

pdp11

DATATRIEVE-11
V1.1 Release Notes
Order No. AA-D110B-TC

digital

First Printing, September, 1978

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

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

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

Copyright © 1978 Digital Equipment Corporation

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECTape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	

September 1978

This document describes the differences between DATATRIEVE-11 V1.0 and DATATRIEVE-11 V1.1

DATATRIEVE-11
V1.1 Release Notes

Order No. AA-D110B-TC

OPERATING SYSTEM AND VERSION:	VAX/VMS	V1.0
	RSTS/E	V06C
	RSX-11M	V3.1
	IAS	V2.0
SOFTWARE VERSION:	DATATRIEVE-11	V1.1

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard. massachusetts

First Printing, September, 1978

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

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

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

Copyright © 1978 Digital Equipment Corporation

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	

Contents

Topic	Section
GENERAL NOTES	1.0
INSTALLATION PROCEDURE CHANGES	2.0
CHANGES AND ADDITIONS.	3.0
Tutorial Software Added (Guide Mode)	3.1
CONTAINING Relational Operator Added	3.2
Command Interruption Supported.	3.3
SET/SHOW DICTIONARY Commands Added.	3.4
Data Input Prompt Shortened	3.5
Unquoted Literal Strings Accepted	3.6
Command Input Prompts Now Indicate Context	3.7
Characters tolerated in Numeric Input Fields	3.8
Special Input Sequence Leaves Field Unchanged.	3.9
Relational Expression Syntax Broadened	3.10
Miscellaneous Syntax Additions.	3.11
Restrictions on Procedures Removed.	3.12
REDEFINES Clause Supported.	3.13
Nested Procedures Supported	3.14
Procedure Editor Added	3.15
DEFINE FILE Command Added	3.16
DELETE Command Improved	3.17
MODIFY VERIFY Supported.	3.18
MODIFY USING Supported	3.19
Definition of Privileged User Changed	3.20
DEFINE RECORD Command Gives Record Length	3.21
Percent Character Allowed in EDIT-STRING	3.22
Command Line Command Files Supported	3.23
Statistical Operations Print Default Changed	3.24
Underscore Character Accepted in Names	3.25
Report Writer Formatter	3.26
Packed Decimal Datatype Supported	3.27
Context Rules Expanded for STORE USING Statement	3.28
Prompts Supported for Report Writer Parameters	3.29
Scope of Prompts in Loops Fixed	3.30
Record Definition Length Limitation Removed.	3.31
Dictionary Compression Utility Added.	3.32
Emulation of Floating Point Processor Added	3.33
Built with RMS Runtime System on RSTS	3.34
Maximum Keys per Index Files Controllable.	3.35
RESTRICTIONS AND LIMITATIONS.	4.0

APPENDIX A DATATRIEVE PROCEDURE EDITOR	A.1
EDITOR SUMMARY	A.1
Summary of EDIT Commands	A.1.1
Summary of EDIT Ranges	A.1.2
EDITOR COMMANDS	A.2
DELETE	A.2.1
EXIT	A.2.2
INSERT	A.2.3
QUIT	A.2.4
REPLACE	A.2.5
SUBSTITUTE	A.2.6
TYPE	A.2.7
RANGE SPECIFICATIONS	A.3
Period	A.3.1
%BEGIN	A.3.2
%END.	A.3.3
The Next Occurrence of a String	A.3.4
%WHOLE	A.3.5
%REST	A.3.6
%BEFORE.	A.3.7
%ALL	A.3.8
%FOR	A.3.9
%AND.	A.3.10
A Fixed Offset from a Given Line	A.3.11

Datatrieve Version V1.1 Release Notes
General Notes

1.0 General notes

Datatrieve Version 1.1 is strictly upwards compatible from Version 1.0. The dictionary format has not changed; all domains, records, procedures, and command files used for Version 1.0 may be used without modification on Version 1.1.

NOTE

Although the maintenance release of Datatrieve-11 is designated as Version 1.1, the corresponding version of the software is V01.10. The latter designation appears in the start-up banner of Datatrieve.

2.0 Installation Procedure Changes

Datatrieve Version 1.1 is built with a fixed central dictionary and messages file. This has two direct effects on the installation procedure:

1. The installation procedure is now completely automated.
2. Installation must be performed from a privileged account.

Both the dictionary and the message file are installed in LB:[1,2] for all operating systems but RSTS where they reside on SY:[1,2].

See the Datatrieve-11 Installation Guide for the specific installation procedures.

3.0 Changes and Additions

The following changes and additions have been made to the Datatrieve Version 1.1 software and are not documented in the Datatrieve-11 User's Guide.

3.1 Tutorial Software Added (Guide Mode)

A CRT based tutorial has been added to Datatrieve Version 1.1. The tutorial feature can be used only by VT100, VT52, and VT52 compatible terminals. A tutorial session is entered by the Datatrieve command:

SET GUIDE

The software is self documenting.

