

CA-EZ/KEY™

Installation Guide

Release 3.2



Computer Associates®

1005EK32IGC

—PROPRIETARY AND CONFIDENTIAL INFORMATION—

THIS MATERIAL CONTAINS, AND IS PART OF A COMPUTER SOFTWARE PROGRAM WHICH IS, PROPRIETARY AND CONFIDENTIAL INFORMATION OWNED BY COMPUTER ASSOCIATES INTERNATIONAL, INC. THE PROGRAM, INCLUDING THIS MATERIAL, MAY NOT BE DUPLICATED, DISCLOSED, OR REPRODUCED IN WHOLE OR IN PART FOR ANY PURPOSE WITHOUT THE EXPRESS WRITTEN AUTHORIZATION OF COMPUTER ASSOCIATES. ALL AUTHORIZED REPRODUCTIONS MUST BE MARKED WITH THIS LEGEND.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the United States Government ("the Government") is subject to restrictions as set forth in A) subparagraph (c)(2) of the Commercial Computer Software — Restricted Rights clause at FAR 52.227-19, and/or B) subparagraphs (c)(1)(ii) of the Rights of Technical Data and Computer Software clause of DFAR 252.227-7013. This software is distributed to the Government by:

Computer Associates International, Inc.
One Computer Associates Plaza
Islandia, NY 11788-7000

Unpublished copyright work — all rights reserved under the copyright laws of the United States.

This material may be reproduced by or for the United States Government pursuant to the copyright license under the clause at DFAR 252.227-7013 (October 1988).

Release 3.2, November 1988

Copyright © 1988 Computer Associates International, Inc.
All Rights reserved.

CA-EZ/KEY is a trademark of Computer Associates International, Inc.

All product names referenced herein are trademarks of their respective companies.

Call Computer Associates technical services for any information not covered in this manual or the related publications. See your Computer Associates *Product Support Directory* for the appropriate telephone number to call for direct support, or you may call 1-800-645-3042 or 1-516-342-4683 and your call will be returned as soon as possible.

Outside North America, contact your local Computer Associates technical support center for assistance.

SECTION 1	GENERAL INFORMATION	
	Introduction	1-1
SECTION 2	CMS INSTALLATION	
	Introduction	2-1
	Reinstalling EZ/KEY under CMS	2-3
	Installation Files	2-5
	Installation Without DCSS	2-6
	Installation in a DCSS	2-27
	EASYTRIEVE PLUS Under EZ/KEY	2-52
	EASYTRIEVE PLUS VSAM Macro Libraries	2-53
	Installing EZ/KEY Under ISPF/PDF	2-56
	EZ/KEY SQL Support	2-57
	Installing CMS SQL Support	2-62
	Using SQL with EZ/KEY	2-68
	SQL/DS Userids	2-69
	Optional IDMS/IDD Installation	2-70
	Additional IDMS Considerations	2-74
	EZKSEgnn Messages	2-87
SECTION 3	TSO INSTALLATION	
	Introduction	3-1
	Reinstalling EZ/KEY under TSO	3-3
	Installation Files	3-4
	Installation JCL	3-7
	EZKEY CLIST	3-10
	EZTPLUS CLIST	3-15
	EZ/KEY Foreground Execution CLIST Considerations	3-18
	Choosing the Size of PIELIBs	3-21
	Optional TSO Installation Procedures	3-22
	Validate TSO Installation	3-27
	Using VSAM PIELIBs with EZ/KEY	3-28
	EASYTRIEVE PLUS VSAM Macro Libraries	3-29
	PANVALET Considerations for TSO	3-30
	TSO/TCAM Considerations	3-34
	TSO/VTAM Considerations	3-36
	Installing EZ/KEY Under ISPF/PDF	3-37
	Optional SQL Installation Procedures	3-38
	Optional IDMS/IDD Installation	3-44
	Additional IDMS Considerations	3-45
SECTION 4	CICS/OS/VS INSTALLATION	
	Introduction	4-1
	Reinstalling EZ/KEY under CICS/OS/VS	4-2
	Installation Files	4-4
	Installation JCL	4-9

