

CA-Easytrieve[®]/

Plus

For OS/390, VSE

CA-Datcom/DB Interface Option Guide

6.2

**COMPUTER[®]
ASSOCIATES**
Software superior by design.

SP3

Release 6.2, May 1996
Updated: April 1999

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

THIS DOCUMENTATION MAY NOT BE COPIED, TRANSFERRED, REPRODUCED, DISCLOSED OR DUPLICATED, IN WHOLE OR IN PART, WITHOUT THE PRIOR WRITTEN CONSENT OF CA. THIS DOCUMENTATION IS PROPRIETARY INFORMATION OF CA AND PROTECTED BY THE COPYRIGHT LAWS OF THE UNITED STATES AND INTERNATIONAL TREATIES.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

© 1996-2000 Computer Associates International, Inc., One Computer Associates Plaza, Islandia, New York 11749. All rights reserved.

All trademarks, trade names, service marks, or logos referenced herein belong to their respective companies.



Contents

Chapter 1: About This Guide

Purpose and Audience	1-1
Organization	1-1
Other CA-Easytrieve/Plus Publications	1-2
Related Publications	1-2
Command Notation	1-3

Chapter 2: Overview

Introduction	2-1
Using the Interface	2-1
Preparatory Work	2-1
Accessing the Database	2-2
Automatic Processing	2-2
Controlled Processing	2-2
How the Interface Works	2-2

Chapter 3: Access Macros

Introduction	3-1
CA-Datcom/DB Access Macros	3-1
Generating Parameter Areas	3-1
Generating Calls	3-2
Macro Synopsis	3-3
%DBUIB - Define User Information Block	3-4
Syntax	3-4
Generated Statements	3-4
%DBRA - Define Request Area	3-5
Syntax	3-5
Generated Statements	3-6

%DBEL - Define Element List	3-6
Syntax	3-7
Generated Statements	3-8
%DBWA - Define Work Area	3-8
Syntax	3-8
Generated Statement	3-9
%DBFILE - Generate FILE Statement	3-9
Syntax	3-9
Generated Statements	3-10
%DBOPEN - Open Tables	3-11
Syntax	3-11
Generated Statements	3-11
%DBCLOSE - Close Tables	3-11
Syntax	3-11
Generated Statements	3-11
%DBLREQ - Direct Request (Long List)	3-12
Syntax	3-12
Generated Statements	3-13
%DBSREQ - Direct Request (Short List)	3-13
Syntax	3-13
Generated Statements	3-14
Examples	3-14
Automatic Processing - All Records	3-14
Macro Generated Statements	3-15
Automatic Processing - Selected Records	3-16
Controlled Processing	3-17
Macro Generated Statements	3-18
Purchase Orders by Line Item	3-20

Index

About This Guide

Purpose and Audience

The CA-Easytrieve/Plus CA-Datacom/DB Interface Option lets you use the power and flexibility of CA-Easytrieve/Plus to access your CA-Datacom/DB database.

To use this guide, you must know how to use CA-Datacom/DB and know the layout of the database that you want CA-Easytrieve/Plus to process.

Organization

This guide contains two additional chapters.

- “Overview” introduces you to the CA-Datacom/DB Interface Option. It provides specific information on what you need to know before using the interface.
- “CA-Datacom/DB Access Macros” describes how to use the CA-Datacom/DB Interface Option. This chapter discusses the CA-Datacom/DB access macros in significant detail.
- The Index provides listings to facilitate references to terms and procedures.

Other CA-Easytrieve/Plus Publications

In addition to this *CA-Easytrieve/Plus Datacom/DB Interface Option Guide*, Computer Associates provides the following CA-Easytrieve/Plus documentation:

Name	Description
<i>CA-Easytrieve/Plus Reference Guide</i>	Contains descriptions of all product features and functions as well as summaries of each CA-Easytrieve/Plus version.
<i>CA-Easytrieve/Plus User Guide</i>	Provides new users with the information they need to become productive quickly. It includes a six-lesson tutorial and a format designed to make the material more interesting and easier to comprehend.
<i>CA-Easytrieve/Plus Installation Guide</i>	Describes the process of installing and tuning the CA-Easytrieve/Plus system.
<i>CA-Easytrieve/Plus Application Guide</i>	Describes basic syntax (a subset of the syntax in the <i>CA-Easytrieve/Plus Reference Guide</i>) and operation, and provides a series of actual applications from single examples to full systems. The <i>Application Guide</i> is an excellent tool for the business-oriented professional.
<i>CA-Easytrieve/Plus Extended Reporting Facility Guide</i>	Provides support of extended reporting capabilities for Impact Dot, Ink Jet, and Electro Photographic printers.
<i>CA-Easytrieve/Plus CA-ACTIVATOR Supplement</i>	Explains how to install and maintain CA-Easytrieve/Plus using CA-Activator.
<i>CA-Easytrieve/Plus Interface Option Guides</i>	Short guides available for users of various system options. These consist of manuals for IMS/DLI processing, CA-IDMS and IDD processing, TOTAL processing, SQL processing, SUPRA processing, and other CA-Easytrieve/Plus options.

Related Publications

The following publication, produced by Computer Associates, is either referenced in this documentation or is recommended reading:

- CA-Datacom/DB Programmer Guide

Command Notation

This guide uses the following command notation:

Notation	Meaning
UPPERCASE	Specific filenames and keywords appear in uppercase when described in text.
bold	An actual name or literal as it appears in a statement.
lowercase and <i>lowercase italic</i>	Used in syntax descriptions where lowercase entries must be coded as shown, and <i>italic</i> entries represent variable information you must supply.
User Entry	Represents text that you type at your terminal.
symbols - and +	Used ahead of program options and must be coded as shown.
other symbols	Must be coded as shown (for example, @ and :).
case in options	Options must be coded as shown. For example, -a is not the same as -A.
<i>italic</i> options	Used in discussion of specific options, <i>italic</i> represents variable information. For example, in <i>wn</i> , <i>n</i> represents a numeric 1-digit variable.
[]	Identifies optional keywords or parameters.
	Separates alternate keywords or parameters. Choose one.
...	The preceding item or groups of items can be repeated more than once.

Introduction

CA-Easytrieve/Plus is an information retrieval and data management system that simplifies complex programming tasks. It enables you to quickly and efficiently produce reports, extract data, and maintain files. The CA-Easytrieve/Plus CA-Datacom/DB Interface Option lets you use the power and flexibility of CA-Easytrieve/Plus to access your CA-Datacom/DB database.

Using the Interface

To use this interface, you must know how to use CA-Datacom/DB and know the layout of the database that you want CA-Easytrieve/Plus to process. Specifically, you must know:

- The purpose and syntax of the commands common to CA-Datacom/DB.
- The contents of the database.
- How the data is organized within the database.
- What elements of data you can read and/or update.

Such knowledge makes the task of accessing the database much simpler.

