementary fields, *UG*, 5-6 to 5-7
- group fields, *UG*, 5-6 to 5-7
- level numbers, *UG*, 5-4 to 5-7
- level numbers in, *REF*, 4-2 to 4-4
- naming fields, *UG*, 5-7
- PICTURE clause, *UG*, 5-5
- rules for writing, *UG*, 5-4 to 5-14; *REF*, 4-9 to 4-10
- specifying field characteristics, *REF*, 4-8
- terminated by period (.), *UG*, 5-4
- valid field names, *UG*, 5-8F
- word boundary alignment, *REF*, 5-22

Field names

- as value expressions, *REF*, 2-3

Field names (cont'd.)

- COMPUTED BY clause, *REF*, 2-7
- duplicate, *UG*, 5-9 to 5-10
- elementary, *REF*, 2-4
- FILLER, *UG*, 5-10 to 5-12; *REF*, 4-5 to 4-7
- group, *REF*, 2-4 to 2-7
- in AT BOTTOM statement (RW), *REF*, 5-221
- in AT TOP statement (RW), *REF*, 5-225
- in field definition, *REF*, 4-1, 4-4 to 4-7
- qualified, *REF*, 2-3 to 2-8
- qualifying, *REF*, 3-6
- REDEFINES, *REF*, 2-4
- restrictions, *UG*, 5-8 to 5-9
- rules for, *REF*, 4-4
- use of hyphens or underscores, *UG*, 1-2
- using as value expressions, *REF*, 2-3
- using query names, *REF*, 2-7

Fields

- alphanumeric, *UG*, 5-12
- Computed by, *UG*, 5-13
- COMPUTED BY, *REF*, 2-7, 5-25, 5-44 to 5-46
- data types, *INTRO*, 3-2 to 3-3; *REF*, 5-274
- date, *UG*, 5-13; *REF*, 5-274
- definition clauses, *UG*, 5-12 to 5-14
- PICTURE, *UG*, 5-12
- USAGE, *UG*, 5-13 to 5-14
- elementary, *REF*, 1-11, 2-4, 4-2
- elementary and group, *UG*, 1-2
- formatting for output, *REF*, 5-102 to 5-119
- formatting storage, *REF*, 5-174 to 5-177, 5-272 to 5-275
- group, *REF*, 1-11, 4-2
- key, *REF*, 5-64
- KEY attributes, *REF*, 5-65t
- KEY defaults, *REF*, 5-65t
- level numbers of, *REF*, 4-3
- list, *UG*, 5-16
- masking, *REF*, 4-5
- multiple occurrences, *REF*, 5-164 to 5-169
- naming, *UG*, 5-7
 - using the ADT command, *INTRO*, 3-2
- numeric, *UG*, 5-13
- query names, *INTRO*, 3-2
- redefining, *REF*, 5-204 to 5-206
- showing, *REF*, 5-247
- size, *INTRO*, 3-3
- specifying types of data, *UG*, 5-12 to 5-14

Field values

- in detail lines, *REPORT*, 2-13

Files

- allocating storage, *REF*, 5-21 to 5-23

Files (cont'd.)

- changing structure of, *UG*, 15–1 to 15–10
- closing log, *REF*, 5–42 to 5–43, 5–171
- comparison of sequential and indexed, *UG*, 6–2T
- creating, *REF*, 5–63 to 5–68
- defining, *UG*, 1–2, 6–1 to 6–10; *REF*, 1–11, 5–63 to 5–68
- indexed, *INTRO*, 3–3 to 3–4, 7–7; *UG*, 6–1 to 6–10; *REF*, 1–11
 - criteria for using, *UG*, 6–2
 - defining, *UG*, 6–4 to 6–5
- opening log, *REF*, 5–170
- organizing, *REF*, 1–11
- sequential, *INTRO*, 3–3 to 3–4; *UG*, 6–1 to 6–9; *REF*, 1–11
 - criteria for using, *UG*, 6–2
 - defining, *UG*, 6–3 to 6–4
- with fixed-length records, *REF*, 5–64
- FILLER fields, *REF*, 4–5 to 4–7
 - accessing data in, *REF*, 4–5
 - as group field name, *UG*, 5–11
 - defining, *REF*, 5–75
 - displaying with DISPLAY, *REF*, 4–5
- FIND statement, *INTRO*, 6–13; *UG*, 1–4; *REPORT*, 2–2 to 2–3; *REF*, 5–130 to 5–132
 - access mode required, *REF*, 5–198t
 - context variable in, *REF*, 3–6
- FINISH command, *INTRO*, 6–24 to 6–25; *UG*, 1–3; *REPORT*, 1–7; *REF*, 5–133 to 5–136
 - releasing domains, *REF*, 5–199
- FIRST clause
 - in SELECT, *REF*, 5–236
- FIRST n clause
 - in RSE, *UG*, 7–3; *REF*, 3–5
- Flat records, *UG*, 5–16F
- FN\$ABS function, *REF*, 6–4
- FN\$DATE function, *REF*, 6–5
- FN\$DAY function, *REF*, 6–6
- FN\$DCL function, *REF*, 6–7
- FN\$HOUR function, *REF*, 6–9
- FN\$HUNDREDTH function, *REF*, 6–10
- FN\$JULIAN function, *REF*, 6–11
- FN\$MINUTE function, *REF*, 6–12
- FN\$MOD function, *REF*, 6–13
- FN\$MONTH function, *REF*, 6–14
- FN\$SECOND function, *REF*, 6–15
- FN\$SIGN function, *REF*, 6–16
- FN\$STR_EXTRACT, *REF*, 6–17
- FN\$STR_LOC function, *REF*, 6–18
- FN\$TIME function, *REF*, 6–19
- FN\$UPCASE function, *REF*, 6–20

FN\$YEAR function, *REF*, 6–21

Formatting output

See also Edit string characters

- date fields, *REF*, 5–114
- default, *REPORT*, 2–2, 2–5
- fields, *REF*, 5–102 to 5–119
- report header, *REPORT*, 2–8 to 2–9
- reports, *REPORT*, 1–4
- with PRINT, *REF*, 5–178 to 5–189
- with the Report Writer, *REF*, 1–13, 5–230