	Choosing the Size of PIELIBs	4-14
	CICS/OS/VS Table Changes	4-15
	CICS Runaway Task Interval	4-19
	Installation in an MVS/XA Environment	4-21
	Validate CICS Installation	4-22
	EASYTRIEVE PLUS VSAM Macro Libraries for CICS/OS/VS	4-23
	PDS Macro Libraries for CICS/OS/VS	4-24
	CICS/OS/VS Submit to Background	4-25
	PANVALET Considerations for CICS/OS/VS	4-28
	Multiple EZ/KEY Systems Under One CICS	4-31
	Optional SQL Installation Procedures	4-33
	DB2 Userids	4-42
	Optional IDMS/IDD Installation	4-43
	Additional IDMS/IDD CICS Considerations	4-49
SECTION 5	CICS/DOS/VSE INSTALLATION	
	Introduction	5-1
	Reinstalling EZ/KEY under CICS/DOS/VSE	5-2
	Installation Files	5-4
	Installation JCL	5-8
	Choosing the Size of PIELIBs	5-14
	CICS/DOS/VSE Table Changes	5-15
	CICS Runaway Task Interval	5-20
	ICCF Considerations for CICS/DOS/VSE	5-22
	Validate CICS Installation	5-23
	EASYTRIEVE PLUS VSAM Macro Libraries for CICS/DOS/VSE	5-24
	PANVALET Considerations for CICS/DOS/VSE	5-25
	Multiple EZ/KEY Systems Under One CICS	5-28
	SQL Interface Option Installation	5-30
	SQL Interface Installation - VSE/SP 2.1	5-32
	SQL Interface Installation - DOS/VSE (pre-SP 2.1)	5-41
	Additional SQL CICS Considerations	5-51
	Optional IDMS/IDD Installation	5-54
	Additional IDMS Considerations	5-59
SECTION 6	MAINTENANCE	
	Introduction	6-1
	Customer Service Interface	6-2
	Maintenance Procedures	6-3

APPENDIX A INSTALLATION SUBMIT USER EXIT

Introduction	A-1
Summary	A-3
Parameter Lists	A-6
Calling Sequence	A-8
Installation of SUBMIT User Exit under CMS	A-9
Installation of SUBMIT User Exit under TSO	A-11
Installation of SUBMIT User Exit under CICS/OS/VS	A-12
Installation of SUBMIT User Exit under CICS/DOS/VSE	A-13
Link-Editing CICS/DOS/VSE Phases	A-14

APPENDIX B PANAUDIT PLUS CONSIDERATIONS

Introduction	B-1
Step 1: Change Members for Foreground and Back- ground Execution	B-2
Step 2: Place the Members in EXAMPLE PIELIB	B-10
Step 3: Allocate the PANAUDIT PLUS Macro Library	B-11
Step 4: Instruct Users to CONNECT and ENABLE the Macro Library	B-13

INDEX	I-1
--------------------	-----

Exhibit 2.1	Sample ISR@PRIM Panel	2-56
Exhibit 3.1	Sample ISR@PRIM Panel	3-37
Exhibit 4.1	Requesting a Newcopy	4-40
Exhibit 4.2	Results of Newcopy Request	4-41
Exhibit 4.3	Requesting a Newcopy	4-46
Exhibit 4.4	Results of Newcopy Request	4-47
Exhibit 5.1	Requesting a Newcopy	5-52
Exhibit 5.2	Results of Newcopy Request	5-53
Exhibit 5.3	Requesting a Newcopy	5-57
Exhibit 5.4	Result of Newcopy Request	5-58
Exhibit B.1	PAPLUSO JCL	B-3
Exhibit B.2	PAPLUSD JCL	B-4
Exhibit B.3	PAPLUS EXEC	B-6
Exhibit B.4	PAPLUS CLIST	B-9
Exhibit B.5	The External LIBSPACE Utility Panel	B-13

SECTION 1 GENERAL INFORMATION

Introduction	1-1
Operating Environment	1-1
EZ/KEY Installation	1-1
Documentation	1-2
Manuals	1-2
Using the Guide	1-2
Updating the Guide	1-4
Identifying Updated Material	1-4

(This page intentionally left blank.)

INTRODUCTION

The EZ/KEY Installation Guide provides technical information for installing and maintaining EZ/KEY. This Guide is intended for systems programmers who install, support, and maintain EZ/KEY.

Before installing EZ/KEY, you should read the sections appropriate to the environments in which you are installing EZ/KEY:

<u>Section</u>	<u>Environment</u>
2	VM/CMS
3	MVS/TSO
4	CICS/OS/VS
5	CICS/DOS/VSE

Operating Environment

EZ/KEY operates on IBM 370, 30xx, 43xx, 9370, or compatible processors that support VM/CMS, MVS/TSO, CICS/OS/VS, and CICS/DOS/VSE operating environments.

CICS Release 1.5 and above are supported.

DOS/VSE AF2 and above with POWER are supported.

MVS/SP Release 1 and above are supported.

VM/SP Release 1 and above are supported.

PANVALET Release 11.0 and above are supported for TSO and CICS.

EZ/KEY operates in conjunction with EASYTRIEVE PLUS 5.0 and later releases.

EZ/KEY executes in as little as one megabyte of virtual storage in CMS and TSO.

All EZ/KEY modules (phases) are reentrant.

EZ/KEY Installation

The installation procedure transfers EZ/KEY from the distribution tape to disk at your site. EZ/KEY is distributed on an IBM standard labeled tape.

Documentation

Manuals

EZ/KEY is documented in a set of four related manuals:

- *Reference Manual* – describes how to use EZ/KEY from an end-user perspective.
- *Administrator's Guide* – provides information, procedures, and examples for use in operating and maintaining EZ/KEY.
- *Installation Guide* – describes how to install EZ/KEY in your environment.
- *Tutorial* – provides the end-user with instruction in the initial use of EZ/KEY.