Preparatory Work

Preparatory work by your database administrator significantly reduces the effort of writing the CA-Easytrieve/Plus programs that access CA-Datacom/DB databases. The database administrator should place into the CA-Easytrieve/Plus macro library the data definition statements necessary to access each element in the database. Keeping these element field definition statements within the control of the database administrator can greatly reduce the number of programming errors associated with database processing. The examples in Chapter 3, "CA-Datacom/DB Access Macros," use the test database supplied with CA-Datacom/DB.

Accessing the Database

You can access the database in one of two ways:

- Automatic processing
- Controlled processing using control macros similar to statements used in COBOL.

Automatic Processing

With automatic processing, you sequentially retrieve all or part of a table within any defined CA-Datacom/DB key. Automatic processing uses the EXIT parameter of the FILE statement. The EXIT parameter of the FILE statement automatically links with the assembler routine that enables CA-Easytrieve/Plus and CA-Datacom/DB to communicate. The name of this routine is EZTPDTCM.

Controlled Processing

With controlled processing, you can perform any retrieval process (for example, sequential access and/or random access) on all or part of a table available to CA-Easytrieve/Plus. With controlled processing, you use the CALL statement to link to the EZTPDTCM routine that enables CA-Easytrieve/Plus and CA-Datacom/DB to communicate.

Note: Never mix automatic and controlled processing in the same CA-Easytrieve/Plus job activity. This is because automatic processing automatically performs functions that are not automatically performed under controlled processing.

How the Interface Works

CA-Easytrieve/Plus communicates with CA-Datacom/DB by means of the User Requirements Table and the CA-Datacom/DB interface module, EZTPDTCM. The CALL statement or the EXIT parameter of the FILE statement invokes the EZTPDTCM routine. The EZTPDTCM routine links CA-Easytrieve/Plus and CA-Datacom/DB.

The EZTPDTCM routine communicates with a driver routine named EZTPDRVR (supplied by Computer Associates). You must link-edit EZTPDRVR with the User Requirement Tables produced by your database administrator. The installation instructions for this interface are discussed in the *CA-Easytrieve/Plus Installation Guide*.

Introduction

This chapter provides an overview of the CA-Datcom/DB Access Macros available to the interface, then discusses each macro in detail.

CA-Datcom/DB Access Macros

The CA-Datcom/DB Interface Option uses a series of CA-Easytrieve/Plus macros that access CA-Datcom/DB and eliminate the need for much of the user coding. These macros fall into two categories:

- Macros that generate parameter areas that CA-Datcom/DB requires.
- Macros that generate the appropriate calls to CA-Datcom/DB.

Generating Parameter Areas

The following macros generate parameter areas that CA-Datcom/DB requires. You must code these macros in the library section of each CA-Easytrieve/Plus program that accesses CA-Datcom/DB:

- **%DBUIB** defines the User Information Block (UIB) for communication with CA-Datcom/DB. Use this macro for automatic or controlled processing.
- **%DBRA** defines the Request Area for communication with CA-Datcom/DB. Use this macro for automatic or controlled processing.
- **%DBEL** defines the Element List for communication with CA-Datcom/DB. Use this macro for automatic or controlled processing.
- **%DBWA** defines the Work Area for communication with CA-Datcom/DB. This macro is used only for controlled processing, since the %DBFILE macro automatically generates an equivalent work area.

Refer to the macro synopsis table for a synopsis of each macro.

Note: The %DBRA, %DBEL, and %DBWA macros each define parameter areas that CA-Datcom/DB requires. For automatic processing, the prefixes to all three of these macros must be identical. A *matched set* occurs when all three prefixes are the same.

Generating Calls

The following macros generate the appropriate calls to CA-Datcom/DB.

- **%DBFILE** generates the FILE statement that you use during automatic processing. Code this macro immediately before the Job Statement in your CA-Easytrieve/Plus program. Use this macro for automatic processing only.
- **%DBOPEN** opens the CA-Datcom/DB tables for further processing. Use this macro for controlled processing only.
- **%DBCLOSE** closes all open CA-Datcom/DB tables. Use this macro for controlled processing only.
- **%DBLREQ (long parameter list)** provides the direct request interface to CA-Datcom/DB using all four parameter areas (UIB, RA, EL, and WA). Use this macro for controlled processing only.
- **%DBSREQ (short parameter list)** provides the direct request interface to CA-Datcom/DB using just the UIB and RA parameter areas. Use this macro for controlled processing only.

Refer to the macro synopsis table for a synopsis of each macro.

Macro Synopsis

The following table is a reference tool that lists information about each macro.

Macro	Function	Type of Processing	Where Coded in CA-Easytrieve/Plus Program
%DBCLOSE	Close CA-Datacom/DB tables	Controlled only	Activity section
%DBEL	Define EL	Automatic or controlled	Library section
%DBFILE	Define WA and generate FILE statement	Automatic only	Library section
%DBLREQ	Provide Direct Request Interface to CA-Datacom/DB using all four parameter areas	Controlled only	Activity section
%DBOPEN	Open CA-Datacom/DB tables	Controlled only	Activity section
%DBRA	Define RA	Automatic or controlled	Library section
%DBSREQ	Provide Direct Request Interface to CA-Datacom/DB using UIB and RA parameter areas	Controlled only	Activity section
%DBUIB	Define UIB	Automatic or controlled	Library section
%DBWA	Define WA	Controlled only	Library section

%DBUIB - Define User Information Block

Use the %DBUIB macro to define the User Information Block (UIB) for communication with CA-Datcom/DB. The 40-byte required UIB area identifies the requester of CA-Datcom/DB services. Use this macro for both automatic and controlled processing.

See the *CA-Datcom/DB Programmer Guide* for more information about other uses of the UIB.

Syntax

```
%DBUIB literal urt-name
```

Both parameters are required for this macro.

literal

Literal is the eight-character name of the current CA-Easytrieve/Plus program. CA-Datcom/DB does not require you to enter a value for this field. However, you should include your CA-Easytrieve/Plus program name here to aid you in debugging your application.

urt-name

Urt-name is the eight-character name of the link-edited URT/EZTPDRVR module prepared by the database administrator for your use. The urt-name is located outside the normal bounds of the UIB as defined by DATACOM. Including this field causes no problems for any of the DATACOM requirements.

A URT/EZTPDRVR must be generated for each table accessed by your application. For instructions to generate the URT, refer to Install Job 13 in the *CA-Easytrieve/Plus Installation Guide*.

Generated Statements

The %DBUIB macro generates the following statements:

```
DEFINE USER-INFO S 40 A  
DEFINE USER-PGM USER-INFO      8 A VALUE 'literal'  
DEFINE URT-NAME  USER-INFO +32  8 A VALUE 'urt-name'
```

Only one UIB is required per program, regardless of the number of CA-Datcom/DB tables you access in the program.