Typing SET GUIDE from an unsupported terminal produces the prompt:

ENTER TERMINAL TYPE:

Although you must enter a character, all responses result in the message:

Terminal not supported for guide mode.

3.2 CONTAINING Relational Operator Added

The substring search relational operator CONTAINING has been added. CONTAINING is used in a record selection expression to retrieve records with a field containing a particular substring. The substring may be anywhere in the field, and need not match the case (upper/lower case) of the search string. For example, the command:

FIND BOOKS WITH TITLE CONTAINING "LASER"

finds all records in BOOKS with the word "laser" somewhere in the field TITLE.

3.3 Command Interruption Supported

A command may be interrupted on the RSTS/E, RSX-11M, and VAX/VMS operating systems by typing ^C. After recognition of the ^C, Datatrieve returns to the DTR> command level. This feature is not available on IAS.

Care should be exercised to avoid typing a second ^C before Datatrieve has returned to its command level.

3.4 SET/SHOW DICTIONARY Commands added

The data dictionary may be changed during a Datatrieve Version 1.1 session with the command:

```
SET DICTIONARY file-spec
```

where "file-spec" is the file specification of a data dictionary. If the file does not exist, cannot be accessed, or is not a valid dictionary, the former dictionary assignment remains in effect. To return to the default system dictionary, type the command:

```
SET DICTIONARY
```

To display the current dictionary, type:

```
SHOW DICTIONARY
```

Since Datatrieve 1.1 is built with a central dictionary, SET DICTIONARY is the preferred way to access a private dictionary. A private dictionary may be created by copying the system dictionary with PIP, COPY, or the QCPRS utility program, or by the following RMS define utility command:

```
DFN PRIVAT.DIC /FO:IDX /RF:FIX:512 /AL:100 /BK:2-  
/KY:30:2:(/OP:DUP) /KY:0:31:(/OP:NODUP)
```

3.5 Data Input Prompt Shortened

The prompt message has been changed from:

```
Please supply value for PROMPT-NAME:
```

to:

```
Enter PROMPT-NAME:
```

3.6 Unquoted Literal Strings Accepted

An undefined name on the right hand side of a relational expression is assumed to be a literal string. The Datatrieve statement:

PRINT YACHTS WITH BUILDER EQ PEARSON

would be rejected as an error (Unknown field name "PEARSON") in version 1.0 of Datatrieve. In Version 1.1, Datatrieve issues the warning message:

"PEARSON" is not field, assumed literal

and executes the command as if it had been entered:

PRINT YACHTS WITH BUILDER EQ "PEARSON"

Datatrieve does not interpret an unquoted string as a literal if the string is known as a keyword, a domain name, a collection name, or a record name. Nor does Datatrieve treat an unquoted string as a literal if it is on the left side of a relational expression. Datatrieve automatically converts unquoted literal to upper case.

3.7 Command Input Prompts Now Indicate Context

The input prompt changes depending on the context within Datatrieve. The following prompts are used:

Prompt	Meaning
DTR>	Datatrieve is waiting for a command
DFN>	A dictionary element is being defined
RW>	The report writer is waiting for a command
QED>	The editor is waiting for a command
IN>	The Datatrieve editor is in input mode

3.8 Characters tolerated in Numeric Input Fields

The characters percent (%), comma (,), and dollars (\$) are tolerated if entered in response to an input prompt for a numeric field. The characters are ignored. In Version 1.0 of Datatrieve, the presence of any of these characters would have forced a warning message.

3.9 Special Input Sequence Leaves Field Unchanged

The input sequence <tab><carriage-return>, when given in response to a prompt for a STORE, MODIFY, or assignment statement causes the field to remain unchanged.

3.10 Relational Expression Syntax Broadened

Relational expressions of the form:

field-name NOT relation value

are now accepted by Datatrieve where "field-name" is a field name, "relation" is a relational operator (EQ, NE, BETWEEN, etc.), and "value" is a value expression. The effect is identical to the construct:

NOT field-name relation value

In other words, the following statement is now legal:

FIND YACHTS WITH LOA NOT BETWEEN 30 AND 40

3.11 Miscellaneous Syntax Additions

The keyword "BUT" has been added as a synonym of "AND". This allows commands such as:

PRINT YACHTS WITH PRICE<20000 BUT LOA>30

The keyword "AND" may be used in place of or in conjunction with a comma when separating print objects in a print list. Hence the following commands are valid:

PRINT TYPE, RIG, AND PRICE OF YACHTS

PRINT RIG AND PRICE

3.12 Restrictions on Procedures Removed

Most restrictions on the types of commands which may be placed in procedures have been lifted. In the maintenance release, any command except DEFINE may be included in a procedure. This specifically includes READY and SHOW.