Using the Guide

Future Enhancements

Plus signs (+) in the left margin of this document identify future extensions to EZ/KEY that have been designed but not yet implemented.

PANSOPHIC welcomes comments on these proposed enhancements and other extensions and revisions that you feel would make EZ/KEY a more useful system. Use the prepaid Comments form at the end of this guide to document your suggestions.

Presentation Conventions

This guide uses the following conventions:

lower case italic text – variable syntax information that you must supply.

bold text – text or messages the computer displays to you.

bold italic text – text that you must input.

FIXED TEXT – code and screen examples. Accompanying descriptions refer to items within such examples using *lower case italic*, **bold**, or ***bold italic*** as appropriate.

Introduction

Enter – When this manual tells you to "enter" a command or series of characters, type the command or series of characters and press the ENTER key.

Page Numbering System

Pages in this Guide are numbered by sections as shown below:

2-1	means Section 2, page 1
4-11	means Section 4, page 11
B-3	means Appendix B, page 3

Updating the Guide

When future versions of EZ/KEY are released, replacement Guides or updates will be provided. Updates will follow the existing numbering system. The scheme for adding new pages is:

Existing Pages	New Pages	
5-1		
5-2		
	5-2a	
	← 5-2b	3-page insert
	5-2c	
5-3		
5-4		
5-5	← 5-5	replacement page
	← 5-6	new page

Identifying Updated Material

Enhancements and extensions to EZ/KEY in new versions are described in the Version Summary in the *EZ/KEY Reference Manual* and the *EZ/KEY Administrator's Guide*. When a new version is released, read through the Version Summary. Use the index to locate detailed information about new features. All changed pages contain the serial number of the Guide at the time of the revision. Vertical lines in the margin, like the ones on this paragraph, identify revised information.

A manual's serial number (printed on the bottom, outside corner of each page) identifies the current product version and each page's revision date. For example, a typical manual serial number is coded as follows:

<u>EKI</u>	<u>03.2</u>	-	<u>88</u>	<u>11</u>
guide identifier	version number		year of revision	month of revision

SECTION 2 CMS INSTALLATION

Introduction	2-1
Storage Estimates	2-1
Use of CMSL with EZ/KEY	2-2
Reinstalling EZ/KEY under CMS	2-3
Reinstallation Procedure	2-3
Installation Files	2-5
Installation Without DCSS	2-6
Installation Procedure	2-6
Minidisk Space Requirements	2-6
EZ/KEY CMS Files	2-7
EZKEY EXEC	2-9
Tailoring EZKEY EXEC	2-12
EZ/KEY Foreground Execution EXEC Considerations	2-13
EZTPLUSO EXEC	2-16
Tailoring EZTPLUSO EXEC	2-18
EZTPLUSD EXEC	2-20
Tailoring EZTPLUSD EXEC	2-24
PIEDIRST EXEC	2-25
Validate CMS Installation	2-26
Installation in a DCSS	2-27
Discontiguous Shared Segments	2-27
Virtual Storage	2-27
Real Storage	2-27
CMS Commands	2-28
Program Loading	2-28
Dynamic Program Loading	2-28
Foreground Execution	2-28
DCSS Installation Overview	2-29
DCSS Security Considerations	2-29
DCSS Installation Space Requirements	2-29
Load the DCSS Data From the Distribution Tape	2-30
Defining the DCSS	2-31
DMKSNTSE and DMKSNTLE ASSEMBLE	2-31
Name the Segment	2-33
Determine Origin Address	2-33
Allocate DASD Space	2-35
Specify Remaining Operands	2-36
Generate a New DMKSNT	2-37
Select the EZKEYSEG EXEC	2-37
EZKEYSEG EXEC	2-38

EZKEYSEG LARGEEXEC	2-41
EZKEYMAP EXEC	2-44
LOADMAP EZT	2-46
Alternate Segment Name	2-47
Alternate Segment Addresses	2-47
LOAD and SAVE the EZ/KEY Shared Segment	2-48
Validate DCSS Installation	2-50
DCSS Errors	2-51
EASYTRIEVE PLUS Under EZ/KEY	2-52
EASYTRIEVE PLUS VSAM Macro Libraries	2-53
Installing EZ/KEY Under ISPF/PDF	2-56
EZ/KEY SQL Support	2-57
SQL Interface Space Requirements	2-57
EZ/KEY SQL Interface Virtual Memory Requirements ..	2-57
SQL/DS Storage Requirements	2-57
SQL/DS User Virtual Machine Size	2-58
Install SQL/DS in a DCSS	2-58
Load the SQL Interface Data from the Distribution Tape	2-58
Installing the EZ/KEY SQL Interface	2-59
SQL/DS Help Tables	2-59
DQSMMTB ASMSQLDS	2-60
DQSMCMD ASMSQLDS	2-61
Installing CMS SQL Support	2-62
EZKEYSQL EXEC	2-62
Tailoring EZKEYSQL EXEC	2-66
Running EZKEYSQL EXEC	2-66
Additional SQL/DS Requirements	2-67
Using SQL with EZ/KEY	2-68
Execution Using EZ/KEY SQL Interface	2-68
SQL/DS Userids	2-69
Optional IDMS/IDD Installation	2-70
Installing EZ/KEY IDMS/IDD Support	2-70
EZKEYIDD Installation EXEC	2-71
Tailoring EZKEYIDD EXEC	2-73
Running EZKEYIDD EXEC	2-73
Additional IDMS Considerations	2-74