Form feed

trailing, *REPORT*, 2–3

FOR statement, *UG*, 8–3; *REPORT*, 3–18 to 3–19; *REF*, 5–137 to 5–141

controlling PRINT output, *REF*, 5–184

creating target record streams, *UG*, A–26 to A–28

declaring variables in, *REF*, 5–48

invoking procedures with, *REF*, 5–9

to form record streams, *REF*, 3–3

with STORE statement, *REF*, 5–261

FORTRAN sample programs, *CALL*, 5–1 to 5–18

FROM clause, *UG*, 13–2

in DEFINE DOMAIN command, *REF*, 5–61

Functions

alphanumeric, *REF*, 6–2

date, *REF*, 6–2

FN\$ABS, *REF*, 6–4

FN\$DATE, *REF*, 6–5

FN\$DAY, *REF*, 6–6

FN\$DCL, *REF*, 6–7

FN\$HOUR, *REF*, 6–9

FN\$HUNDREDTH, *REF*, 6–10

FN\$JULIAN, *REF*, 6–11

FN\$MINUTE, *REF*, 6–12

FN\$MOD, *REF*, 6–13

FN\$MONTH, *REF*, 6–14

FN\$SECOND, *REF*, 6–15

FN\$SIGN, *REF*, 6–16

FN\$STR_EXTRACT, *REF*, 6–17

FN\$STR_LOC, *REF*, 6–18

FN\$TIME, *REF*, 6–19

FN\$UPCASE, *REF*, 6–20

FN\$YEAR, *REF*, 6–21

format of statistical, *REF*, 2–13

listed alphabetically, *REF*, 6–3

numeric, *REF*, 6–2

statistical, *REF*, 2–13 to 2–15

using statistical, *REF*, 2–13

value expressions, *REF*, 6–1

values derived with statistical, *REF*, 2–13t

Index–10

G

GE

See *GREATER_EQUAL relational operator*

Global variables, *UG*, 11–4

named in context block, *UG*, A–4

GREATER_EQUAL relational operator, *INTRO*, 6–6;
UG, 7–6; *REF*, 2–21

GREATER_THAN relational operator, *INTRO*, 6–6;
UG, 7–6; *REF*, 2–20

Group fields, *REPORT*, 2–13

alphanumeric, *REF*, 4–7

as primary key, *UG*, 6–5 to 6–6

assigning values to, *REF*, 5–29 to 5–31

defined, *UG*, 5–6 to 5–7; *REF*, 1–11

in assignment statements, *REF*, 2–4 to 2–7

in field definition, *REF*, 4–2

qualifying, *REF*, 2–7

using FILLER field, *UG*, 5–11

GT

See *GREATER_THAN relational operator*

Guide Mode, *INTRO*, 4–3 to 4–4; *UG*, 2–3, 2–8 to 2–9

invoking, *UG*, 2–9

using a question mark (?), *UG*, 2–9

H

Halting execution

with ABORT, *REF*, 5–14 to 5–18

HELP command, *INTRO*, 4–1 to 4–3; *UG*, 1–3, 2–7 to 2–8; *REF*, 5–142 to 5–145

ADVANCED HELP, *UG*, 2–8

Hierarchical domains, *REF*, 5–164 to 5–169

Hierarchical records, *REPORT*, 3–16 to 3–21

Hierarchies, *UG*, 14–1 to 14–13

defining, *UG*, 5–16 to 5–22

eliminating empty print lines, *UG*, 14–9

retrieving values

sublists, *UG*, 14–11 to 14–13

using ALL in nested print lists, *UG*, 14–6

using FIND and SELECT statements, *UG*, 14–3

using nested RSEs, *UG*, 14–9

using OF rse clause, *UG*, 14–5

with nested FOR loops, *UG*, 14–9

saving space, *UG*, 5–16

using, *UG*, 5–17 to 5–22

Hour function (FN\$HOUR), *REF*, 6–9

Hundredth of second function (FN\$HUNDREDTH),
REF, 6–10

Hyphen (-), *REPORT*, 2–17, 2–18

as a minus sign, *REF*, 1–5

continuation character, *UG*, 5–8; *REF*, 1–7

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

conversion to underscore (.), *REF*, 1–5

edit string character, *REF*, 5–109

in record name, *UG*, 5–4

in string literals, *REF*, 2–2

I

IF-THEN-ELSE statement, *UG*, 8–5; *REF*, 5–146 to 5–149

using with ABORT statement, *REF*, 5–16

Inclusion files, DAB, *CALL*, 4–10

INCREASING

in SORT statement, *REF*, 5–254

in SUM statement, *REF*, 5–266

sort key, *REF*, 3–8

Indexed files, *INTRO*, 3–3 to 3–4; *UG*, 6–1 to 6–10

compared with sequential, *UG*, 6–2T

compressing, *UG*, 20–7

creating, *REF*, 5–63, 5–66

defining, *UG*, 6–4 to 6–5

deleting records, *REF*, 5–67

ERASE statement with, *INTRO*, 8–1 to 8–4

KEY field attributes, *REF*, 5–65t

KEY field defaults, *REF*, 5–65t

keys, *REF*, 1–11

modifying, *INTRO*, 7–7 to 7–9

modifying records in, *REF*, 5–67

multikey, *UG*, 6–3

optimizing storage, *UG*, 20–7

Information management, *INTRO*, 1–1

Initializing

global variables, *REF*, 5–47

local variables, *REF*, 5–37, 5–48

variables, *REF*, 2–8

Initializing the Call Interface, *CALL*, 4–11, 4–12, 8–17

Input line prompt, *UG*, 2–5

IN relational operator, *UG*, 12–5; *REF*, 2–22, 2–24

Installation kit, *UG*, 2–2

INTEGER data type

See *COMP data type*

Interactive DATATRIEVE, *CALL*, 1–2

Interactive DATATRIEVE–11, *UG*, 1–6

Internal storage, *REF*, 5–272 to 5–275

@ (Invoke Command File) Command

See *AT*

: (Invoke procedure)

See Colon

Invoking

an ADT session, *INTRO*, 3–6

command files, *REF*, 5–11 to 5–13

DATATRIEVE, *INTRO*, 2–1; *UG*, 2–1 to 2–2

procedures, *REPORT*, 1–7; *REF*, 1–8, 5–7 to 5–10, 5–72

tables, *REF*, 5–80

J

J (Julian date)

edit string character, *REF*, 5–107

Julian date function (FN\$JULIAN), *REF*, 6–11

K

KEY clause, *UG*, 6–4

in DEFINE FILE command, *REF*, 5–64

KEY field

attributes, *REF*, 5–65t

defaults, *REF*, 5–65t

duplicate values in, *REF*, 5–65

modifying, *REF*, 5–64

primary, *REF*, 5–64

Keys

accessing dictionary objects, *UG*, 19–2

alternate index, *INTRO*, 3–4

defining alternate keys, *UG*, 6–6

defining key fields

rules for, *UG*, 6–7

duplicate values, *INTRO*, 3–4

primary index, *INTRO*, 3–4

Keywords, *REF*, 1–5

L

LAST

in SELECT, *REF*, 5–236

LCDDMF.TSK, *CALL*, 1–2

LE

See LESS_EQUAL relational operator

LEFT_RIGHT allocation, *REF*, 5–21; *CALL*, 2–3

LESS_EQUAL relational operator, *INTRO*, 6–6; *UG*, 7–6; *REF*, 2–23

LESS_THAN relational operator, *INTRO*, 6–6; *UG*, 7–6; *REF*, 2–23

Letters

in string literals, *REF*, 2–3

in table strings, *REF*, 5–79

Letters (cont'd.)