3.13 REDEFINES Clause Supported

The COBOL record definition clause REDEFINES is supported. The following is an example of the REDEFINES clause:

```
03 DATES.  
05 DATE      PIC X(6) EDIT-STRING IS XX/XX/XX .  
05 SEP-DATE  REDEFINES DATE.  
07 MONTH    PIC 99.  
07 DAY      PIC 99.  
07 YEAR     PIC 99.
```

The field SEP-DATE redefines the field DATE in that it provides an alternate description of the same data. Specifically, the field MONTH is the first two bytes of the field DATE, the field DAY is the third and fourth bytes of DATE, and the field YEAR is the last two bytes of DATE. This allows you to enter separate values for the month, day, and year field, to sort the file based on the natural order of dates, yet print the date with a standard default format. The redefining field must be contained in the same group as the original field, must have the same level number, and must be exactly the same length as the original field. Redefinitions are ignored in PRINT and MODIFY lists unless referenced explicitly.

3.14 Nested Procedures Supported

Procedures may contain references to other procedures (nested procedures) provided that no procedure invokes itself either directly or indirectly. The maximum depth of procedure nesting varies from about 10 to about 30 depending on the amount of memory available, number and size of established collections, etc.

3.15 Procedure Editor Added

A Datatrieve procedure editor has been added. The editor is invoked by the command:

```
EDIT procedure-name
```

where "procedure-name" is the name of an existing procedure. The editor is a small subset of EDT, the DEC Standard Editor. See Appendix A for the editor specification.

3.16 DEFINE FILE Command Added

A command to create the RMS file for a domain has been added. The syntax of the command is:

```
DEFINE FILE FOR domain [option-1] [, option-2...]
```

where "domain" is a domain name. Each "option" must be one of the following:

ALLOCATION=blocks

SUPERCEDE

KEY=field [(key-modifiers...)]

where "blocks" is the number of disk blocks to be allocated for the file, "field" is the name of a field in the domain to be used as a key, and "key-modifiers" are one or two modifiers from the list:

CHANGE

NO CHANGE

DUP

NO DUP

If two key modifiers are given, they must be separated by commas.

Any number of keys may be specified (up to a maximum of 254); if no keys are specified a sequential file is created.

To define a file, the following conditions must be met:

1. The domain must be defined in the dictionary.
2. The record referenced in the domain definition must be defined in the dictionary.
3. The user must have control access to the domain.

An indexed file is created if at least one KEY is specified; if no keys are specified, the file created is sequential. If more than one key is specified, the first is the primary key, the second the first alternate key, etc. The default options for the primary key are (NO CHANGE, NO DUP). For alternate keys the defaults options are (CHANGE, DUP).

For all files, the record format is fixed length with the

length determined by the record definition. The device, UIC or PPN, file name, extension, and version number are taken from the domain definition.

Examples:

```
DEFINE FILE FOR YACHTS KEY=BUILDER(NO CHANGE, DUP),  
KEY=MODEL,  
ALLOCATION=30, SUPERCEDE
```

```
DEFINE FILE FOR YACHTS
```

In the first example, an indexed file with two keys is created. The first key corresponds to the field BUILDER in the record definition (YACHT) associated with the domain YACHTS, and has the NO CHANGE, DUP attributes. Duplicate values are allowed for the BUILDER field, but once stored, that field may not be modified. If NO DUP were specified, any attempt to store a second record with the same builder would be rejected. The second key corresponds to the field MODEL, may be modified, and may have duplicates.

On RSTS/E, the option SUPERCEDE indicates that the old version of the file specified in the domain definition (YACHT.DAT) should be deleted and replaced with the new version. Without the SUPERCEDE option DEFINE FILE will not create a new file on RSTS if a file of the same name exists in the designated directory.

On RSX-11M, IAS, and VAX/VMS, DEFINE FILE creates a new version of the file unless the version number is explicit in the domain definition. In that case, SUPERCEDE can be used to create another file with the same extension, and delete the old one.

In the second example, a sequential file is created. It too uses the record description YACHT to determine the record length, and the domain definition of YACHTS to find the file name. is lost. If an error is found, correct it immediately, or cancel the edit with the editor QUIT command.

3.17 DELETE Command Improved

An unsuccessful DELETE command no longer kills command files. A command file containing a record definition may conveniently contain a command to delete the previous definition.

3.18 MODIFY VERIFY Supported

The VERIFY USING clause is now supported on the MODIFY command. It works the same way as does the VERIFY clause on the STORE command.

3.19 MODIFY USING Supported

The USING clause on the MODIFY command is recognized in Datatrieve version 1.1. The syntax of the USING form of MODIFY is:

```
MODIFY [ALL] USING stmt-1 [VERIFY USING stmt-2] [of rse]
```

where stmt-1 is a single Datatrieve statement (simple or compound) containing at least one assignment statement.