Central Version Mode Operation	2-74
IDMSOPTI Generation	2-74
Linking IDMSOPTI with EZKYIDMS	2-75
Accessing an MVS Central Version from CMS	2-76
IDMSMSVC Generation	2-76
Accessing a VSE Central Version from CMS	2-77
IDMS\$SVC Generation	2-77
Local Mode Operation	2-78
IDMS OS Foreground Execution Considerations	2-79
Central Version Mode OS Foreground Operation	2-79
Local Mode OS Foreground Operation	2-80
IDMS DOS Foreground Execution Considerations	2-81
Central Version Mode DOS Foreground Operation	2-81
Local Mode DOS Foreground Operation	2-81
IDMS OS Background Execution Considerations	2-82
Central Version Mode OS Background Operation	2-82
Local Mode OS Background Operation	2-82
IDMS DOS Background Execution Considerations	2-84
Central Version Mode DOS Background Operation	2-84
Local Mode DOS Background Operation	2-85
EZKSEGnn Messages	2-87

(This page intentionally left blank.)

INTRODUCTION

This section contains detailed procedures for installing EZ/KEY under CMS. You should also review the *EZ/KEY Administrator's Guide* to determine how you intend to operate EZ/KEY at your site.

Install EZ/KEY without a DCSS, and ensure that it operates correctly in your environment (see "Validate CMS Installation" later in this section). After validating the normal installation and operation of EZ/KEY, you can use the additional data in the second file to install EZ/KEY in a DCSS. The DCSS installation procedures assume that the non-DCSS installation has already been performed.

Storage Estimates

EZ/editor is an in-storage editor, similar to XEDIT or ISPF/PDF. The size of the program or data being edited is limited by the availability of virtual storage.

Without a DCSS, EZ/KEY requires a virtual machine size of at least one megabyte. With the small DCSS option, EZ/KEY can operate in a one megabyte virtual machine. With the large DCSS option, EZ/KEY can operate in the minimum 320K CMS virtual machine (512K is recommended).

These storage estimates are subject to user activities and installation options; because of these activities and options, you may need additional virtual storage to use EZ/KEY. The following factors can affect virtual storage availability:

- The number of CMS minidisks currently ACCESSed (CMS reads the entire directory into virtual storage)
- The number of files on each ACCESSed minidisk (each directory entry occupies 64 bytes)
- Presence of CMS nucleus extensions in user storage
- Presence of CMS MODULEs in user virtual storage

Memory requirements, disk space sizes, and EXECs presented in this section are for planning purposes only. More precise information is provided in the files on the EZ/KEY distribution tape.

Use of CMSL with EZ/KEY

If your users currently have IPL CMS specified in the VM directory, you may want to change this to IPL CMSL (the CMS Large system). CMSL places the CMS operating system code in a shared segment just below the 16 megabyte line in virtual storage. This frees up additional virtual storage in each user's virtual machine, making it available for use by EZ/KEY.

If you have the DIRMAINT program product, you can allow your EZ/KEY users to change their own directory entry to IPL CMSL by default by instructing them to enter:

DIRM IPL CMSL PARM AUTO CR

The ***PARM AUTO CR*** is optional. It instructs CMS to continue the IPL without waiting for the user to press ENTER.

Since IBM has relocated CMS to just below 16 megabytes, the need for CMSL has been eliminated. Therefore, CMSL may not be available at your site in VM/SP Release 5.

REINSTALLING EZ/KEY UNDER CMS

If you have already installed EZ/KEY under CMS, almost everything that you need is already in place. For Version 3.2 of EZ/KEY there are some important changes:

- The EXECs have been revised.
- EZ/KEY is now approximately 1.5 megabytes in total size. If you installed EZ/KEY in a DCSS and used a 896K segment size, you must change the DMKSNT entry for EZ/KEY to specify either a 256K segment (the small DCSS option) or a 1.5 megabyte segment (the high-performance, large DCSS option). See "Installation in a DCSS" for further information.

Reinstallation Procedure

Follow the procedure described here to reinstall EZ/KEY under CMS:

1. Review the output produced by the installation when the last version of EZ/KEY was installed. If you cannot locate this information, review the appropriate material in the previous version of the *EZ/KEY Installation Guide*.
2. Review the new sample EXECs for EZ/KEY and compare them with your current EXECs. Note the differences. Make the necessary changes to tailor the new EXECs for your environment. If you are converting from VSE SP 1.3 or lower to VSE SP 2.1 or above, re-evaluate the parameters for the EZTPLUSD EXEC.
3. Review the sections of this guide that pertain to the new features described above.
4. Once you have made any changes, install EZ/KEY as described under "Installation Without DCSS" in this manual.