treatment of, *REF*, 1–4

treatment with CONTAINING, *REF*, 2–23

Level numbers

in field definition, *REF*, 4–1, 4–2 to 4–4

rules for assigning, *REF*, 4–3

valid values, *REF*, 5–75

Line pointer, *UG*, 16–4

Lists, *REF*, 2–24

as record source, *REF*, 3–3

changing length of, *UG*, 5–21 to 5–22

creating with OCCURS FOR, *REF*, 5–61

defining fixed occurrences, *UG*, 5–18 to 5–19

defining variable occurrences, *UG*, 5–19 to 5–20

defining with OCCURS clause, *UG*, 5–16 to 5–22; *REF*, 5–164 to 5–169

modifying, *REF*, 3–4

nesting to form sublists, *UG*, 5–21 to 5–22

printing, *REPORT*, 3–19 to 3–21

Lists, in records

See Hierarchies

Literals, *REPORT*, 2–14; *REF*, 2–1 to 2–3

character string, *REF*, 2–1 to 2–3

hyphens in, *REF*, 2–2

in detail lines, *REPORT*, 2–13

numeric, *REF*, 2–3

printing, *REF*, 5–180

Local Call Interface, *CALL*, 1–1, 8–18

Local Server, *CALL*, 1–2

Local variables, *UG*, 11–4 to 11–5

effect on context stack, *UG*, A–8

Lock types, *UG*, 19–2

Log files

closing, *REF*, 5–171

opening, *REF*, 5–170

Logical Unit Numbers

allocating, *CALL*, 3–9

Loops

ABORT statement in, *REF*, 5–14

FOR statement, *REF*, 1–3, 5–137 to 5–141

REPEAT statement, *REF*, 1–3, 5–212

WHILE statement, *REF*, 1–3, 5–278 to 5–280

Lowercase letters

conversion of, *REF*, 1–4

in string literals, *REF*, 2–3

LT

See LESS_THAN relational operator

LUNMAP, area for LUN specification, *CALL*, 3–10

LUNs

See Logical Unit Numbers

M

M (month letter)
 edit string character, *REF*, 5-106
MACRO-11 assembly language, *CALL*, 8-8
MAJOR_MINOR allocation, *REF*, 5-21; *CALL*, 2-3
MAX clause, *UG*, 5-21, 6-8
 in DEFINE FILE command, *REF*, 5-64
MAX statistical function, *REPORT*, 2-21 to 2-22;
 REF, 2-13
 in AT BOTTOM statement (RW), *REF*, 5-221
 in AT TOP statement (RW), *REF*, 5-225
Memory, conserving
 See Optimizing workspace
Menu interface, example, *CALL*, 5-1
Menus
 creating with the Call Interface, *CALL*, 1-7
Messages, obtaining, DTMSG, *CALL*, 4-16, 8-22
MIN statistical function, *REPORT*, 2-21 to 2-22;
 REF, 2-13
 in AT BOTTOM statement (RW), *REF*, 5-221
 in AT TOP statement (RW), *REF*, 5-226
Minus sign (-)
 edit string character, *REF*, 5-105
 floating edit string character, *REF*, 5-114
 in arithmetic expressions, *REF*, 2-16
Minute function (FN\$MINUTE), *REF*, 6-12
MODIFY access mode, *REF*, 5-196
Modifying
 access control list, *REF*, 5-82 to 5-85
 data, *INTRO*, 7-1 to 7-11
 domains, *REF*, 5-59
 key fields, *REF*, 5-64
 procedures, *REF*, 5-72
 record definitions, *REF*, 5-76
 records in indexed files, *REF*, 5-67
 records in sequential files, *REF*, 5-67
 selected record, *INTRO*, 7-4 to 7-7
 tables, *REF*, 5-80
MODIFY statement, *UG*, 1-4, 2-6; *REF*, 5-150 to
 5-163
 access mode required, *REF*, 5-198t
 ALL, *INTRO*, 7-9 to 7-10
 assigning field values with, *REF*, 5-25, 5-29
 changing fields, *UG*, 6-2
 effect of ABORT statement, *REF*, 5-15
 preventing, *INTRO*, 7-5 to 7-6
 VERIFY clause, *REF*, 5-260
 with lists, *REF*, 3-4
Modulus function (FN\$MOD), *REF*, 6-13
Month function (FN\$MONTH), *REF*, 6-14

N

N (month number)
 edit string character, *REF*, 5-107
Names, *REF*, 1-6
 continuing, *REF*, 1-6
 rules for, *REF*, 1-6
Naming fields
 using the ADT command, *INTRO*, 3-2
Naming reports, *REPORT*, 1-4
NE
 See NOT_EQUAL relational operator
NEW_PAGE, *REPORT*, 3-11, 3-14
 in AT BOTTOM statement (RW), *REF*, 5-222t
 in AT TOP statement (RW), *REF*, 5-226t
 print list element, *REF*, 5-182t
 RW print list element, *REF*, 5-230t
NEW_SECTION
 in AT BOTTOM statement (RW), *REF*, 5-222t
 in AT TOP statement (RW), *REF*, 5-226t
 RW print list element, *REF*, 5-230t
NEXT
 in SELECT, *REF*, 5-236, 5-237
9 (numeric)
 edit string character, *REF*, 5-109
 picture string character, *REF*, 5-175
NO CHANGE, *UG*, 6-9
 in DEFINE FILE command, *REF*, 5-64
Node specification, *CALL*, 2-1, 8-18
NO DUP, *UG*, 6-9
 in DEFINE FILE command, *REF*, 5-65
NOT BETWEEN relational operator, *REF*, 2-22
NOT Boolean operator, *INTRO*, 6-6; *UG*, 7-9; *REF*,
 2-25
NOT BT
 See NOT BETWEEN relational operator
NOT CONT
 See NOT CONTAINING relational operator
NOT CONTAINING relational operator, *REF*, 2-23
NOT EQUAL relational operator, *UG*, 7-5
NOT IN relational operator, *UG*, 12-5; *REF*, 2-22
NOT_EQUAL relational operator, *INTRO*, 6-6; *REF*,
 2-20
Numeric fields, *UG*, 5-13
 edit string characters, *REF*, 5-109 to 5-119
 picture string characters, *REF*, 5-175
Numeric functions, *REF*, 6-2
Numeric literals
 See Literals

O

Object module libraries, *CALL*, 3–9

Obtaining messages, *DTMSG*, *CALL*, 4–16, 8–22

Obtaining print lines, *DTLINE*, *CALL*, 4–15, 8–21

Obtaining records from *DATATRIEVE*, *DTGETP*, *CALL*, 4–24, 8–15

OCCURS clause, *UG*, 5–16 to 5–22; *REPORT*, 3–16; *REF*, 5–164 to 5–169

- changing list length, *UG*, 5–21
- defining hierarchical records, *UG*, 5–17
- fixed number of occurrences, *UG*, 5–18 to 5–19; *REF*, 5–165 to 5–167
- in field definition, *REF*, 4–8
- variable number of occurrences, *UG*, 5–19 to 5–20; *REF*, 5–168 to 5–169

OCCURS FOR clause, *UG*, 13–2

- in *DEFINE DOMAIN* command, *REF*, 5–61

OF *rse* clause

- targeting record streams, *UG*, A–24 to A–25

OPEN command, *REF*, 5–170 to 5–173

Operating systems

- for *DATATRIEVE*, *UG*, 1–1

Operators

- arithmetic, *REF*, 2–15, 2–16t
- Boolean, *REF*, 2–25 to 2–27
- relational, *REF*, 2–19 to 2–25

Optimizing

- response time, *UG*, 17–7 to 17–11
- SHOW SPACE* command, *REF*, 5–247
- with keyed access, *UG*, 17–7 to 17–10
- workspace, *UG*, 17–5 to 17–7

Options, for *DTINIT*, *CALL*, 8–17, 8–18t

OR Boolean operator, *INTRO*, 6–6; *UG*, 7–9; *REF*, 2–25

Output

- controlling, *UG*, 18–1 to 18–5
- default settings, *UG*, 18–1

Overlays, *CALL*, 3–12

OWNERS sample domain, *UG*, 2–2

P

P (decimal scaling)

- picture string character, *REF*, 5–177

PACKED data type

- See COMP-3 data type*

Page format, *REPORT*, 2–5 to 2–8

- length, *REPORT*, 2–6
- width, *REPORT*, 1–6, 2–2, 2–5

Page numbers, *REPORT*, 2–8

Page numbers (cont'd.)