3.20 Definition of Privileged User Changed

A login project code of one is now required to get automatic, de facto control access to dictionary elements. In Version 1.0 of Datatrieve, any login project code less than eight was sufficient.

3.21 DEFINE RECORD Command Gives Record Length

The DEFINE RECORD command, when successful, issues a message containing the length in bytes of the record defined.

3.22 Percent Character Allowed in EDIT-STRING

The special symbol % is allowed in EDIT-STRING. It is treated as a literal character. For example, the command:

```
PRINT 15 USING 99%
```

prints as:

15%

3.23 Command Line Command Files Supported

If Datatrieve is installed (on RSX-11M or IAS) or invoked via a CCL (on RSTS), any data on the invocation line is accepted by Datatrieve as the sole input, i.e. Datatrieve executes the line (or command file) and exits immediately.

For example, if Datatrieve were installed (INS command on RSX-11M and IAS, a CCL on RSTS/E), the acceptance test could be run by the operating system command:

```
DTR @DTR.TST
```

3.24 Statistical Operators Print Default Changed

When used in a print list, MAX, MIN, and AVERAGE of a single field acquire the same default column header and edit string as the field itself.

3.25 Underscore Character Accepted in Names

The underscore character () is permitted in Datatrieve names provided that it is neither the first nor last character.

3.26 Report Writer Formatter

An initial literal (or skip followed by a literal) in a print list is now aligned with the first column of data. to the first logical columns group. See the procedure LOA-REPORT in the Version 1.1 acceptance test for an example.

3.27 Packed Decimal Datatype Supported

The packed decimal datatype is supported. A number is represented in packed decimal as a sequence of 4 bit nibbles, stored two nibbles per byte. The low order byte (high address) of a packed decimal field contains only one digit nibble, plus a sign nibble. The sign nibble must contain either one of the positive sign codes (10, 12, 14, 15, decimal) or one of negative sign codes (11 or 13 decimal). A packed decimal field is indicated in the record definition by a USAGE of COMP-3. For example, the field definition

05 NUMBER-ON-HAND PIC S9(5) USAGE COMP-3

defines a packed decimal field 3 bytes long.

The packed decimal datatype used by Datatrieve is compatible with the packed decimal used by VAX-11 hardware, the RPG-II language, and IBM hardware.

3.28 Context rules expanded for STORE USING statement

The recognition rules for names occurring on the right hand side of assignment statements have been modified. Now, references to data in fields in the record can be stored. Two conditions must be met:

1. A context variable must be present on the STORE USING statement. The syntax is

STORE context IN domain USING statement

where "domain" is the domain into which the new record is to be stored, "context" is the context variable, and "statement" is a Datatrieve statement.

2. Any reference to a field in the record being stored must be explicitly qualified by the context variable.

Examples:

```
STORE NEW IN REVENUES USING
BEGIN
  Q1=*.Q1      ! FIRST QUARTER
  Q2=*.Q2      ! SECOND QUARTER
  Q3=*.Q3      ! THIRD QUARTER
  Q4=*.Q4      ! LAST QUARTER
  YEAR-TOTAL = NEW.Q1 + NEW.Q2 + NEW.Q3 +NEW.Q4
END
```

```
STORE SAIL IN YACHTS USING
BEGIN
    ...
    LOA=*.LOA
    IF SAIL.LOA NOT BETWEEN 20 AND 45 THEN
        PRINT "BAD LENGTH-OVER-ALL, RE-ENTER" THEN
        LOA=*.LOA-AGAIN
    ...
END
```

3.29 Prompts Supported for Report Writer Parameters

If a REPORT-NAME is not included in a report specification, Datatrieve Version 1.1 prompts for one. The report name entered must conform to the same rules as SET REPORT-NAME. It can not be longer than the input line, although it can include slash characters so that it will print on more than one line.

The report writer also supports *.prompts for both SET LINES-PAGE and SET COLUMNS-PAGE.

3.30 Scope of prompts in loops fixed

In version 1.0 the command:

```
REPEAT 5 PRINT YACHTS WITH BUILDER EQ *.BUILDER
```

prompts once for BUILDER, then prints the same list of YACHTS five times. In Version 1.1 of Datatrieve, that statement prompts five times, print a list each time.

3.31 Record Definition Length Limitation Removed

The Version 1.0 limitation on the maximum length of dictionary elements (record definitions) has been eliminated. There is no practical bound on the length of a record or procedure definition.

3.32 Dictionary Compression Utility Added

A data dictionary file compression utility (QCPRS) has been provided. See the Datatrieve Installation Guide for specifications.

3.33 Emulation of Floating Point Processor Added

A floating point processor is no longer necessary to process the floating point datatypes COMP-1 and COMP-2. If a floating point processor is present, Datatrieve automatically uses it. If a floating point processor is not present, Datatrieve emulates the floating instructions.