Reinstalling EZ/KEY under CMS

NOTE: If you have installed a previous release of EZ/KEY in a DCSS and you wish to test this release from the MODULE rather than the DCSS, edit the EZ/KEY EXEC as follows:

Find the line that contains:

```
EXECOS EZKEYSEG &PARM
```

and change it to:

```
EXECOS EZKEYMOD &PARM
```

This change should be removed when you install this release into your DCSS.

5. Verify the results as described under "Verify CMS Installation" in this manual.
6. Load the DCSS materials onto a minidisk, as described in "Installation in a DCSS" in this manual.
7. Make any necessary changes to DMKSNT in your CP nucleus and allocate the necessary CP DASD space as described in "Installation in a DCSS" in this manual.
8. Review the new EZKEYSEG EXEC, the DCSS installation EXEC. Make any changes needed for your installation's environment.
9. Run the new EZKEYSEG EXEC to install EZ/KEY in the DCSS.
10. Verify the results as described under "Verify DCSS Installation" in this manual.
11. Review the new topics "EZ/KEY SQL Support" and "Optional IDMS/IDD Installation" for information regarding support of the SQL and IDD statements within EZ/KEY.

INSTALLATION FILES

The EZ/KEY system for CMS is distributed on a standard labeled tape that contains three files in CMS TAPE DUMP format:

- File 1 (DSN=FILE01) - Contains the data to install EZ/KEY as a normal program that operates in the CMS user area.
- File 2 (DSN=FILE02) - Contains additional data to install EZ/KEY in a Discontiguous Shared Segment (DCSS).
- File 3 (DSN=FILE03) - Contains additional data to install EZ/KEY SQL support for use with IBM's SQL/DS program product under VM/CMS.

INSTALLATION WITHOUT DCSS***Installation Procedure***

Have the VM operator:

1. ATTACH a 1600 BPI tape drive to your USERID as 181
2. MOUNT the EZ/KEY distribution tape
3. READY the tape drive

When the tape is ready, issue the following CMS commands to transfer the EZ/KEY system from the tape to a read/write CMS minidisk:

```
TAPE REW
TAPE FSF 1
TAPE LOAD * * fm
```

where *fm* is the filemode of your read/write minidisk. This minidisk must be formatted with a blocksize of at least 1024. For optimum performance, the blocksize of the minidisk should be 4096. EZ/KEY uses the extended I/O facilities of VM/CMS; thus it cannot be installed or executed on minidisks with 800-byte blocks.

Minidisk Space Requirements

The total space required for the EZ/KEY system is about 2,000 4096-byte blocks (approximately 15 cylinders of 3380 space). You can install EZ/KEY on a system minidisk or on a separate minidisk. These installation procedures assume a separate minidisk since this simplifies subsequent maintenance and reinstallations.

EZ/KEY CMS Files

Following the successful load of the EZ/KEY system, ten files exist:

Fname	Ftype	Description	CMS 4K Blocks Required
EZKEYMOD	MODULE	EZ/KEY Main module	50
EZKEYSEG	MODULE	EZ/KEY Segment Loader	1
SYSTEM	PIELIB	EZ/KEY System Library	1500
EXAMPLE	PIELIB	Example Library	200
EZKEY	EXEC	CMS EXEC to invoke EZ/KEY	1
PIEDIRST	EXEC	CMS EXEC to create a CMS EXEC containing a sorted list of members on a selected minidisk	1
EZREPORT	LISTING	EZ/REPORT User's Guide	10
EZTPLUSO	EXEC	CMS EXEC to execute OS EASYTRIEVE PLUS from a CMS TXTLIB or from an OS LOADLIB in the foreground	1
EZTPLUSD	EXEC	CMS EXEC to execute DOS EASYTRIEVE PLUS from a CMS DOSLIB or DOS Core Image Library in the foreground	1
EZKEYIDD	EXEC	CMS EXEC to install IDMS/IDD support for use with EZ/KEY	1
EZKEYIDD	TXTLIB	TXTLIB containing a dummy EZKYIDMS module	1
EZKYOPTN	TEXT	EZ/KEY Options mode	1

The SYSTEM PIELIB contains all of the panels and messages necessary for the correct operation of EZ/KEY.

The EXAMPLE PIELIB contains the examples described in the *EASYTRIEVE PLUS Application Guide* and the *EZ/KEY User's Guide* as well as sample JCL and EXECs.

After installing EZ/KEY, you may want to tailor some of the sample JCL members provided in the EXAMPLE PIELIB.