%DBRA - Define Request Area

The %DBRA macro statement defines the Request Area (RA) within your CA-Easytrieve/Plus program. This 286-byte, required Request Area, is used to specify requests made to CA-Datcom/DB and to test the results. CA-Datcom/DB uses the Request Area to save control information between related requests. A program must have an active Request Area for each CA-Datcom/DB table you access. The program can also use multiple Request Areas for the same table. Use this macro for either automatic or controlled processing.

See the *CA-Datcom/DB Programmer Guide* for more information about the full definition and uses of the Request Area.

Syntax

```
%DBRA prfx tbl-nm ky-nm [DBID db-id-no]
```

The *prfx*, *tbl-nm*, and *ky-nm* parameters are required. The *db-id-no* parameter is optional.

prfx

Prfx is the prefix for the CA-Datcom/DB access set you are using to access the CA-Datcom/DB table. This unique prefix enables multiple access paths into the same CA-Datcom/DB table and must match the prefix you use for the %DBEL and %DBWA macros.

Since the interface uses *prfx* to generate other field names, *prfx* must follow CA-Easytrieve/Plus naming conventions and be no longer than 30 characters when you use controlled processing. When you use this macro with automatic input, *prfx* can only be eight characters long for OS/390 and seven characters long for VSE. This is because the *prfx* is also used as a filename.

The %DBRA macro loads *prfx* into the variable *prfx-PREFIX* to aid the interface in identifying the RA. *Prfx-PREFIX* is located outside the normal bounds of the RA, as defined by DATACOM. Therefore, *prfx-PREFIX* causes no problems for any of the CA-Datcom requirements.

tbl-nm

Tbl-nm is the three-character table name you assign in CA-Datcom/DB to the set of records you are accessing. The *tbl-nm* parameter must match a valid table name in the CA-Datcom/DB data dictionary. If not, all requests that use the Request Area fail.

ky-nm

Ky-nm is the five-character name of the key you use to traverse the table. The *ky-nm* parameter must match a valid name in the CA-Datcom/DB data dictionary.

db-id-no

Db-id-no is the optional binary value for a specific database ID. When you use the SYNONYM=YES parameter in your URT, you must code the *db-id-no* parameter to address the correct database.

Generated Statements

The %DBRA macro generates the following statements:

```
DEFINE prfx-REQ-AREA S 1 A OCCURS 284
DEFINE prfx-CMND      prfx-REQ-AREA      5 A
DEFINE prfx-TBL-NM    prfx-REQ-AREA      +5 3 A VALUE 'tbl-nm'
DEFINE prfx-KEY-NM    prfx-REQ-AREA      +8 5 A VALUE 'ky-nm'
DEFINE prfx-RET-CD    prfx-REQ-AREA      +13 2 A
DEFINE prfx-INT-RC    prfx-REQ-AREA      +15 1 B VALUE 0
DEFINE prfx-DB-ID     prfx-REQ-AREA      +16 2 B VALUE db-id-no
DEFINE prfx-REC-ID    prfx-REQ-AREA      +18 7 A
DEFINE prfx-TBL-ID    prfx-REQ-AREA      +18 2 B VALUE 0
DEFINE prfx-BLK       prfx-REQ-AREA      +20 4 B VALUE 0
DEFINE prfx-REC       prfx-REQ-AREA      +24 1 B VALUE 0
DEFINE prfx-RQA-ERR   prfx-REQ-AREA      +26 4 A
DEFINE prfx-RQA-SEC   prfx-REQ-AREA      +26 1 A
DEFINE prfx-RQA-NTRY  prfx-REQ-AREA      +27 3 N VALUE 0
DEFINE prfx-SET-CNT   prfx-REQ-AREA      +40 4 B VALUE 0
DEFINE prfx-SET-NBR   prfx-REQ-AREA      +44 4 B VALUE 0
DEFINE prfx-CNT-MAX   prfx-REQ-AREA      +50 2 B VALUE 0
DEFINE prfx-IO-CNT    prfx-REQ-AREA      +52 2 B VALUE 0
DEFINE prfx-SKP-CNT   prfx-REQ-AREA      +71 2 B VALUE 0
DEFINE prfx-KEY-VAL   prfx-REQ-AREA      +76 180 A
DEFINE prfx-PREFIX    prfx-REQ-AREA      +256 28 A VALUE 'prfx'
```

%DBEL - Define Element List

The %DBEL macro statement defines the Element List (EL) for communication with CA-Datcom/DB. This optional area controls the data elements you reference. Use this macro for either automatic or controlled processing.

See the *CA-Datcom/DB Programmer Guide* for more information about the full definition and use of the Element List.

Syntax

```
%DBEL prfx elmt-nm [elmt-nm ...]
```

prfx

Prfx is the prefix for the CA-Datacom/DB access set. It must match the prefix you use for the %DBRA and %DBWA macros. *Prfx* must follow CA-Easytrieve/Plus naming conventions and be no longer than 30 characters when you use controlled processing. When you use this macro with automatic input, *prfx* can only be eight characters long for OS/390 and seven characters long for VSE. This is because the *prfx* is also used as a filename.

elmt-nm

Elmt-nm is the data element that you retrieve, update, or add to the database. The format of the *elmt-nm* parameter is:

```
EEEEES
```

Where *EEEE* is the element name as defined in the data dictionary, and *S* is the security code defined for that element.

- If the element name does not have a security code, you do not have to enter it.
- If the element name is less than five characters long, and there is a security code, you must enter the element name in quotes with filling blanks. For example:

```
ABC 'D'
```

In this example, *ABC* is the element name and *D* is the security code.

Since the element name is three characters long, there are two blanks between *C* and *D*.

You must enter at least one element name when you use this macro. You can enter as many as 16 element names, depending on the needs of your program.

Generated Statements

The %DBEL macro generates the following statements. In this example, the macro increments the offsets (+6, +12, and so forth) by six and defines the length as seven. As a result, the macro generates one blank after the *last* element name (*elmt-nm*). To avoid an error, CA-Easytrieve/Plus must see a blank after the last *elmt-nm* and not a null.

```
DEFINE prfx-ELEM-LIST S 101 A
DEFINE prfx-ELM01 prfx-ELEM-LIST      7 A VALUE 'elmt-nm '
DEFINE prfx-ELM02 prfx-ELEM-LIST   +6 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM03 prfx-ELEM-LIST  +12 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM04 prfx-ELEM-LIST  +18 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM05 prfx-ELEM-LIST  +24 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM06 prfx-ELEM-LIST  +30 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM07 prfx-ELEM-LIST  +36 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM08 prfx-ELEM-LIST  +42 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM09 prfx-ELEM-LIST  +48 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM10 prfx-ELEM-LIST  +54 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM11 prfx-ELEM-LIST  +60 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM12 prfx-ELEM-LIST  +66 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM13 prfx-ELEM-LIST  +72 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM14 prfx-ELEM-LIST  +78 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM15 prfx-ELEM-LIST  +84 7 A VALUE 'elmt-nm '
DEFINE prfx-ELM16 prfx-ELEM-LIST  +90 7 A VALUE 'elmt-nm '
```

%DBWA - Define Work Area

The %DBWA macro statement defines the Work Area (WA) for communication with CA-Datcom/DB. You can use the optional Work Area to provide a local working storage area in which to manipulate the data elements listed in the matching Element List. Use the %DBWA macro for controlled processing requests only.