3.34 Built with RMS Runtime System on RSTS

Datatrieve Version 1.1 on RSTS/E is built with the RMS runtime system to improve performance and increase the maximum task extension quantity.

3.35 Maximum Keys per Index Files Controllable

Datatrieve Version 1.0 could READY indexed files with a maximum of 10 keys. In Datatrieve Version 1.1, the maximum number of keys can be controlled with the global symbol MAXKEY in the task build command file (DTR11M.TKB, DTRIAS.LNK, DTRVAX.TKB, or DTRSTS.TKB). To increase the maximum number of keys accessible by Datatrieve, edit the appropriate task build file to increase the value assigned to MAXKEY, and rebuild Datatrieve.

4.0 Restrictions and Limitations

4.1 Collection Sizes

The maximum size of an established collection (resulting from execution of a FIND statement) is about 5,000 records. To circumvent this limitation, use RSE forms of the various Datatrieve statements. This also improves performance.

4.2 Dictionary Access on VAX/VMS

VAX/VMS does not support record locking in compatibility mode. To avoid problems associated with a shared data dictionary, Datatrieve opens the dictionary only when required. Occasionally, however, the dictionary may not be available when required. When this happens, Datatrieve waits five seconds, and attempts to re-access the dictionary; if necessary, it repeats this three times. If, at the end of 15 seconds, the dictionary is still not accessible, the operation is cleanly aborted leaving the dictionary unchanged.

4.3 Computing Ratios of Statistical Expressions

When it is necessary to compute the ratio of two statistical expressions, enclose each in parentheses to avoid ambiguity. For example, avoid expressions like

$$\text{TOTAL PRICE} / \text{TOTAL COST}$$

using instead the expression

$$(\text{TOTAL PRICE}) / (\text{TOTAL COST})$$

4.4 Arithmetic Limitations

In general Datatrieve is capable of arithmetic operations of up to 18 decimal digits. The multiplication and division operations (including TOTAL), however, are limited to operands and results with an absolute value less than $2^{*}31$ (about nine decimal digits).

Floating point calculations (expressions containing a field defined with a USAGE of COMP-1 or COMP-2) are limited by the

accuracy of the data type used.

4.5 GUIDE, High Baud Rates, and RSX-11M

When GUIDE mode is set, Datatrieve interrogates the user's terminal to determine its type. On RSX-11M, particularly with a VT100 transmitting above 1200 baud, the terminal may respond faster than Datatrieve can react. If this happens, one of the messages:

Terminal not support for GUIDE mode

Enter terminal type:

will result.

To overcome this problem, one of two solutions may be used:

1. If split baud rates are available, set the terminal to transmit at 150 or 300 baud.
2. Hard code the terminal type in the Datatrieve task build command file (DTR11M.TKB) and rebuild Datatrieve. The terminal type is controlled by the global TRMTYP. A value of 0 indicates that the terminal is to be determined; a value of 1 indicates a VT52; a VT100 is indicated by a value of 2.

APPENDIX A

Datatrieve Procedure Editor

The command `EDIT procedure-name` invokes an editor which can insert, replace, or delete text from procedures defined in the Data Dictionary.

The new version of the procedure replaces the old; if you realize during an edit that you have made a serious error, leave the editor with the command `QUIT` which kills the edit and restores the old version of the procedure.

A.1 Editor Summary

Purpose

This command invokes a subset of the DEC Standard Editor (EDT), opens the selected procedure to the editor, and changes the prompt from `DTR>` to `QED>`. When you have finished editing the procedure issuing an `EXIT` command or `^Z` replaces the old version with the new, and return you to Datatrieve.

Format

`EDIT procedure-name [(password)]`

A.1.1 Summary of EDIT commands

All of the editor commands follow the general syntax:

command [range-specification]

where "command" is one of the editor commands, and "range-specification" is a range specification.

The commands are:

T(YPE)	TYPE displays lines. Used alone it displays the next line. When followed by a "." it displays the current line.
<cr>	<cr> is an acceptable abbreviation for TYPE. Thus entering a carriage return without a command before it displays (and advances the cursor to) the next line.
D(ELETE)	DELETE deletes the line described by the range specification.
S(UBSTITUTE)	SUBSTITUTE searches for a string of characters and substitutes a second string of characters for them.
R(EPLACE)	REPLACE deletes the line described by the range specification and enters INSERT mode to accept a replacement which can be longer or shorter than the original.
I(NSERT)	INSERT changes the editor from command mode to insert mode, changing the prompt from QED> to IN>. All input is inserted into the procedure until a ^Z is entered, when the editor returns to command mode.
EX(IT)	EXIT is the normal way of leaving the editor. It replaces the old version of the procedure with one just created.
QUIT	QUIT is an abnormal ending to an edit. The new version of the procedure is deleted, and the old version is restored. Use it when your edit has gone awry. QUIT can not be abbreviated.