EZKEY EXEC

The EZKEY EXEC listed below invokes EZ/KEY under VM/CMS. The EZKEYSEG MODULE determines whether EZ/KEY has been installed in a DCSS; if so, it executes EZKEY in the DCSS. Otherwise, the EZKEYSEG MODULE loads and calls the EZKEYMOD MODULE in the CMS user area. You should review this EXEC and tailor it to your specific environment:

```

      &TRACE
* EZKEY EXEC:
*
*   LINK to and ACCESS the EZ/KEY residence minidisk.
*   Invoke EZ/KEY in a DCSS.  If no DCSS then call EZKEYMOD MODULE.
*
*   For foreground execution, execute EASYTRIEVE PLUS using
*   the PIETEMP EXEC, then cycle back and reinvoke EZKEY.
*
* Invoking parameters:
*
*   &1 = Initial Menu Selection value
*       (e.g., 3.3 = go directly to the copy/move panel)
*
*   &PARM = &1
*
* Installation tailorable symbolic definitions:
*
*   &EZKEYUID = USERID that owns the EZ/KEY minidisk
*   &EZKEYCUU = address of EZ/KEY minidisk in USERID's directory
*   &EZKEYVUU = virtual address to LINK the EZ/KEY minidisk to
*   &EZKEYMOD = CMS Filemode to ACCESS the EZ/KEY minidisk as
*   &SQLDSUID = USERID that owns the SQL/DS minidisk
*   &SQLDSCUU = address of SQL/DS minidisk in USERID's directory
*   &SQLDSVUU = virtual address to LINK the SQL/DS minidisk to
*   &SQLDSMOD = CMS Filemode to ACCESS the SQL/DS minidisk as
*   &SQLDBASE = the name of your installation's SQL/DS database
*   &EZKEYSQL = name of EZKEYSQL TXTLIB, or null for no SQL/DS
*   &EZKEYIDD = name of EZKEYIDD TXTLIB, or null for no IDMS/IDD

```

continued ...

... continued

```

*
* Modify the following based on the above definitions:
*
    &EZKEYUID = userid
    &EZKEYCUU = cuu
    &EZKEYVUU = vuu
    &EZKEYMOD = fm
    &SQLDSUID = userid
    &SQLDSCUU = cuu
    &SQLDSVUU = vuu
    &SQLDSMOD = fm
    &SQLDBASE = dbasename
    &EZKEYSQL =
    &EZKEYIDD =

*
-INIT
    Q CMSLEVEL ( STACK
    &READ VARS &A &B &REL
    &REL = &PIECE OF &REL 1 1
    SET DOS OFF
    &IF &REL >= 5 SET FULLSCREEN SUSPEND
    CP SET IMSG ON

*
* LINK and ACCESS the EZ/KEY minidisk:
*
    CP LINK &EZKEYUID &EZKEYCUU &EZKEYVUU RR
    ACCESS &EZKEYVUU &EZKEYMOD

*
* Set up EZ/KEY PRINTER and PUNCH output file definitions:
*
    FILEDEF EZKPRT PRINTER ( RECFM FBA LRECL 133 BLKSIZE 133
    FILEDEF EZKPUN PUNCH ( RECFM F LRECL 80

*
* Link and access SQL/DS residence disk:
*
    &IF .&EZKEYSQL = . &SKIP 3
    CP LINK &SQLDSUID &SQLDSCUU &SQLDSVUU RR
    ACCESS &SQLDSVUU &SQLDSMOD
    EXEC SQLINIT DBNAME(&SQLDBASE)

*
* Access any needed TXTLIBs:
*
    GLOBAL TXTLIB &EZKEYSQL &EZKEYIDD

```

continued ...

... continued

```
*
* Invoke EZ/KEY (in the shared segment if possible):
*
    SET CMSTYPE RT
    EXECOS EZKEYSEG &PARM
*
    &IF &RETCODE NE 4 &GOTO -OUT
*
* Foreground execution section:
*
    &IF &REL >= 5 SET FULLSCREEN RESUME
    EXEC PIETEMP
    SET CMSTYPE HT
    ERASE PIETEMP EXEC
    SET CMSTYPE RT
    &TYPE
    &TYPE Press ENTER to continue, or type END to exit EZ/KEY.
    &READ ARGS
    &IF &N>GT 0 &IF .&1 EQ .END &GOTO -OUT
    &PARM = 4
    &GOTO -INIT
-OUT
    &IF &REL >= 5 SET FULLSCREEN RESUME
*
* Reset TXTLIB, and release EZ/KEY minidisk and SQL/DS residence disk:
*
    GLOBAL TXTLIB
    RELEASE &EZKEYMOD ( DET
    &IF .&EZKEYSQL NE . RELEASE &SQLDSMOD ( DET
*
* Reset CMS environment and return to CMS:
*
    CP SET IMSG ON
    SET CMSTYPE RT
    &TYPE EZKEY018 - EZ/KEY completed
    &EXIT 0
```

Tailoring EZKEY EXEC

Tailor the EZKEY EXEC to your installation using the symbolic substitution parameters, as follows:

- &EZKEYUID - The userid of the virtual machine owning the minidisk where EZ/KEY is installed.
- &EZKEYCUU - The address of the minidisk in &EZKEYUID's directory where EZ/KEY is installed.
- &EZKEYVUU - The virtual address that you want the minidisk to be linked as in each EZ/KEY user's virtual machine.
- &EZKEYMOD - The CMS mode letter that you want the minidisk to be ACCESSEd as in each EZ/KEY user's virtual machine.
- &SQLDSUID - The userid of the virtual machine owning the minidisk where SQL/DS is installed.
- &SQLDSCUU - The address of the minidisk in &SQLDSUID's directory where SQL/DS is installed.
- &SQLDSVUU - The virtual address that you want the minidisk to be linked as in each EZ/KEY user's virtual machine.
- &SQLDSMOD - The CMS mode letter that you want the minidisk to be ACCESSEd as in each EZ/KEY user's virtual machine.
- &SQLDBASE - The name of the SQL/DS database at your installation. SQL/DS must be initialized for this user prior to use by EZ/KEY. This is accomplished by invoking the SQLINIT EXEC provided by IBM. If this is already specified in the user's PROFILE EXEC, you can comment out the LINK and ACCESS of the SQL residence minidisk, as well as the EXEC SQLINIT DBNAME (&SQLDBASE), and the corresponding RELEASE of the minidisk in the EZKEY EXEC. Consult the SQL/DS support person at your installation for the correct database name to use.

Installation Without DCSS

&EZKEYSQL - The name of the EZ/KEY SQL TXTLIB, if SQL is to be used with EZ/KEY; otherwise, null (leave blank). The default name for the TXTLIB provided with EZ/KEY is EZKEYSQL. See "EZ/KEY SQL Support" later in this section for more information.

&EZKEYIDD - The name of the EZ/KEY IDMS TXTLIB, if IDD is to be used with EZ/KEY; otherwise, null (leave blank). The default name for the TXTLIB provided with EZ/KEY is EZKEYIDD. See "Optional IDMS/IDD Installation" later in this section for more information.

The EZKEY EXEC must be placed on a common minidisk, such as the Y-disk, that is accessible to all CMS users who may be using EZ/KEY.

EZ/KEY Foreground Execution EXEC Considerations

The following discussion pertains to all of the EXECs that you may need to tailor for use with EZ/KEY Foreground Execution.

When tailoring any EXECs for use with EZ/KEY Foreground Execution, you can store these EXECs in one of several places for use by all of your EZ/KEY users.

One alternative is to place the tailored EXECs on the same minidisk where EZ/KEY resides. Since the EZ/KEY EXEC LINKs to and ACCESSes this minidisk whenever EZ/KEY is invoked, these EXECs are always available to your EZ/KEY users for use with foreground execution.

To implement this alternative, use the following steps to tailor the EXECs:

1. Select all of the EXECs in EXAMPLE by placing **S** next to **EXAMPLE**, and typing **EXEC** under the Member Type on the PROULSM panel.
2. Edit the EXEC you want to tailor by typing **ED** next to the member name.
3. After making your changes, press the END PF Key. This saves the member in your library.
4. Use utility option 3.3 to COPY the member(s) you modify from your library to the EXAMPLE Library. Specify **YES** for the Replace option.

The other choice is to place the tailored EXECs in a common library, such as EXAMPLE PIELIB. When a user specifies the name of the EXEC to use for foreground execution, EZ/KEY searches all enabled PIELIBs for an EXEC by that name.

To use this alternative, your EZ/KEY users must have EXAMPLE PIELIB enabled. This way, these EXECs are always available to your EZ/KEY users for use with foreground execution. You can accomplish this in one of two ways:

- Instruct your users to Enable EXAMPLE using utility option 3.2.

Or

- Use the Report Administrator Utility functions to have EZ/KEY automatically enable EXAMPLE for all or selected users. For detailed instructions on how to do this, see Section 7 of the *EZ/KEY Administrator's Guide*. Read the subsection that covers establishing default CONNECT and LIBCHAIN members.

To implement this alternative, use the following steps to tailor the EXECs:

1. Disable EXAMPLE PIELIB using utility option 3.2.
2. Select all of the EXECs in EXAMPLE by placing *S* next to EXAMPLE, and typing *EXEC* under the Member Type on the PROULSM panel.
3. Edit the EXEC you want to tailor by typing *ED* next to the member name.
4. After making your changes, press the END PF Key. This saves the member in your library.
5. Use utility option 3.a.3 to CONNECT EXAMPLE PIELIB with CREATE access rights. Fill in *EXAMPLE* for the LIBSPACE Name, *CREATE* for Access Rights, and press ENTER.
6. Use utility option 3.3 to COPY the member(s) you modified from your library to the EXAMPLE library. Specify *YES* for the Replace option.
7. Use utility option 3.2 to enable EXAMPLE if you want to have it normally enabled for your userid.

Installation Without DCSS

8. You can optionally use the Report Administrator functions of **EZ/REPORT** to enable **EXAMPLE** automatically for all or selected users.

You can use a library other than **EXAMPLE** PIELIB if you have one or more *common* PIELIBs for different groups of users in your environment. To do this, just substitute the name of your common library for **EXAMPLE** in the above instructions.