See the *CA-Datcom/DB Programmer Guide* for more information about the full definition and uses of the Work Area.

Syntax

%DBWA prfx wrk-area-sz

Both parameters are required.

prfx

Prfx is the prefix you use for the CA-Datcom/DB access set. It must match the prefix you use for the %DBRA and %DBEL macros. *Prfx* must follow CA-Easytrieve/Plus naming conventions and be no longer than 30 characters.

wrk-area-sz

Wrk-area-sz is the length of the Work Area that you define to contain the elements listed in the Element List. It is a numeric value from 1 to 32767. The size must be large enough to fit the combined lengths of all the elements listed. If the size is too small, retrieved elements overlay whatever follows the Work Area in storage, such as CA-Easytrieve/Plus code.

Generated Statement

The %DBWA macro generates the following statement:

```
DEFINE prfx-WORK-AREA W 1 A OCCURS wrk-area-sz
```

You must provide working storage definitions to redefine the Work Area with the appropriate field definitions. The field definitions describe the data elements that your CA-Easytrieve/Plus program manipulates. The example in of controlled processing shows such a working storage definition with the %DBWA macro.

%DBFILE - Generate FILE Statement

The %DBFILE macro statement generates the FILE statement that you use during automatic processing. Use this macro for automatic processing only.

Syntax

```
%DBFILE prfx wrk-area-sz [START strt-val]
```

prfx

Prfx is the prefix associated with the Request Area and Element List. *Prfx* must follow CA-Easytrieve/Plus naming conventions. *Prfx* can only be eight characters long for OS/390 and seven characters long for VSE. This is because the *prfx* is also used as a filename.

The interface uses *prfx* to tailor the generated code which in turn establishes the appropriate CA-Datcom/DB parameter list. The %DBUIB, %DBRA, and %DBEL macros generate the required data areas that communicate with CA-Datcom/DB.

wrk-area-sz

Wrk-area-sz is the length of the Work Area that you define to contain the elements listed in the Element List. The size of this work area must be large enough to fit the combined lengths of all the elements listed. If the size is too small, retrieval commands overlay whatever follows the Work Area in storage, such as CA-Easytrieve/Plus code.

strt-val

Strt-val is an optional value that you can use for skip sequential processing. When you omit *strt-val*, CA-Easytrieve/Plus processes the table beginning with the first key value found. When you supply *strt-val*, CA-Easytrieve/Plus processes the table beginning with the first key value found that is equal to or greater than the value supplied. If the *strt-val* contains embedded blanks, you must enclose it in triple quotes.

For example:

```
START   '''Smith   91700'''
```

In this example, there are three blanks between Smith and 91700. Therefore, you must place triple quotes around them (you have to place the triple quotes, even if there is one blank). If you did not, CA-Easytrieve/Plus sees a blank after Smith and stops processing the parameter.

Generated Statements

The %DBFILE macro generates the following statements:

```
DEFINE prfx-KEY-VAL2 prfx-REQ-AREA +76 180 A VALUE 'strt-val'  
FILE prfx                                     +  
      WORKAREA(wrk-area-sz)                   +  
      EXIT (EZTPDTCM USING (USER-INFO        +  
                           prfx-REQ-AREA    +  
                           prfx-ELEM-LIST))  
      WORK-AREA                               1 1 A OCCURS wrk-area-sz
```

WORK-AREA

CA-Datacom/DB needs the *WORK-AREA* defined after the *FILE* statement for data transfer. It is defined here as a file field in order to make it possible to use automatic processing. See the *CA-Datacom/DB Programmer Guide* for more information about the full definition and uses of the Work Area.

You must provide *FILE* field definitions to redefine the *WORK-AREA* with the appropriate field definitions that describe the data elements that your CA-Easytrieve/Plus program manipulates. The examples at the end of this chapter show such a working storage definition following the %DBFILE macro.

%DBOPEN - Open Tables

The %DBOPEN macro statement opens the CA-Datcom/DB tables for further processing. You must code this macro in the activity section of the CA-Easytrieve/Plus program (before any other CA-Datcom/DB commands). Use this macro for controlled processing only.

Syntax

%DBOPEN

Generated Statements

The %DBOPEN macro generates the following statements:

```
DBOPEN.  
  PROC  
    CALL EZTPDTCM USING(USER-INFO 'OPEN')  
  END-PROC
```

You must invoke the %DBOPEN macro (that generates the DBOPEN PROCEDURE) once in each CA-Easytrieve/Plus program that uses controlled processing. Automatic processing does not require the %DBOPEN macro, since the FILE EXIT routine automatically opens the tables.

%DBCLOSE - Close Tables

The %DBCLOSE macro statement closes all open CA-Datcom/DB tables. The %DBCLOSE macro has no parameters. Use this macro for controlled processing only.

Syntax

%DBCLOSE

Generated Statements

The %DBCLOSE macro generates the following statements:

```
DBCLOSE.  
  PROC  
    CALL EZTPDTCM USING(USER-INFO 'CLOSE')  
  END-PROC
```

You must invoke the %DBCLOSE macro (that generates the DBCLOSE PROCEDURE) once at the end of each CA-Easytrieve/Plus program that uses controlled processing. The DBCLOSE PROCEDURE closes all open CA-Datcom/DB tables.

%DBLREQ - Direct Request (Long List)

The %DBLREQ (long parameter list) macro statement provides the direct request interface to CA-Datcom/DB using all four DATACOM parameter areas (UIB, RA, EL, WA). Use %DBLREQ only for controlled processing requests into CA-Datcom/DB.

Syntax

```
%DBLREQ prfx cmnd-cd [EL el-prfx] [WA wa-prfx]
```

The first two parameters are required for this macro. The *el-prfx* and *wa-prfx* parameters are optional.

prfx

Prfx is the prefix previously associated with a set of RA, EL, and WA areas. *Prfx* must follow CA-Easytrieve/Plus naming conventions and can be no longer than 30 characters. The prefix tailors the generated code that accesses the appropriate CA-Datcom/DB parameter areas for the request.

cmnd-cd

Cmnd-cd is a valid five-character CA-Datcom/DB command code. Use *cmnd-cd* to specify the operation to be performed. See the *CA-Datcom/DB Programmer Guide* for a list of valid command codes.

el-prfx

The optional *el-prfx* parameter points to an element list other than the one you specify in *prfx*. When you specify this parameter, DATACOM accesses the element list that *el-prfx* points to and ignores the value in *prfx*. If you do not enter a value for *el-prfx*, then the *el-prfx* defaults to the value in *prfx*. The *el-prfx* parameter enables you to communicate with CA-Datcom/DB using the exact data areas that provide you with the optimum access path.

wa-prfx

The optional *wa-prfx* parameter points to a work area other than the one you specify in *prfx*. When you specify this parameter, CA-Datcom/DB accesses the work area that *wa-prfx* points to and ignores the value in *prfx*. If you do not enter a value for *wa-prfx*, then the *wa-prfx* defaults to the value in *prfx*. The *wa-prfx* parameter enables you to communicate with CA-Datcom/DB using the exact data areas that provide you with the optimum access path.