A.1.2 Summary of EDIT ranges

Single range specifications place the editor at the beginning of the line specified. They are:

"." A period specifies the current line. It is the default range for all the commands except TYPE.

%BE(GIN) The first line of the procedure.

%E(ND) The end of the procedure.

'string' or "string"

 The next line containing the string.

%WH(OLE) The whole procedure.

%R(EST) The rest of the procedure (including the current line).

%BEF(ORE) All the lines between the first and the current, inclusive.

%ALL 'string' or %ALL "string"

 Every line containing the string.

Range specifications can be connected. The position of the editor after using combined range specifications is not easily predictable. The connectors are:

%FOR or ;

 Lines from range-1 through range-1 plus n.

%AND or ,

 The line described by range-1, and the line described by range-2, etc.

+

 The line n lines after the line described by range-1.

A.2 Editor Commands

A.2.1 Delete one or more lines -- DELETE

Purpose

The DELETE command is used to delete one or more lines from a procedure.

Format

```
DELETE [range-1]  
D [range-1]
```

1. If you specify range-1, it must be a valid range specification

Requirements

None.

Prompts & Responses

None.

Effects

1. If you supply a range specification, the line or lines identified by range-1 are deleted from the procedure.
2. If you do not supply a range specification, the current line is deleted from the procedure.
3. The editor is now positioned at the line following the last line deleted.

Examples

```
DELETE  
  
DELETE %ALL "BOTTOM OF"
```

Hints

Remember that DELETE deletes complete lines.

The second example above deletes all the lines containing the string "BOTTOM OF", not just the characters "BOTTOM OF".

A.2.2 Exit from the Editor -- EXIT

Purpose

The EXIT command terminates the editor, stores the updated procedure in the data dictionary, and returns control to Datatrieve.

Format

EXIT
EX
^Z

Requirements

The updated procedure must not be empty.

Prompts & Responses

Datatrieve returns with the DTR> prompt.

Effects

1. The editor is terminated.
2. The edited procedure is written back to the data dictionary in place of its original definition.
3. Control is returned to Datatrieve.
4. The old definition of the procedure is deleted.

Example

EXIT

Hints

Before exiting the editor, it is good practice to print the entire procedure to verify that the intended modifications were made correctly. Once the editor is exited, the old procedure definition is lost. If an error is found, correct it immediately, or cancel the edit with the editor QUIT command.

A.2.3 Insert one or more lines -- INSERT

Purpose

The insert command allow new lines to be inserted into a procedure.

Format

```
INSERT [range-1]
```

```
I [range-1]
```

1. If you specify range-1, it must be a valid range specification

Requirements

None.

Prompts & Responses

The prompt changes from QED> to IN> when in insert mode.

Effects

1. The editor enter insert mode. Subsequent lines typed are inserted in the procedure. Insert mode is terminated by typing a <control-Z>.
2. If you do not supply a range specification, the new lines are inserted ahead of the current line.
3. If you supply a range specification, the new lines are inserted ahead of the first line indicated by the range specification.

Examples

```
I  
INSERT %BEGIN
```

Hints

To insert lines after the bottom line, the following procedure may be used:

1. Position the editor at the end of the procedure by typing the command %END (an implicit TYPE command).
2. Type the INSERT command to enter insert mode.

A.2.4 Abandon the Editor -- QUIT

Purpose

The QUIT command is used to abort an edit and return to Datatrieve without alter the procedure being edited.

Format

QUIT

Requirements

None.

Prompts & Responses

None.

Effects

1. The editor is terminated.
2. No modifications to the procedure are made.
3. Control is returned to Datatrieve.

Examples

QUIT

Hints

QUIT undoes any damage you may have inadvertently done to your procedure. QUIT when you're behind.

A.2.5 Replace One or More Lines -- REPLACE

Purpose

This command may be used to replace one or more complete lines with completely new lines.

Format

```
REPLACE [range-1]  
R [range-1]
```

1. If you specify range-1, it must be a valid range specification

Requirements

None.

Prompts & Responses

The prompt changes from QED> to IN> when in insert mode.

Effects

1. If you specified a range, the line or lines identified are deleted from the procedure.
2. If you do not specify a range, the current line is deleted from the procedure.
3. The editor enters insert mode. Subsequent lines typed are inserted in the procedure. Insert mode is terminated by typing a <control-Z>.
4. The editor is now positioned at the line following the last line deleted (after the last line inserted).

Examples

```
REPLACE  
R %WHOLE
```

Hints

Do not confuse REPLACE with SUBSTITUTE. The former replaces lines, the latter substitutes strings.

A.2.6 Substitute One Character String for Another -- Substitute

Purpose

This command substitutes one character string for another. It may substitute only the first occurrence, or all occurrences of a string.