- suppressing, *REPORT*, 2–12

Parentheses

- in arithmetic expressions, *REF*, 2–16
- in Boolean expressions, *REF*, 2–26
- in statistical expressions, *REF*, 2–14

Passing command lines to *DATATRIEVE*, *DTCMD*, *CALL*, 4–13, 8–9

Passing end-of-file marker, *DTPEOF*, *CALL*, 4–26, 8–24

Passing records from *DATATRIEVE*, *DTGETP*, *CALL*, 8–15

Passing records to *DATATRIEVE*, *DTPUTP*, *CALL*, 4–26, 8–26

Passing values to *DATATRIEVE*, *DTPVAL*, *CALL*, 4–20, 8–28

Passwords

- keys, *UG*, 19–2
- specifying to the Remote Terminal Interface, *CALL*, 2–1

Percent sign (%)

- edit string character, *REF*, 5–113

Period

- in *DECLARE* statement, *REF*, 5–47
- in *DEFINE DOMAIN* clause, *REF*, 5–61
- in *DEFINE RECORD* command, *REF*, 5–75
- in field definition, *UG*, 5–4; *REF*, 4–1

PERSONNEL sample domain, *UG*, 2–2, 2–4

- record data items, *UG*, 5–1F

PERSONNEL_REC

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

PICTURE clause, *UG*, 5–12; *REF*, 5–174 to 5–177

- alphanumeric fields, *REF*, 5–175
- in *DECLARE* statement, *REF*, 5–47
- in field definition, *UG*, 5–5; *REF*, 4–8
- numeric fields, *REF*, 5–175
- restrictions, *REF*, 2–3

Picture string characters

- P* (decimal scaling), *REF*, 5–177
- S* (sign), *REF*, 5–176
- V* (decimal point), *REF*, 5–177
- X* (alphanumeric), *REF*, 5–175

Plus sign (+)

- edit string character, *REF*, 5–111
- floating edit string character, *REF*, 5–114

Pool space

- See Workspace*

Ports, *CALL*, 8–15

- declaring temporary, *REF*, 5–51 to 5–52
- defining, *REF*, 5–69 to 5–70

Ports (cont'd.)

- definition of, *CALL*, 4-22
- global, *REF*, 5-52
- local, *REF*, 5-52

Primary index key, *INTRO*, 3-4

Print items

- column position of, *REPORT*, 2-15 to 2-16
- cross tabulations, *REPORT*, 3-15
- edit string format, *REPORT*, 2-16 to 2-17
- order of, *REPORT*, 2-14 to 2-15

Print lines, obtaining, *DTLINE*, *CALL*, 4-15, 8-21

PRINT statement, *INTRO*, 6-1; *UG*, 1-4; *REF*, 5-178 to 5-189

- access mode required, *REF*, 5-198t
- creating reports with, *INTRO*, 9-1
- defaults, *REF*, 5-185
- displaying data with, *INTRO*, 6-6 to 6-12
- lists, *UG*, 14-9
- print list elements, *REF*, 5-181t
- print list modifiers, *REF*, 5-183t
- retrieving data, *UG*, 2-3 to 2-4

PRINT statement (Report Writer), *REPORT*, 1-3, 1-4, 2-1; *REF*, 5-229 to 5-232

- detail lines, *REPORT*, 2-12 to 2-13
- order of print items, *REPORT*, 2-15

Privileges

- access codes, *REF*, 5-83t
- assigning, *UG*, 19-11
- defining with *DEFINEP*, *REF*, 5-82 to 5-85
- showing, *REF*, 5-250 to 5-251

Procedures, *UG*, 1-4, 9-1 to 9-17

- aborting, *UG*, 9-13 to 9-14
- comments in, *UG*, 9-5 to 9-6
- compared with command files, *UG*, 10-1
- contents, *UG*, 3-1, 9-4 to 9-6
- defining, *UG*, 9-1 to 9-2; *REF*, 1-7, 5-71 to 5-73
- deleting, *UG*, 9-16 to 9-17; *REF*, 5-86
- displaying, *UG*, 9-15
- editing, *UG*, 9-15 to 9-16; *REF*, 5-100 to 5-101
- for reports, *REPORT*, 2-22
- including command files in, *REF*, 5-11
- in compound statements, *UG*, 9-11 to 9-13
- invoking, *UG*, 9-2 to 9-4; *REF*, 1-8, 5-7 to 5-10, 5-72
- locating errors, *UG*, 9-6 to 9-7
- maintaining, *UG*, 9-14 to 9-17
- modifying, *REF*, 5-72
- nesting, *UG*, 9-9 to 9-11; *REF*, 5-8
- REPEAT* statements in, *REF*, 5-213

Procedures (cont'd.)

- samples, *UG*, 9-7 to 9-9
- SET ABORT/SET NO ABORT*, *UG*, 9-13 to 9-14
- showing, *REF*, 5-247
- stopping execution of, *REF*, 5-14
- using with the Call Interface, *CALL*, 1-7

Prompting expressions, *CALL*, 4-20

Prompting value expressions, *REPORT*, 1-6 to 1-8, 2-14; *REF*, 2-11 to 2-12

- as print item, *REPORT*, 2-13
- forming, *REF*, 2-11
- for page width, *REPORT*, 2-6
- for report length, *REPORT*, 2-7
- for storing and modifying values, *UG*, 2-5; *REF*, 1-10
- for syntax, *REF*, 1-9
- in Boolean expressions, *REF*, 2-23
- in loops, *REF*, 2-11

Prompts, *REF*, 1-8 to 1-10

- CON>*, *UG*, 2-5; *REF*, 1-9
- DFN>*, *UG*, 2-5; *REF*, 1-9
- DTR>*, *UG*, 2-5; *REF*, 1-9
- EDT*, *REF*, 1-10, 1-13, 5-101
- REMDTR>*, *REF*, 1-14
- RW>*, *UG*, 2-5; *REPORT*, 1-5; *REF*, 1-9
- syntax, *UG*, 2-5

PROTECTED access option, *REF*, 5-195

Protecting dictionary tables, *UG*, 12-10

Q

QCPRS utility

- conserving disk space, *UG*, 20-4 to 20-6
- data dictionary compression, *UG*, 20-5
- defaults, *UG*, 20-5
- improving performance, *UG*, 20-5
- invoked by Digital Command Language, *UG*, 20-5

Qualified field names, *REF*, 2-7

QUERY.DIC default dictionary, *UG*, 20-1

QUERY.INI file, *UG*, 2-2

- sample, *UG*, 2-3
- SET DICTIONARY* in, *UG*, 2-3
- SET GUIDE* in, *UG*, 2-3

QUERYHEADER clause, *UG*, 5-12

Query names, *INTRO*, 3-2

- specifying, *UG*, 5-14 to 5-15
- using, *REF*, 2-7

QUERY_HEADER clause, *REF*, 5-190 to 5-191

- in field definition, *REF*, 4-4, 4-8

QUERY_NAME clause, *UG*, 5-12; *REF*, 5-192 to 5-193

QUERY_NAME clause (cont'd.)

in field definition, *REF*, 4-4, 4-8

Question mark (?)

used in Guide mode, *UG*, 2-9

Quotation marks

around string literals, *REF*, 2-2

in string literals, *REF*, 2-2

QXTR utility, *UG*, 20-7 to 20-8; *REF*, 5-127

creating command files, *UG*, 20-7

transferring data dictionary objects, *UG*, 20-7

R

READ access mode, *REF*, 5-196

READY

in SHOW command, *REF*, 5-247

READY command, *INTRO*, 2-4; *REPORT*, 2-2;