Generated Statements

The %DBLREQ macro generates the following statements:

```
prfx-CMD = 'cmd-cd'  
CALL EZTPDTCM USING(USER-INFO      +  
                    prfx-REQ-AREA  +  
                    wa-prfx-WORK-AREA +  
                    el-prfx-ELEM-LIST)
```

After the call, the result of the operation is found in *prfx-RET-CD*.

%DBSREQ - Direct Request (Short List)

The %DBSREQ (short parameter list) macro statement provides the direct request interface into CA-Datcom/DB using the UIB and RA parameter areas only. This macro saves programming time because it eliminates the need to code the WA and EL areas. Use this macro with only those commands that require the UIB and WA areas (such as in LOCATE, DELETE, and RELEASE). See the *CA-Datcom/DB Programmer Guide* for a complete list of these commands.

Use the %DBSREQ macro only for controlled processing requests into CA-Datcom/DB.

Syntax

```
%DBSREQ prfx cmd-cd
```

Both parameters are required for this macro.

prfx

Prfx is the prefix previously associated with a set of RA, EL, and WA areas. *Prfx* must follow CA-Easytrieve/Plus naming conventions and can be no longer than 30 characters. Use this prefix to tailor the generated code that accesses the appropriate CA-Datcom/DB parameter areas for the request.

cmd-cd

Cmd-cd is a valid five-character CA-Datcom/DB command code. Use *cmd-cd* to specify an operation to be performed.

Generated Statements

The %DBSREQ macro generates the following statements:

```
prfx-CMD = 'cmd-cd'  
CALL EZTPDTM USING(USER-INFO      +  
                    prfx-REQ-AREA)
```

After the call, the results of the operation is found in *prfx-RET-CD*.

Examples

The following CA-Easytrieve/Plus programs with CA-Datcom/DB interface macro examples are based upon the sample database and programs provided with CA-Datcom/DB and illustrated in the *CA-Datcom/DB Programmer Guide*.

Automatic Processing - All Records

In the following example of automatic processing, the program prints a report. This report lists every record in the purchase order header (POH) table:

```
1:  %DBUIB EZT+DB MYURT  
2:  %DBRA ABC POH POLI  
3:  %DBEL ABC PO LI VENDR TPVAL LATE  
4:  %DBFILE ABC 20  
5:  PO      1 5 A  
6:  LI      6 3 A  
7:  VENDR   9 3 A  
8:  TPVAL  12 8 N MASK 'ZZZ,ZZ9.99'  
9:  LATE    20 1 A  
10: *  
11: JOB NAME ACCESS-ALL-ROWS INPUT ABC  
12: PRINT ABC-RPT  
13: *  
14: REPORT ABC-RPT LINESIZE 80  
15: LINE PO VENDR TPVAL LATE
```

Line 1 defines the standard User Information Block (UIB) parameter area for CA-Datcom/DB calls. Only one of these is required in any program.

Lines 2 and 3 define two other standard parameter areas: the Request Area (RA) and the Element List (EL).

Line 4 defines the Work Area (WA) size, sets up the automatic processing file, and defines the starting value for the sequential scan (LOW-VALUES is the default).

The %DBFILE macro automatically generates the FILE statement and EXIT parameter, causing the CA-Datcom/DB files to be properly opened, accessed, and closed.

Note: The RA, EL, and WA are matched together by the prefix ABC.

Lines 5-9 define the field structure of the WA.

Lines 11-15 are the typical CA-Easytrieve/Plus statements required to produce a sequential report.

Macro Generated Statements

The example of automatic processing is now shown with all statements that the macros generate. The statements generated by the macros are identified with an asterisk in the left column:

```
%DBUIB EZT+DB MYURT
* DEFINE USER-INFO S                40 A
* DEFINE USER-PGM  USER-INFO      8 A VALUE 'EZT+DB'
* DEFINE URT-NAME  USER-INFO +32 8 A VALUE 'MYURT'
%DBRA ABC POH POLI
* DEFINE ABC-REQ-AREA S                284 A
* DEFINE ABC-CMND   ABC-REQ-AREA      5 A
* DEFINE ABC-TBL-NM ABC-REQ-AREA +5 3 A VALUE 'POH'
* DEFINE ABC-KEY-NM ABC-REQ-AREA +8 5 A VALUE 'POLI'
* DEFINE ABC-RET-CD ABC-REQ-AREA +13 2 A
* DEFINE ABC-INT-RET-CD ABC-REQ-AREA +15 1 B VALUE 0
* DEFINE ABC-DB-ID  ABC-REQ-AREA +16 2 B VALUE 0
* DEFINE ABC-REC-ID ABC-REQ-AREA +18 7 A
* DEFINE ABC-TBL-ID ABC-REQ-AREA +18 2 B VALUE 0
* DEFINE ABC-BLK    ABC-REQ-AREA +20 4 B VALUE 0
* DEFINE ABC-REC     ABC-REQ-AREA +24 1 B VALUE 0
* DEFINE ABC-RQA-ERR ABC-REQ-AREA +26 4 A
* DEFINE ABC-RQA-SEC ABC-REQ-AREA +26 1 A
* DEFINE ABC-RQA-ENTRY ABC-REQ-AREA +27 3 N VALUE 0
* DEFINE ABC-SET-CNT ABC-REQ-AREA +40 4 B VALUE 0
* DEFINE ABC-SET-NBR ABC-REQ-AREA +44 4 B VALUE 0
* DEFINE ABC-CNT-MAX ABC-REQ-AREA +50 2 B VALUE 0
* DEFINE ABC-IO-CNT  ABC-REQ-AREA +52 2 B VALUE 0
* DEFINE ABC-SKP-CNT ABC-REQ-AREA +71 2 B VALUE 0
* DEFINE ABC-KEY-VAL ABC-REQ-AREA +76 180 A
* DEFINE ABC-WHO-AM-I ABC-REQ-AREA +256 28 A VALUE 'ABC'
%DBEL ABC PO LI VENDR TPVAL LATE
* DEFINE ABC-ELEM-LIST S                101 A
* DEFINE ABC-ELM01  ABC-ELEM-LIST      7 A VALUE 'PO '
* DEFINE ABC-ELM02  ABC-ELEM-LIST +6 7 A VALUE 'LI '
* DEFINE ABC-ELM03  ABC-ELEM-LIST +12 7 A VALUE 'VENDR '
* DEFINE ABC-ELM04  ABC-ELEM-LIST +18 7 A VALUE 'TPVAL '
* DEFINE ABC-ELM05  ABC-ELEM-LIST +24 7 A VALUE 'LATE '
* DEFINE ABC-ELM06  ABC-ELEM-LIST +30 7 A VALUE ' '
* DEFINE ABC-ELM07  ABC-ELEM-LIST +36 7 A VALUE ' '
* DEFINE ABC-ELM08  ABC-ELEM-LIST +42 7 A VALUE ' '
* DEFINE ABC-ELM09  ABC-ELEM-LIST +48 7 A VALUE ' '
```

```

* DEFINE ABC-ELM10      ABC-ELEM-LIST +54  7 A VALUE ' '
* DEFINE ABC-ELM11      ABC-ELEM-LIST +60  7 A VALUE ' '
* DEFINE ABC-ELM12      ABC-ELEM-LIST +66  7 A VALUE ' '
* DEFINE ABC-ELM13      ABC-ELEM-LIST +72  7 A VALUE ' '
* DEFINE ABC-ELM14      ABC-ELEM-LIST +78  7 A VALUE ' '
* DEFINE ABC-ELM15      ABC-ELEM-LIST +84  7 A VALUE ' '
* DEFINE ABC-ELM16      ABC-ELEM-LIST +90  7 A VALUE ' '
%DBFILE ABC 20
* DEFINE ABC-KEY-VAL2 ABC-REQ-AREA +76 180 A VALUE '00000000'
* FILE ABC
*   WORKAREA(20)
*   EXIT (EZTPDTCM USING (USER-INFO
*                       ABC-REQ-AREA
*                       ABC-ELEM-LIST)
*   WORK-AREA      1  1  A      OCCURS  20
PO      1  5  A
LI      6  3  A
VENDR   9  3  A
TPVAL   12 8  N   MASK 'ZZZ,ZZ9.99'
LATE    20  1  A
JOB NAME ACCESS-ALL-ROWS INPUT ABC
PRINT ABC-RPT
REPORT ABC-RPT LINESIZE 80
LINE PO VENDR TPVAL LATE