Format

```
SUBSTITUTE /string-1/string-2/ [range-1]  
S /string-1/string-2/ [range-1]
```

1. If you specify range-1, it must be a valid range specification
2. The delimiting character "/" may be replaced with any printing character.

Requirements

None.

Prompts & Responses

All lines in which substitutions take place are typed.

Effects

1. If you supply a range specification, all occurrence of string-1 in all lines identified by range-1 are replaced with string-2. Any lines identified by range-1 which do not contain string-1 are not modified.
2. If you do not supply a range specification, only the first occurrence of string-1 is replaced with string-2. If string-1 is not found between the current line and the end of the procedure, no substitution is performed.

Examples

```
SUBSTITUTE /PRINT/PRINP/  
S *PRICE/LOB*PRICE/LOA*
```

S /FOOPAR/FOOBAR/ %WHOLE

Hints

Substitution of one string for another is probably the easiest strategy to follow when making minor modifications to a procedure.

A.2.7 Display One or More Lines -- TYPE

Purpose

The TYPE command is used to print one or more lines in a procedure.

Format

TYPE [range-1]
T [range-1]
[range-1]

1. If you specify range-1, it must be a valid range specification

Requirements

None.

Prompts & Responses

The appropriate lines are displayed on your terminal.

Effects

1. If you supply a range specification, the lines identified by range-1 are displayed on your terminal. Following completion of the command the editor is now positioned at the first line displayed.
2. If you do not supply a range specification, the editor automatically advances to the line following the current line, and displays that line. A TYPE command without a range specification is equivalent to the command

TYPE .+1

Examples

TYPE

T %WHOLE

T .

Hints

Note that the TYPE command does not require either "T" or "TYPE" to be included in the command. A range specification alone is interpreted as a TYPE command. A blank line, in turn, is considered an implicit TYPE command without a range specification. For this reason, simple carriage-returns may be used to scroll thru a procedure. For short procedures, this is an excellent mechanism for locating a particular line.

A.3 Range Specifications

A.3.1 The Current Line -- . (period)

Purpose

This range specification indicates that the editor command is to act on the current line.

Format

- .
- 1. A period.

Requirements

None.

Prompts &
Responses

None.

Effects

1. The command acts on the current line.
2. If you entered a period without a command, the current line is typed.

Examples

TYPE .
S/ORIGINAL STRING/REPLACEMENT STRING/ .

Hints

A Type command with a range of <period> types the current line without advancing to the next line.

A Substitute command with a range of <period> replaces all occurrences of the original string on the current line with the replacement string. Without any range, the same substitute command would replace only the first occurrence of the original string

on the line.

A.3.2 The First Line of a Procedure -- %BEGIN

Purpose

This range specification indicates that the editor command is to act on the first line of the procedure.

Format

%BEGIN
%BE

Requirements

None.

Prompts & Responses

None.

Effects

1. The command containing the %BEGIN acts on the first line of the procedure.
2. If you entered %BEGIN without an explicit command, the first line of the procedure is typed.

Examples

TYPE %BEGIN
DELETE %BEGIN

Hints

%BEGIN typed without a command (equivalent to TYPE %BEGIN) is a convenient way to get to the first line of the procedure.

A.3.3 The last line of the procedure -- %END

Purpose

The %END range specifies that the editor is to apply the command to the last line of the procedure.

Format

%END
%E

Requirements

None.

Prompts & Responses

None.

Effects

1. The command acts on the last line of the procedure.
2. If you entered a period without a command, the last line is typed.

Example

INSERT %END

Hints

The command

INSERT %END

is the best way to add new lines after the last line of a procedure.

A.3.4 The Next Occurrence of a String

Purpose

This range specification indicates that the editor command is to be applied to the next line containing a given string.

Format

"string-1"
'string-1'

1. The string may be enclosed in either single or double quotes.

Requirements

When the string is not found between the current line and the end of the procedure, the command has no effect.

Prompts & Responses

A messages is printed if the string is not found.

Effects

1. The command containing the range specification acts on the next line containing the string-1.
2. If the string-1 is not found, the command is aborted. If this happens, the result is just as if the command were not typed.
3. The search for string-1 is case insensitive. An upper case letter matches is lower case equivalent and vice versa.

Examples

```
TYPE "AT BOTTOM"  
  
DELETE 'PRINX YACHTS'
```

Hint

A quoted string without an explicit command is a dandy way to locate and print a particular line.

A.3.5 The whole procedure -- %WHOLE

Purpose

The range %WHOLE specifies that the command is apply to all lines of the procedure.

Format

%WHOLE
%WH

Requirements

None.

Prompts & Responses

None.

Effects

1. The command acts on all lines in the procedure.
2. If you entered %WHOLE without a command, the entire procedure is typed.