REF, 5-194 to 5-201

gaining access to domains, *UG*, 1-3

retrieving data, *UG*, 2-3 to 2-4

with PRINT statement, *REF*, 5-180

REAL data type

See COMP-1 data type

Record definition

changing, *UG*, 15-1 to 15-10

contents, *UG*, 3-1

field definitions within, *UG*, 5-4 to 5-14

modifying, *REF*, 5-76

OCCURS clause, *REF*, 5-164 to 5-169

PICTURE clause, *REF*, 5-174 to 5-177

QUERY_HEADER clause, *REF*, 5-190 to 5-191

QUERY_NAME clause, *REF*, 5-192 to 5-193

REDEFINES clause, *REF*, 5-204 to 5-206

SIGN clause, *REF*, 5-252 to 5-253

steps in writing, *UG*, 5-1 to 5-22

using OCCURS clause, *UG*, 5-17

VALID IF clause, *UG*, A-9; *REF*, 5-276 to 5-277

Record names

qualifying, *REF*, 2-7

Records

comparing, *UG*, 7-4 to 7-6

components of, *REF*, 1-11

defining, *UG*, 1-2, 5-1 to 5-22; *REF*, 1-11, 4-1 to 4-10, 5-74 to 5-77

field definitions within, *UG*, 5-4 to 5-14

field level numbers, *UG*, 5-5 to 5-7

specifying names, *UG*, 5-2 to 5-4

with variable length list, *UG*, 5-19 to 5-20

deleting, *UG*, 6-2; *REF*, 5-86

editing, *REF*, 5-100 to 5-101

Records (cont'd.)

establishing a single record context, *REF*, 5-237

example data items, *UG*, 5-1F

flat, *UG*, 5-16F

forming streams, *REF*, 3-1

grouping

by table reference, *UG*, 7-7

in range of values, *UG*, 7-6 to 7-7

hierarchical, *UG*, 5-17 to 5-22; *REF*, 2-24

limiting access, *UG*, 13-3

limiting fields in a, *UG*, 13-3

logical, *REF*, 4-1

modifying, *UG*, 6-2

passing to DATATRIEVE, DTPUTP, *CALL*, 4-26, 8-26

removing from collections, *REF*, 5-94 to 5-99

retrieving from DATATRIEVE, DTGETP, *CALL*, 4-24, 8-15

rules for naming, *UG*, 5-4

sample definition, *UG*, 5-2F

selecting, *REF*, 5-236 to 5-241

conditional expressions, *UG*, 7-4

showing, *REF*, 5-247

sorting, *REF*, 5-254 to 5-256

storing, *REF*, 5-257 to 5-265

transferring, *CALL*, 4-22 to 4-30

with missing values, *INTRO*, 5-3

Record selection expression, *INTRO*, 6-2 to 6-6; *UG*, 7-1 to 7-10

accessing records, *UG*, 7-1 to 7-2

ALL clause, *UG*, 7-2; *REF*, 3-5

Boolean expressions, *UG*, 7-4 to 7-6

context variable in, *REF*, 3-6

definition, *UG*, 7-1

FIRST n clause, *UG*, 7-1 to 7-3; *REF*, 3-5

format, *REF*, 3-1

limiting the number, *UG*, 7-3

record source, *REF*, 3-2 to 3-4

retrieving field values, *UG*, 5-11

SORTED BY clause, *UG*, 7-1, 7-10; *REF*, 3-5, 3-8

tables, *UG*, 7-7, 12-5

WITH clause, *UG*, 7-1, 7-4 to 7-10; *REF*, 3-7

Record stream, *INTRO*, 6-1; *UG*, 7-1

displaying fields, *INTRO*, 6-11 to 6-12

erasing, *INTRO*, 8-3, 8-4

forming, *INTRO*, 6-2 to 6-6; *REF*, 3-1

from collections, *REF*, 3-2

from domains, *REF*, 3-2

from lists, *REF*, 3-3

limiting, *INTRO*, 6-2 to 6-8

Record stream (cont'd.)

- modifying fields, *INTRO*, 7-10 to 7-11
- modifying records in, *INTRO*, 7-10 to 7-11
- naming, *REF*, 3-6
- sorting, *INTRO*, 6-8 to 6-11; *REF*, 3-8
 - by field values, *UG*, 7-10
- specifying conditions, *REF*, 3-7
- specifying number, *REF*, 3-5
- stopping display, *INTRO*, 6-12 to 6-13

Record streams

- context block, *UG*, A-6 to A-8

REDEFINE command, *REF*, 5-202 to 5-203

- changing domain definitions, *UG*, 15-2 to 15-10

REDEFINES clause, *UG*, 5-12; *REF*, 5-204 to 5-206

- in field definition, *REF*, 4-8

Redefining objects

- See* REDEFINE command, *REF*, 5-202

Relational operators, *INTRO*, 6-5 to 6-6; *UG*, 7-4 to 7-9; *REF*, 2-19 to 2-25

- summary of, *UG*, 7-8T

RELEASE command, *INTRO*, 6-23; *UG*, 1-3; *REF*, 5-207 to 5-210

Releasing

- domains and collections, *REF*, 5-133 to 5-136
- global variables, *REF*, 5-48, 5-50
- local variables, *REF*, 5-48

REMDTR, *CALL*, 1-2, 1-3

Remote Access Interface, *REF*, 1-14

Remote Call Interface

- See* Call Interface

Remote Terminal Interface, *UG*, 1-7 to 1-8; *REF*, 1-14; *CALL*, 1-1, 1-2

- advantages of, *CALL*, 1-4
- copying domains, *CALL*, 2-3 to 2-4
- example, *CALL*, 2-2, 2-4
- exiting, *CALL*, 2-2
- invoking, *CALL*, 2-1
- prompt, *REF*, 1-14
- specifying an account, *CALL*, 2-1
- specifying a node, *CALL*, 2-1
- testing DATATRIEVE, *CALL*, 2-2 to 2-3
- using REMDTR, *CALL*, 1-3

REPEAT statement, *UG*, 8-1 to 8-3; *REF*, 5-211 to 5-214

- BEGIN-END statement in, *REF*, 5-38
- declaring variables in, *REF*, 5-48
- invoking procedures with, *REF*, 5-8
- nesting in procedures, *REF*, 5-213
- with STORE statement, *REF*, 5-260

Report header, *REPORT*, 2-8 to 2-12

Reports, *INTRO*, 9-1 to 9-7

- cross tabulations, *REPORT*, 3-15 to 3-16
- dating, *REPORT*, 2-8, 2-10
- designing, *REPORT*, 2-1 to 2-24
- general features, *INTRO*, 9-1 to 9-2
- hierarchical records, *REPORT*, 3-16 to 3-19
- naming, *REPORT*, 2-1, 2-9
- output to disk file, *INTRO*, 9-5
- output to line printer, *INTRO*, 9-5
- page numbers, *REPORT*, 2-8
- periodic, *REPORT*, 1-1
- print list options, *INTRO*, 9-4
- query, *REPORT*, 1-1
- sample, *REPORT*, 1-2 to 1-3
- saving specifications, *INTRO*, 9-6
- special page headings, *REPORT*, 3-13 to 3-15
- specification, *INTRO*, 9-2 to 9-5; *REPORT*, 1-4 to 1-6, 2-1 to 2-2
- title page, *REPORT*, 3-11 to 3-13
- writing, *REF*, 5-215 to 5-219

REPORT statement, *UG*, 1-4; *REPORT*, 1-4, 1-5, 2-1, 2-3; *REF*, 5-215 to 5-219

- access mode required, *REF*, 5-198t
- identifying data, *REPORT*, 2-2
- retrieving data, *UG*, 2-3 to 2-4
- selecting output medium, *REPORT*, 2-22 to 2-24
 - default, *REPORT*, 2-22
 - file, *REPORT*, 2-23
 - printer, *REPORT*, 2-23
 - prompt option, *REPORT*, 2-23
 - terminal, *REPORT*, 2-23