```

Automatic Processing - Selected Records

The following program prints all purchase orders between PO numbers 1000 and 1999. The purchase order records reside in the purchase order header (POH) table.

```

1 %DBUIB EZT+DB MYURT
2 %DBRA DEF POH POLI
3 %DBEL DEF PO LI VENDR TPVAL LATE
4 %DBFILE DEF 20 START ''01000''
5 PO      1  5  A
6 LI      6  3  A
7 VENDR   9  3  A
8 TPVAL   12 8  N   MASK 'ZZZ,ZZ9.99'
9 LATE    20  1  A
10 *
11 JOB NAME ACCESS-SOME-ROWS INPUT DEF
12 IF PO > '01999'
13 STOP
14 END-IF
15 PRINT DEF-RPT
16 *
17 REPORT DEF-RPT LINESIZE 80
18 LINE PO VENDR TPVAL LATE

```

Line 1 defines the standard User Information Block (UIB) parameter area for CA-Datcom/DB calls. Only one of these is required in any program.

Lines 2 and 3 define two other standard parameter areas: the Request Area (RA) and the Element List (EL).

Line 4 defines the Work Area (WA) size, sets up the automatic processing file, and defines the starting value for the sequential scan ('01000').

The %DBFILE macro automatically generates the FILE statement and EXIT parameter causing the CA-Datcom/DB files to be properly opened, accessed and closed.

Note: The RA, EL, and WA are matched together by the prefix DEF.

Lines 5-9 define the field structure of the WA.

Lines 11-18 are the typical CA-Easytrieve/Plus statements required to produce a report for the desired range of records.

Controlled Processing

The following program is an example of controlled processing. Like the first example program, it prints a report of all records in the purchase order header (POH) table. This program duplicates the results of the first example under Automatic Processing.

```

1  %DBUIB EZT+DB MYURT
2  %DBRA GHI POH POLI
3  %DBEL GHI PO LI VENDR TPVAL LATE
4  %DBWA GHI 20
5  DEFINE GHI-PO    GHI-WORK-AREA    5  A
6  DEFINE GHI-LI    GHI-WORK-AREA  +5  3  A
7  DEFINE GHI-VENDR GHI-WORK-AREA  +8  3  A
8  DEFINE GHI-TPVAL GHI-WORK-AREA +11  8  N  MASK 'ZZZ,ZZ9.99'
9  DEFINE GHI-LATE  GHI-WORK-AREA +19  1  A
10 *
11 JOB NAME ALL-RECORDS INPUT NULL START STARTUP FINISH DBCLOSE
12 PRINT GHI-RPT      . * Print this record
13 %DBLREQ GHI REDNX  . * Read next record
14 IF GHI-RET-CD = '14' . * Have we reached end of file?
15     STOP           . * Yes; we're done
16 END-IF
17 STARTUP.
18 PROC              . * Procedure for start-up processing
19 PERFORM DBOPEN    . * Open the CA-Datcom/DB file
20 GHI-KEY-VAL = X'0000000000' . * Set the first key value
21 %DBSREQ GHI LOCKY . * Locate key => start value
22 IF GHI-RET-CD = '14' . * Did we not find one?
23     STOP           . * Then we must not have any records
24 END-IF
25 %DBLREQ GHI REDLE . * Read the located record
26 END-PROC
27 %DBOPEN
28 %DBCLOSE
29 *
30 REPORT GHI-RPT LINESIZE 80
31 LINE GHI-PO GHI-VENDR GHI-TPVAL GHI-LATE

```

Line 1 defines the standard User Information Block (UIB) parameter area for CA-Datcom/DB calls. Only one of these is required in any program.

Lines 2-4 define the other three standard parameter areas - the Request Area (RA), the Element List (EL), and the Work Area (WA).

Note: The RA, EL, and WA are matched together by the prefix GHI.

Lines 5-9 define the field structure of the WA.

Line 11 is a JOB statement which specifies the initializing and terminating procedures you use. You must always code these procedures to point to the %DBOPEN or %DBCLOSE macros or to PROCedures that invoke them.

Lines 12-16 contain the logic to print the current record and read the next record in the table, checking for end of file.

Lines 17-26 perform the required initializing operations. This operation calls the DBOPEN PROCEDURE, sets the initial key value, and reads a record.

Line 27 contains the %DBOPEN macro that opens the required DATACOM tables.

Line 28 contains the %DBCLOSE macro that closes the required DATACOM tables.

Lines 30-31 contain the typical CA-Easytrieve/Plus statements needed to produce a report.