Examples

%WHOLE
S/PLATAPUS/PLATYPUS/ %WHOLE

Hints

If you become seriously lost during an edit, type

%WHOLE

to determine the extent of the damage and position the editor on the first line of the procedure. And, if worse comes to worse, the command

REPLACE %WHOLE

is a good way to start over (though you might consider QUIT).

A substitution command with the range %WHOLE changes all occurrences of the original string in the procedure to the replacement string.

A.3.6 The Rest of the Procedure -- %REST

Purpose

This range specification indicates that the editor command is to act on all lines from the current to the last line of the procedure.

Format

%REST
%R

Requirements

None.

Prompts &
Responses

None.

Effects

1. The command acts on the current and remaining lines of the procedure.
2. If %REST is entered without an explicit command, the rest of the lines are typed.

Examples

D %REST

%REST

A.3.7 The preceding lines -- %BEFORE

Purpose

The %BEFORE range specification applies the editor command to all lines from the first to the current line.

Format

%BEFORE
%BEF

Requirements

None.

Prompts & Responses

None.

Effects

1. The command acts on all lines from the top of the procedure to and including the current line.
2. If %BEFORE is entered without an explicit command, the first through the current lines are typed. The editor, however, is positioned to the first line of the procedure after the command.

Examples

S/PRINT ALL/PRINT/ %BEFORE

Hints

Since almost any use of %BEFORE causes the editor to be positioned to the first line of the procedure, %BEFORE is of limited use.

A.3.8 All Lines Containing a Particular String -- %ALL

Purpose

The range %ALL is used to apply an editor command to all lines containing a particular string.

Format

%ALL "string-1"
%ALL 'string-1'

1. The string may be enclosed in either single or double quotes.

Requirements

None.

Prompts &
Responses

No indication is given if no occurrence of string-1 was found.

Effects

1. The command contain the %ALL range acts on all lines containing the string-1.
2. If the range is given without an explicit command, all lines containing the given string are typed.
3. The search is case insensitive.

Examples

```
TYPE %ALL "EDIT-STRING"  
DELETE %ALL 'PRINT "'
```

Hints

The range %ALL can be used to print all occurrences of a particular field in a procedure.

A.3.9 Repeat for a Number of Consecutive Lines -- %FOR

Purpose

This range specification allows a command to be applied to a number of consecutive lines.

Format

```
range-1 %FOR n  
range-1 ; n
```

1. Range-1 must be one of the range specifications: . (period), %BEGIN, quoted or double quoted string.
2. N must be an unsigned, non-zero number.

Requirements

None.

Prompts & Responses

None.

Effects

1. The command is applied to n lines beginning with the line identified by range-1.
2. If the range is entered without an explicit command, the n lines are typed.

Examples

```
T %BEGIN ; 5  
DELETE . %FOR 3
```

A.3.10 Multiple Range Specification -- %AND

Purpose

This range form allows two or more simple range specifications to be given for a single command.

Format

range-1 %AND range-2 ...
range-1 , range-2 ...

1. Range-1, range-2, etc. must be taken from the following list of range specifications: . (period), %BEGIN, %END, quoted or double quoted string.

Requirements

None.

Prompts & Responses

None.

Effects

1. The command is applied to the line identified by range-1, then to the line identified by range-2, etc.
2. Following execution of the command, the editor is positioned at the line identified by range-1, unless that line was deleted. In that case the editor is positioned where the line used to be.
3. If the range is entered without an explicit command, the lines indicated by the various ranges are typed. The editor is now positioned at the line identified by range-1.

Examples

TYPE %BEGIN, %END

TYPE %BEGIN, "FOO"

Hints

The command:

`%BEGIN, "FOO"`

can be used to type the line containing the first occurrence of the string "FOO". The editor, however, is now positioned at the first line of the procedure.

A.3.11 A Fixed Offset from a Given Line -- +

Purpose

This range specification indicates that the editor command is to be applied to a line a fixed number of lines from a given line.

Format

range-1 + n

1. Range-1 must be a range specification take from the list . (period), %BEGIN, and quoted or double quoted string.
2. N must be an unsigned, no-zero number.

Requirements

None.

Prompts &
Responses

None.

Effects

1. The command is applied to the n-th line from the line indicated by range-1.

Examples

TYPE %BEGIN + 3

S /FOO/BAR/ .+1

READER'S COMMENTS

NOTE: This form is for document comments only. **DIGITAL** will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or
Country

Fold Here

Do Not Tear - Fold Here and Staple

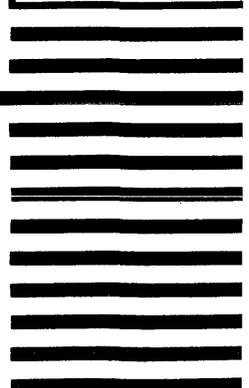
FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
146 Main Street ML 5-5/E39
Maynard, Massachusetts 01754



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

digital

digital equipment corporation