Report Writer, *REF*, 1-13

- AT BOTTOM statement, *REF*, 5-220 to 5-223
- AT BOTTOM statement summary elements, *REF*, 5-221t
- AT TOP statement, *REF*, 5-224 to 5-227
- AT TOP statement header and summary elements, *REF*, 5-225t
- invoking, *REPORT*, 1-5, 2-2 to 2-3; *REF*, 5-215 to 5-219
- print list elements, *REF*, 5-230t
- print list modifiers, *REF*, 5-231t
- PRINT statement, *REF*, 5-229 to 5-232
- prompt, *REF*, 1-9
- SET statement, *REF*, 5-233 to 5-235

REPORT_HEADER, *REPORT*, 3-13, 3-14

- in AT TOP statement (RW), *REF*, 5-226t

Response time

- reducing with QCPRS utility, *UG*, 20-5

Restructuring domains

- examples, *UG*, 15-1 to 15-10

- Retrieving data, *INTRO*, 6-1 to 6-25; *UG*, 2-3 to 2-4
 - displaying all records, *INTRO*, 6-1
- Retrieving messages, *DTMSG*, *CALL*, 4-16, 8-22
- Retrieving print lines, *DTLINE*, *CALL*, 4-15, 8-21
- Retrieving records from DATATRIEVE, *DTGETP*, *CALL*, 4-24, 8-15
- RETURN key, *REF*, 1-7
- RMS domains
 - defining, *REF*, 5-57 to 5-59
- RMS facility
 - capabilities, *UG*, 6-1 to 6-3
- Routines, *CALL*, 8-7 to 8-30
 - DTCMD*, *CALL*, 4-13, 8-9
 - DTCONT*, *CALL*, 4-25, 8-12
 - DTFINI*, *CALL*, 4-31, 8-14
 - DTGETP*, *CALL*, 4-24, 8-15
 - DTINIT*, *CALL*, 4-12, 8-17
 - DTLINE*, *CALL*, 4-15, 8-21
 - DTMSG*, *CALL*, 4-16, 8-22
 - DTPEOF*, *CALL*, 4-26, 8-24
 - DTPUTP*, *CALL*, 4-26, 8-26
 - DTPVAL*, *CALL*, 4-20, 8-28
 - DTUNWD*, *CALL*, 4-31, 8-30
- RSE
 - See Record selection expression*
- Running programs that call DATATRIEVE, *CALL*, 3-1 to 3-12
- RW> prompt, *UG*, 2-5; *REF*, 1-9

S

- S (sign)
 - picture string characters, *REF*, 5-176
- Sample report, *REPORT*, 1-2 to 1-3
- Saving reports
 - using command files, *REPORT*, 1-8 to 1-9
 - using procedures, *REPORT*, 1-6 to 1-8
- Second (time) function (FN\$SECOND), *REF*, 6-15
- Security
 - dictionary definitions, *UG*, 19-1 to 19-14
- SELECT statement, *INTRO*, 7-1 to 7-4; *REF*, 5-236 to 5-241
 - forms of, *INTRO*, 7-2 to 7-4
 - SHOW command with, *INTRO*, 7-4
 - with DROP statement, *REF*, 5-238
- Semicolon, *REF*, 1-6
 - in DECLARE PORT statement, *REF*, 5-51
 - in DEFINE DOMAIN clause, *REF*, 5-61
 - in DEFINE PORT command, *REF*, 5-69
 - in DEFINE RECORD command, *REF*, 5-75

Semicolon (cont'd.)

- in DELETE command, *REF*, 5-86
- in domain definition, *REF*, 5-58
- with DEFINE DOMAIN command, *UG*, 4-2
- with DELETE command, *UG*, 20-4
- Sequence numbers, *UG*, 19-2
- Sequential files, *INTRO*, 3-3 to 3-4; *UG*, 6-1 to 6-9; *REF*, 1-11
 - compared with indexed, *UG*, 6-2T
 - creating, *REF*, 5-63, 5-66
 - defining, *UG*, 6-3 to 6-4
 - modifying, *INTRO*, 7-1 to 7-7
 - modifying records in, *REF*, 5-67
 - MODIFY statement with, *INTRO*, 8-4
- SET ABORT command, *UG*, 10-6, 18-4; *REF*, 5-242, 5-244
 - effect on ABORT statement, *UG*, 18-4; *REF*, 5-15
 - with REPEAT statement, *REF*, 5-212
- SET COLUMNS_PAGE command, *UG*, 18-1 to 18-4; *REF*, 5-243, 5-244
- SET command, *REF*, 5-242 to 5-245
 - terminal control, *UG*, 1-3
- SET DICTIONARY command, *UG*, 3-3 to 3-4; *REF*, 5-243, 5-245
 - in QUERY.INI file, *UG*, 2-3
- SET GUIDE command, *INTRO*, 4-3; *UG*, 2-9, 10-3; *REF*, 5-243
 - in QUERY.INI file, *UG*, 2-3
- SET NO ABORT command, *UG*, 10-6; *REF*, 5-243, 5-244
- SET NO PROMPT command, *REPORT*, 1-4; *REF*, 5-244
- SET PROMPT command, *UG*, 18-4 to 18-5; *REF*, 5-244
- SET statement (Report Writer), *REPORT*, 1-4; *REF*, 5-233 to 5-235
 - COLUMNS_PAGE, *INTRO*, 9-4; *REPORT*, 2-2, 2-5 to 2-6; *REF*, 5-234
 - DATE, *REPORT*, 2-10 to 2-11; *REF*, 5-235t
 - LINES_PAGE, *REPORT*, 2-6 to 2-7; *REF*, 5-234
 - MAX_LINES, *REPORT*, 2-7 to 2-8; *REF*, 5-234
 - MAX_PAGES, *REPORT*, 2-7 to 2-8; *REF*, 5-234
 - NO DATE, *REPORT*, 2-11 to 2-12
 - NO NUMBER, *REPORT*, 2-12
 - [NO] DATE, *REF*, 5-233
 - [NO] NUMBER, *REF*, 5-233
 - NUMBER, *REF*, 5-235t

SET statement (Report Writer) (cont'd.)