Macro Generated Statements

The example of controlled processing is now shown with all statements that the macros generate. The statements generated by the macros are identified with an asterisk in the left column:

```
%DBUIB EZT+DB MYURT
* DEFINE USER-INFO S          40 A
* DEFINE USER-PGM USER-INFO  8 A VALUE 'EZT+DB'
* DEFINE URT-NAME USER-INFO +32 8 A VALUE 'MYURT'
%DBRA GHI POH POLI
* DEFINE GHI-REQ-AREA S          284 A
* DEFINE GHI-CMND GHI-REQ-AREA  5 A
* DEFINE GHI-TBL-NM GHI-REQ-AREA +5 3 A VALUE 'POH'
* DEFINE GHI-KEY-NM GHI-REQ-AREA +8 5 A VALUE 'POLI'
* DEFINE GHI-RET-CD GHI-REQ-AREA +13 2 A
* DEFINE GHI-INT-RET-CD GHI-REQ-AREA +15 1 B VALUE 0
* DEFINE GHI-DB-ID GHI-REQ-AREA +16 2 B VALUE 0
* DEFINE GHI-REC-ID GHI-REQ-AREA +18 7 A
* DEFINE GHI-TBL-ID GHI-REQ-AREA +18 2 B VALUE 0
* DEFINE GHI-BLK GHI-REQ-AREA +20 4 B VALUE 0
* DEFINE GHI-REC GHI-REQ-AREA +24 1 B VALUE 0
* DEFINE GHI-RQA-ERR GHI-REQ-AREA +26 4 A
* DEFINE GHI-RQA-SEC GHI-REQ-AREA +26 1 A
* DEFINE GHI-RQA-ENTRY GHI-REQ-AREA +27 3 N VALUE 0
* DEFINE GHI-SET-CNT GHI-REQ-AREA +40 4 B VALUE 0
* DEFINE GHI-SET-NBR GHI-REQ-AREA +44 4 B VALUE 0
* DEFINE GHI-CNT-MAX GHI-REQ-AREA +50 2 B VALUE 0
* DEFINE GHI-IO-CNT GHI-REQ-AREA +52 2 B VALUE 0
* DEFINE GHI-SKP-CNT GHI-REQ-AREA +71 2 B VALUE 0
* DEFINE GHI-KEY-VAL GHI-REQ-AREA +76 180 A
* DEFINE GHI-WHO-AM-I GHI-REQ-AREA +256 28 A VALUE 'GHI'
```

```

%DBEL GHI PO LI VENDR TPVAL LATE
* DEFINE GHI-ELEM-LIST S          101 A
* DEFINE GHI-ELM01      GHI-ELEM-LIST      7 A VALUE 'PO '
* DEFINE GHI-ELM02      GHI-ELEM-LIST +6 7 A VALUE 'LI '
* DEFINE GHI-ELM03      GHI-ELEM-LIST +12 7 A VALUE 'VENDR '
* DEFINE GHI-ELM04      GHI-ELEM-LIST +18 7 A VALUE 'TPVAL '
* DEFINE GHI-ELM05      GHI-ELEM-LIST +24 7 A VALUE 'LATE '
* DEFINE GHI-ELM06      GHI-ELEM-LIST +30 7 A VALUE ' '
* DEFINE GHI-ELM07      GHI-ELEM-LIST +36 7 A VALUE ' '
* DEFINE GHI-ELM08      GHI-ELEM-LIST +42 7 A VALUE ' '
* DEFINE GHI-ELM09      GHI-ELEM-LIST +48 7 A VALUE ' '
* DEFINE GHI-ELM10      GHI-ELEM-LIST +54 7 A VALUE ' '
* DEFINE GHI-ELM11      GHI-ELEM-LIST +60 7 A VALUE ' '
* DEFINE GHI-ELM12      GHI-ELEM-LIST +66 7 A VALUE ' '
* DEFINE GHI-ELM13      GHI-ELEM-LIST +72 7 A VALUE ' '
* DEFINE GHI-ELM14      GHI-ELEM-LIST +78 7 A VALUE ' '
* DEFINE GHI-ELM15      GHI-ELEM-LIST +84 7 A VALUE ' '
* DEFINE GHI-ELM16      GHI-ELEM-LIST +90 7 A VALUE ' '
%DBWA GHI 20
* DEFINE GHI-WORK-AREA W          1 A OCCURS 20
  DEFINE GHI-PO      GHI-WORK-AREA      5 A
  DEFINE GHI-LI      GHI-WORK-AREA +5 3 A
  DEFINE GHI-VENDR   GHI-WORK-AREA +8 3 A
  DEFINE GHI-TPVAL   GHI-WORK-AREA +11 8 N MASK 'ZZZ,ZZZ9.99'
  DEFINE GHI-LATE    GHI-WORK-AREA +19 1 A
JOB NAME ALL-RECORDS INPUT NULL START STARTUP FINISH DBCLOSE
PRINT GHI-RPT          . * Print this record
%DBLREQ GHI REDNX      . * Read next record
* GHI-CMD = 'REDNX'
* CALL EZTPDTCM USING(USER-INFO      +
*                   GHI-REQ-AREA +
*                   GHI-WORK-AREA +
*                   GHI-ELEM-LIST)
  IF GHI-RET-CD = '14' . * Have we reached end of file?
    STOP                . * Yes; we're done
  END-IF
STARTUP.
  PROC                  . * Procedure for start-up processing
  PERFORM DBOPEN        . * Open the CA-Datcom/DB file
  GHI-KEY-VAL = X'0000000000' . * Set the first key value
  %DBSREQ GHI LOCKY     . * Locate key => start value
* GHI-CMD = 'LOCKY'
* CALL EZTPDTCM USING(USER-INFO      +
*                   GHI-REQ-AREA)
  IF GHI-RET-CD = '14' . * Did we not find one?
    STOP                . * Then we must not have any records
  END-IF
  %DBLREQ GHI REDLE     . * Read the located record
* GHI-CMD = 'REDLE'
* CALL EZTPDTCM USING(USER-INFO      +
*                   GHI-REQ-AREA +
*                   GHI-WORK-AREA +
*                   GHI-ELEM-LIST)
  END-PROC
%DBOPEN
* DBOPEN.
*   PROC
*   CALL EZTPDTCM USING(USER-INFO 'OPEN ')
*   END-PROC
%DBCLOSE
* DBCLOSE.
*   PROC
*   CALL EZTPDTCM USING(USER-INFO 'CLOSE')
*   END-PROC
REPORT GHI-RPT LINESIZE 80
  LINE GHI-PO GHI-VENDR GHI-TPVAL GHI-LATE

```

Purchase Orders by Line Item

This final program lists all purchase orders by line item, showing the part name on each line. The POH, POL, and PNM tables are accessed in this program.

```

1      %DBUIB EZT+DB MYURT
2      %DBRA JKL POH POLI
3      %DBEL JKL PO LI VENDR
4      %DBWA JKL 11
5          DEFINE JKL-PO    JKL-WORK-AREA    5    A
6          DEFINE JKL-LI    JKL-WORK-AREA +5  3    A
7          DEFINE JKL-VENDR JKL-WORK-AREA +8  3    A
8      %DBRA MNO POL POLI
9      %DBEL MNO PO LI PN
10     %DBWA MNO 11
11         DEFINE MNO-PO    MNO-WORK-AREA    5    A
12         DEFINE MNO-LI    MNO-WORK-AREA +5  3    A
13         DEFINE MNO-PN    MNO-WORK-AREA +8  3    A
14     %DBRA PQR PNM PN
15     %DBEL PQR PN DESC
16     %DBWA PQR 23
17         DEFINE PQR-PN    PQR-WORK-AREA +1  3    A
18         DEFINE PQR-DESC  PQR-WORK-AREA +3 20    A
19     *
20     JOB NAME TESTJOB4 INPUT NULL START STARTUP FINISH DBCLOSE
21     POL-KEY-VAL = POH-PO . * Set up POL key from POH record
22     %DBSREQ POL LOCKY . * Locate the key =>
23     IF POL-RET-CD = '14' . * Have we reached end of file?
24     GOTO NEXT-PO . * Yes; we don't have any line items
25     END-IF
26     %DBLREQ POL REDLE . * Read the located record
27     DO WHILE POL-PO = POH-PO
28         PNM-KEY-VAL = POL-PN . * Set up PNM key from POL record
29         %DBLREQ PNM REDKY . * Read the record
30         PRINT JKL-MNO-PQR-RPT . * Print the line from the 3 records
31         %DBLREQ POL REDNX . * Read next POL record
32     END-DO
33     NEXT-PO.
34     %DBLREQ POH REDNX . * Read the next POH record
35     IF POH-RET-CD = '14' . * Have we reached end of file?
36     STOP . * Yes; we're done with report
37     END-IF
38     STARTUP.
39     PROC . * Proc for startup processing
40     PERFORM DBOPEN . * Open the CA-Datcom/DB files
41     POH-KEY-VAL = X'0000000000' . * Set initial key value
42     %DBSREQ POH LOCKY . * Locate the key => start value
43     IF POH-RET-CD = '14' . * Have we reached end of file?
44     STOP . * Yes; we must not have any records
45     END-IF
46     %DBLREQ POH REDLE . * Read the located record
47     END-PROC
48     %DBOPEN
49     %DBCLOSE
50     *
50     REPORT JKL-MNO-PQR-RPT LINESIZE 80
51     LINE JKL-PO JKL-LI JKL-VENDR MNO-PN PQR-DESC

```

Line 1 defines the standard User Information Block (UIB) parameter area for CA-Datcom/DB calls. Only one of these is required in any program.

Lines 2-18 define the other three standard parameter areas: the Request Area (RA), the Element List (EL), and the Work Area (WA). These lines also define the particular field structure of each WA for each of the access paths you need into the database.

In lines 2-4, the RA, EL, and WA are matched together by the prefix JKL.

In lines 8-10, the RA, EL, and WA are matched together by the prefix MNO.

In lines 14-16, the RA, EL, and WA are matched together by the prefix PQR.

Line 20 is a JOB statement which specifies the initializing and terminating procedures you use. You must always code these procedures to point to the %DBOPEN or %DBCLOSE macros or to PROCedures that invoke them.

Lines 21-26 define your position in the POL file and read any line items that are to be printed.

Lines 27-32 contain a loop that reads the part name record for the current line item, prints the line, and reads the next line item. This process continues until the line items for the current purchase order are exhausted.

Lines 33-37 contain the logic to read the next purchase order and check for end of file.

Lines 38-47 contain the logic to perform the required initializing operations, such as invoking the DBOPEN procedure, setting the initial key value, and reading that record.

Line 48 contains the %DBOPEN macro that opens the required DATACOM tables.

Line 49 contains the %DBCLOSE macro that closes the required DATACOM tables.

Lines 51-52 are the typical CA-Easytrieve/Plus statements required to produce a report.

Index

%

%DBCLOSE, 3-11
%DBEL, 3-6
%DBFILE, 3-9
%DBLREQ, 3-12
%DBOPEN, 3-11
%DBRA, 3-5
%DBSREQ, 3-13
%DBUIB, 3-4
%DBWA, 3-8

A

Assembler routine, EZTPDTCM, 2-2
Automatic and controlled processing, 2-2, 3-1, 3-2
Automatic processing
 example, 3-14, 3-16
 EXIT parameter, 2-2
 EZTPDTCM routine, 2-2
 sequential retrieval, 2-2

C

CA-Datcom/DB access macros
 automatic & controlled processing DBEL, 3-1, 3-6
 automatic & controlled processing DBRA, 3-5
 automatic & controlled processing DBUIB, 3-1,
 3-4

 automatic processing DBFILE, 3-9
 controlled processing DBCLOSE, 3-2, 3-11
 controlled processing DBLREQ, 3-2, 3-12
 controlled processing DBOPEN, 3-2, 3-11
 controlled processing DBRA, 3-1
 controlled processing DBSREQ, 3-2, 3-13
 controlled processing DBWA, 3-1, 3-8
 DBCLOSE, 3-11
 DBEL, 3-6
 DBFILE, 3-9
 DBLREQ, 3-12
 DBOPEN, 3-11
 DBRA, 3-5
 DBSREQ, 3-13
 DBUIB, 3-4
 DBWA, 3-8

CA-Datcom/DB basic knowledge, 2-1

CA-Datcom/DB tables
 closing, 3-11
 opening, 3-11

CA-Easytrieve/Plus
 features, 2-1

CA-Easytrieve/Plus CA-Datcom/DB interface, 2-1
 database administrator tasks, 2-1
 preparing, 2-1
 using, 2-1

CA-Easytrieve/Plus CA-Datcom/DB Interface
 advantages, 2-1

CALL statement, 2-2
 controlled processing, 2-2

Calls, generating, 3-2

Close CA-Datcom/DB tables, 3-11

Close tables, 3-11

Command notation, 1-3

Communication between CA-Easytrieve/Plus and
CA-Datcom/DB, 2-2

Controlled and automatic processing, 2-2

Controlled processing, 3-1

CALL statement, 2-2

example, 3-17

retrieval methods, 2-2

D

Data definition statements, in CA-Easytrieve/Plus macro library, 2-2

Database access

automatic processing, 2-2

controlled processing, 2-2

Database administrator, tasks, 2-1

DBCLOSE, 3-11

DBEL, 3-6

DBOPEN, 3-11

DBRA, 3-5

DBSREQ, 3-13

DBUIB, 3-4

DBWA, 3-8

Define

element list, 3-6

request area, 3-5

user information block, 3-4

work area, 3-8

Direct request, 3-12, 3-13

Driver routine, EZTPDRVR, 2-2

E

EXIT parameter, of FILE statement, 2-2

EZTPDRVR, driver routine, 2-2

EZTPDTCM, assembler routine, 2-2

F

FILE statement, 2-2

generating, 3-9

G

Generate FILE statement, 3-9

L

Link editing driver (EZTPDRVR)

user requirements table, 2-2

Long parameter list, 3-12

O

Open CA-Datcom/DB tables, 3-11

Open tables, 3-11

P

Parameter areas, generating, 3-1

Programming errors, reducing, 2-2

Publications

CA-Easytrieve/Plus, 1-2

related, 1-2

S

Sequential retrieval, 2-2

Short parameter list, 3-13

U

User requirements table, 2-2