REPORT_NAME, *REPORT*, 2-1, 2-8, 2-9 to 2-10; *REF*, 5-233, 5-235t

SET TERMINAL command, *UG*, 18-1

SETUP.DTR command

- RSTS/E systems, *INTRO*, 2-1
- RSX systems, *INTRO*, 2-1

Severity codes, DAB\$W_ERR_SEV, *CALL*, 4-17, 8-4

SHARED access option, *REF*, 5-195

SHOW ALL command, *INTRO*, 2-3; *UG*, 20-2; *REF*, 5-246

SHOW COLLECTIONS command, *REF*, 5-208, 5-246

SHOW command, *UG*, 1-3, 2-2; *REF*, 5-239, 5-246 to 5-249

- with domains, *UG*, 4-3

SHOW CURRENT command, *REF*, 5-247

SHOW DICTIONARY command, *INTRO*, 2-2; *UG*, 3-2 to 3-4; *REF*, 5-247

SHOW DOMAINS command, *INTRO*, 2-3; *REF*, 5-247

SHOW FIELDS command, *REPORT*, 1-4; *REF*, 5-208, 5-247

SHOWP command, *UG*, 1-3; *REF*, 5-250 to 5-251

SHOW PROCEDURES command, *REF*, 5-247

SHOW READY command, *INTRO*, 2-4; *REF*, 5-198, 5-208, 5-247

SHOW RECORDS command, *INTRO*, 2-4; *REF*, 5-247

SHOW SPACE command, *REF*, 5-247

SHOW TABLES command, *UG*, 12-8; *REF*, 5-248

SIGN clause, *UG*, 5-12; *REF*, 5-252 to 5-253

- in field definition, *REF*, 4-8
- LEADING, *REF*, 5-252
- SEPARATE, *REF*, 5-252
- TRAILING, *REF*, 5-252

Sign of number function (FN\$SIGN), *REF*, 6-16

Single record context, *UG*, A-16

SKIP

- in AT BOTTOM statement (RW), *REF*, 5-221t
- in AT TOP statement (RW), *REF*, 5-226t
- print list element, *REF*, 5-181t
- RW print list element, *REF*, 5-230t

Slash (/), *REPORT*, 2-9, 2-18

- edit string character, *REF*, 5-107, 5-113

SORTED BY clause, *UG*, 7-10; *REPORT*, 3-2 to 3-8

- format, *REF*, 3-8
- in RSE, *REF*, 3-8
- sort keys, *REF*, 3-8

Sort keys, *INTRO*, 6-8; *REPORT*, 3-2 to 3-8

Sort keys (cont'd.)

ASCENDING, *REF*, 3-8

DECREASING, *REF*, 3-8

DESCENDING, *REF*, 3-8

INCREASING, *REF*, 3-8

- in SORT statement, *REF*, 5-254
- in SUM statement, *REF*, 5-266
- multiple, *REPORT*, 3-5; *REF*, 3-9

SORT statement, *UG*, 1-4; *REF*, 5-254 to 5-256

- access mode required, *REF*, 5-198t
- ASCENDING order, *REF*, 5-254
- DECREASING order, *REF*, 5-254
- DESCENDING order, *REF*, 5-254
- INCREASING order, *REF*, 5-254

SPACE, *REPORT*, 2-15, 2-16

- in AT BOTTOM statement (RW), *REF*, 5-221t
- in AT TOP statement (RW), *REF*, 5-226t
- in SHOW command, *REF*, 5-247
- print list element, *REF*, 5-181t
- RW print list element, *REF*, 5-230t

Spawning processes (FN\$DCL), *REF*, 6-7

Special page headings, *REPORT*, 3-13 to 3-15

Specifying domain names

- DEFINE DOMAIN command, *UG*, 4-1 to 4-3

Stallpoints, *CALL*, 4-1

Starting

- an ADT session, *INTRO*, 3-6
- DATATRIEVE, *INTRO*, 2-1 to 2-3

Startup banner, *UG*, 2-1

Statements

- See Commands and statements*

Statistical expressions, *REF*, 2-13 to 2-15

- RSE in, *REF*, 2-14

Statistical functions, *REPORT*, 1-9, 2-14, 2-19 to 2-22; *REF*, 2-13t

- AVERAGE, *REPORT*, 1-9, 2-19
- COUNT, *REPORT*, 1-9, 2-19
- MAX, *REPORT*, 1-9, 2-21
- MIN, *REPORT*, 1-9, 2-21
- TOTAL, *REPORT*, 1-9, 2-19

Status code, DAB\$W_ERR_CODE, *CALL*, 4-17, 8-4

Stopping command execution, DTUNWD, *CALL*, 4-31, 8-30

STORE statement, *INTRO*, 5-1 to 5-5; *UG*, 1-4, 2-6; *REF*, 5-257 to 5-265

- access mode required, *REF*, 5-198t
- assigning field values with, *REF*, 5-25, 5-29
- cancelling, *INTRO*, 5-2
- data types, *INTRO*, 5-4
- effect of ABORT statement, *REF*, 5-15
- in FOR statement, *REF*, 5-261

STORE statement (cont'd.)

- prompts, *REF*, 5-258
- REPEAT with, *INTRO*, 5-3
- using, to store into a port, *CALL*, 4-24
- VERIFY clause, *REF*, 5-260
- String extracting function (FN\$STR_EXTRACT), *REF*, 6-17
- STRLEN, parameter to DTINIT, *CALL*, 4-10
- Sublists, *UG*, 5-21 to 5-22
 - retrieving values, *UG*, 14-11 to 14-13
- Substitution directive (ICMD), *CALL*, 4-13, 8-9
- Substring
 - starting position function (FN\$STR_LOC), *REF*, 6-18
- Summaries
 - group, *REPORT*, 3-9 to 3-10
- Summarizing data, *REPORT*, 2-19 to 2-22
- Summary lines, *REPORT*, 1-9, 2-19 to 2-22
- Summary statistics, *REPORT*, 1-9
- SUM statement, *UG*, 1-4; *REPORT*, 1-3, 1-3 to 1-4; *REF*, 5-266 to 5-269
 - access mode required, *REF*, 5-198t
 - ASCENDING order, *REF*, 5-266
 - DECREASING order, *REF*, 5-266
 - DESCENDING order, *REF*, 5-266
 - INCREASING order, *REF*, 5-266
- SUPERSEDE clause, *UG*, 6-8
 - in DEFINE FILE command, *REF*, 5-64
- 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*, 19-1 to 19-14

T

T (text)

- edit string character, *REPORT*, 2-17; *REF*, 5-108

TAB

- in AT BOTTOM statement (RW), *REF*, 5-222t
- in AT TOP statement (RW), *REF*, 5-226t
- print list element, *REF*, 5-181t
- RW print list element, *REF*, 5-230t

Tables, *UG*, 12-1 to 12-11

- and workspace, *UG*, 12-7 to 12-8
- code/translation strings, *UG*, 1-5, 12-1 to 12-7
- code:translation strings, *REF*, 5-78
- comparing with IN, *REF*, 2-24
- creating, *UG*, 12-2 to 12-5
- default edit string, *REF*, 5-80

Tables (cont'd.)

- defining, *UG*, 3-1; *REF*, 1-12, 5-78 to 5-80
- defining and using, *UG*, 12-4 to 12-5
- deleting, *UG*, 12-10; *REF*, 5-86
- displaying, *UG*, 12-8
- displaying contents, *UG*, 12-8 to 12-9
- editing, *UG*, 12-4, 12-9 to 12-10; *REF*, 5-100 to 5-101
- functions and uses of, *UG*, 1-5; *REF*, 1-12
- IF-THEN-ELSE statement, *UG*, 12-6
- in a record selection expression, *UG*, 12-5
- in value expressions, *REF*, 2-12 to 2-13
- invoking, *REF*, 5-80
- maintaining, *UG*, 12-10
- modifying, *REF*, 5-80
- protecting, *UG*, 12-10 to 12-11
- releasing, *REF*, 5-207 to 5-210
- retrieving values with VIA, *REF*, 2-12
- sample, *UG*, 12-1 to 12-2
- showing, *REF*, 5-248
- using IN relational operator, *UG*, 12-5
- using with the Call Interface, *CALL*, 1-7
- validating data, *UG*, 12-6 to 12-7
- VALID IF clause, *UG*, 12-6 to 12-7
- VIA value expression, *UG*, 12-7
- WITH in, *UG*, 12-5 to 12-6
- Task Builder
 - using, with Callable DATATRIEVE, *CALL*, 3-4 to 3-8
- Task Builder command file
 - example of, on RSTS, *CALL*, 3-4
 - example of, on RSX-11M-PLUS, *CALL*, 3-5
- Task size, reducing
 - See Optimizing workspace*
- Terminal Interface
 - See Remote Terminal Interface*
- Termination characters, *REF*, 1-6 to 1-7
 - RETURN key, *REF*, 1-7
 - semicolon, *REF*, 1-6
- Terminology
 - DATATRIEVE, *INTRO*, 1-2
- THEN statement, *REF*, 5-270 to 5-271
- TI
 - for output to terminal, *REPORT*, 2-23
- Time
 - FN\$DATE function, *REF*, 6-5
- Time function (FN\$TIME), *REF*, 6-19
- Title page, *REPORT*, 3-11
- TODAY value expression, *REF*, 2-10
- Totals
 - SUM statement, *REF*, 5-268

TOTAL statistical function, *REPORT*, 2-19 to 2-20;
REF, 2-13
 in AT BOTTOM statement (RW), *REF*, 5-222t
 in AT TOP statement (RW), *REF*, 5-227
 Transferring control
 with ABORT, *REF*, 5-14
 Transferring data, *CALL*, 4-15
 Transferring records, *CALL*, 4-22 to 4-30
 Transfers
 with IF-THEN-ELSE statement, *REF*, 1-3

U

UIC

See User identification code

Underscore (*_*), *UG*, 1-2, 4-2
 Underscore (*_*), *REF*, 1-5
 in record name, *UG*, 5-4
 UNITS Task Builder qualifier, *CALL*, 3-10
 Unwinding, DTUNWD, *CALL*, 4-31, 8-30
 Uppercase function (FN\$UPCASE), *REF*, 6-20
 Uppercase letters
 in string literals, *REF*, 2-3
 USAGE clause, *UG*, 5-12, 5-13 to 5-14; *REF*,
 5-272 to 5-275
 COMP-1 data type, *REF*, 5-274
 COMP-2 data type, *REF*, 5-274
 COMP-3 data type, *REF*, 5-274
 COMP-5 data type, *REF*, 5-274
 COMP data type, *REF*, 5-274
 DATE, *REF*, 5-274
 in DECLARE statement, *REF*, 5-47
 in field definition, *REF*, 4-8
 User identification code, *UG*, 3-3
 keys, *UG*, 19-2 to 19-4
 USING clause, *REPORT*, 2-14, 2-17
 in PRINT statement, *REF*, 5-183
 in RW PRINT statement, *REF*, 5-231
 in STORE statement, *REF*, 5-257

V

V (decimal point)
 picture string character, *REF*, 5-177
 VALID IF clause, *UG*, 5-12; *REF*, 5-26, 5-33, 5-276
 to 5-277
 in field definition, *REF*, 4-8
 in record definition, *UG*, A-9
 Value expressions
 concatenated, *REF*, 2-17 to 2-19
 DATE, *REF*, 2-10 to 2-11

Value expressions (cont'd.)

defined, *REF*, 2-1
 displaying, *REF*, 5-91 to 5-93
 global variables in, *REF*, 2-9
 in arithmetic expressions, *REF*, 2-15 to 2-17
 in AT BOTTOM statement (RW), *REF*, 5-222t
 in AT TOP statement (RW), *REF*, 5-227t
 in detail lines, *REPORT*, 2-13, 2-13 to 2-14
 local variables in, *REF*, 2-9
 prompting, *REPORT*, 1-6 to 1-8; *REF*, 2-11 to
 2-12
 statistical, *REF*, 2-13 to 2-15
 table, *REF*, 2-12 to 2-13
 TODAY, *REF*, 2-10
 types of, *REF*, 2-1
 use of, *REF*, 2-1
 variables in, *REF*, 2-8 to 2-10
 Variables, *UG*, 11-1 to 11-10; *REPORT*, 2-14
 as counters, *UG*, 11-7 to 11-10
 assigning values to, *REF*, 5-32 to 5-35, 5-49
 context, *REF*, 2-7
 declaring, *UG*, 11-1; *REF*, 2-8, 5-47 to 5-50
 defining global, *REF*, 2-9, 5-48
 defining local, *REF*, 2-9, 5-37
 displaying with SHOW FIELDS, *REF*, 5-50
 formatting for output, *REF*, 5-102 to 5-119
 global, *UG*, 11-4
 in detail lines, *REPORT*, 2-13
 initializing, *REF*, 2-8
 initializing global, *REF*, 5-47
 initializing local, *REF*, 5-37
 in value expressions, *REF*, 2-8 to 2-10
 local, *UG*, 11-4 to 11-5
 releasing global, *REF*, 2-9, 5-50, 5-207 to 5-210
 storing values, *UG*, 11-2 to 11-7
 VERIFY clause
 in MODIFY statement, *REF*, 5-260
 in STORE statement, *UG*, A-8; *REF*, 5-260,
 5-263
 VERIFY USING clause
 in STORE statement, *REF*, 5-257, 5-259
 VIA clause, *REF*, 2-12, 5-80
 VIA value expression, *UG*, 12-7
 View domains, *UG*, 13-1 to 13-8
 combining data, *UG*, 13-4 to 13-6
 containing a list, *UG*, 13-7 to 13-8
 defining, *UG*, 13-2; *REF*, 5-60 to 5-62
 limiting record access, *UG*, 13-3
 using, *UG*, 13-6 to 13-8
 Virtual fields
 See COMPUTED BY clause

W

W (day letter)

edit string character, *REF*, 5–107

WHILE statement, *UG*, 11–9; *REF*, 5–278 to 5–280

with STORE statement, *REF*, 5–263

WITH clause

in RSE, *REF*, 3–7

Workspace, *UG*, 12–7

defined, *UG*, 17–1

optimizing use of, *UG*, 17–5 to 17–7

WRITE access mode, *REF*, 5–196

X

X (alphanumeric)

edit string character, *REF*, 5–108

picture string character, *REF*, 5–175

Y

Y (year)

edit string character, *REF*, 5–107

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*, 18–2 to 18–4

Year function (FN\$YEAR), *REF*, 6–21

Z

Z

edit string character, *REF*, 5–109

Zero (0)

edit string character, *REF*, 5–113

ZONED data type

See COMP-5 data type

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-DEC-DEMO (800-332-3366) using a 1200- or 2400-baud modem. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local DIGITAL subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	_____	Local DIGITAL subsidiary or approved distributor
Internal ¹	_____	SDC Order Processing - WMO/E15 <i>or</i> Software Distribution Center Digital Equipment Corporation Westminster, Massachusetts 01473

¹For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

DATATRIEVE-11
Summary Description
AA-U052C-TC

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

What I like best about this manual is _____

What I like least about this manual is _____

I found the following errors in this manual:

Page	Description
------	-------------

_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

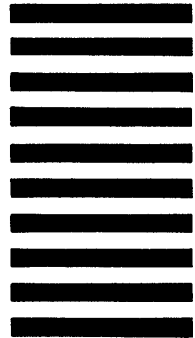
_____ Phone _____

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

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987



Do Not Tear - Fold Here

Cut Along Dotted Line

Reader's Comments

DATATRIEVE-11
Summary Description
AA-U052C-TC

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

What I like best about this manual is _____

What I like least about this manual is _____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

I am using **Version** _____ of the software this manual describes.

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

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

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987



Do Not Tear - Fold Here

Cut Along Dotted Line

digital



SHREWSBURY LIBRARY
DIGITAL EQUIPMENT CORPORATION
SHR1-3/G18
DTN 237-3400

Printed in U.S.A.