In this way, you can set up different versions of the Foreground Execution EXECs to satisfy varying requirements of different departments while maintaining standard naming conventions.

For example, you can have a different **EZTPLUS** EXEC for each department, and you can instruct all **EZ/KEY** users to specify the name **EZTPLUS** as the name to use for Foreground Execution. Each user uses the correct **EZTPLUS** EXEC for the department that they belong to.

EZTPLUSO EXEC

The EZTPLUSO EXEC is supplied as the default EXEC by which EZ/KEY invokes OS EASYTRIEVE PLUS from the Run On-Line foreground execution panel, PROEXECC (=4). Review this EXEC and tailor it to your environment. If your site primarily uses the OS version of EASYTRIEVE PLUS under CMS, you should rename this EXEC to EZTPLUS to be consistent with the documentation in the *EZ/KEY Administrator's Guide*.

```

      &TRACE
*
* EZTPLUSO EXEC:
*
*      LINK to and ACCESS the OS EASYTRIEVE PLUS residence minidisk.
*      Invoke OS EASYTRIEVE PLUS via the "EZTPAOO MODULE".
*
* Invoking Parameters:
*
*      &1 = Name of EASYTRIEVE PLUS program
*      &2 = DDname of file to be processed by EASYTRIEVE PLUS.
*      &3 =   CMS Filename of above file.
*      &4 =   CMS Filetype of above file.
*      &5 = "B" to Browse the SYSPRINT dataset;
*          "D" to route SYSPRINT to a Disk file;
*          "P" to Print SYSPRINT to your virtual printer;
*          "T" to Type SYSPRINT on your virtual console terminal.
*      &6 = DDname of 2nd file to be processed by EASYTRIEVE PLUS.
*      &7 =   CMS Filename of above file.
*      &8 =   CMS Filetype of above file.
*      &9 =   CMS Filetype of above program (usually "EZT").
*
* Installation tailorable symbolic definitions:
*
*      &EZTPVER = EASYTRIEVE PLUS version release number
*      &EZTPPGM = EASYTRIEVE PLUS MODULE name
*      &EZTPLIB = EASYTRIEVE PLUS TXTLIB name
*      &EZTPUID = USERID that owns the EASYTRIEVE PLUS minidisk
*      &EZTPCUU = address of minidisk in USERID's directory
*      &EZTPVUU = virtual address to LINK the minidisk to
*      &EZTPMOD = Filemode to ACCESS the minidisk as
*      &EZTPSRT = name of SORTLIB TXTLIB, or null if no SORT support
*      &EZTPSQL = name of EZTPSQL TXTLIB, or null for no SQL support

```

continued ...

... continued

```

*
* Modify the following based on the above definitions:
*   (You may elect to place this EXEC as well as all of
*   the EASYTRIEVE PLUS MODULES and TXTLIBS on the Y-DISK.)
*
    &EZTPVER = vrn
    &EZTPPGM = EZTP&EZTPVER
    &EZTPLIB = EZTP&EZTPVER
    &EZTPUID = userid
    &EZTPCUU = cuu
    &EZTPVUU = vuu
    &EZTPMOD = fm
    &EZTPSRT = SORTLIB
    &EZTPSQL =

*
* LINK and ACCESS the OS EASYTRIEVE PLUS residence minidisk
*
    SET CMSTYPE HT
    CP LINK &EZTPUID &EZTPCUU &EZTPVUU RR
    ACCESS &EZTPVUU &EZTPMOD

*
* LINK and ACCESS any sequential, VSAM, SQL/DS or other database disk(s)
*
    CP LINK userid mmm nnn RR
    ACCESS nnn m
    ...add more LINKs and ACCESSs here as needed...
*
* Set up FILEDEFs for all files needed by EASYTRIEVE PLUS program
*
    FILEDEF SYSPRINT CLEAR
    &IF .&5 NE .P &SKIP 1
        FILEDEF SYSPRINT PRINTER
    &IF .&5 NE .T &SKIP 1
        FILEDEF SYSPRINT TERMINAL
    &IF .&5 EQ .B &SKIP 1
    &IF .&5 NE .D &SKIP 1
        FILEDEF SYSPRINT DISK &1 LISTING A
    &IF .&2 = .2 &SKIP 1
        FILEDEF &2 DISK &3 &4 *
    &IF .&6 = .6 &SKIP 1
        FILEDEF &6 DISK &7 &8 *

```

continued ...

... continued

```
FILEDEF SYSOUT CLEAR
FILEDEF SYSOUT DISK  SYSOUT DATA A
FILEDEF SYSIN  CLEAR
FILEDEF SYSIN  DISK  &1 &9 *
*   ... add any additional FILEDEFs here as needed...
*
* Set up TXTLIBs, clear the screen and invoke OS EASYTRIEVE PLUS
*
GLOBAL TXTLIB &EZTPLIB &EZTPSQL &EZTPSRT
VMFCLEAR
SET CMSTYPE RT
EXECOS &EZTPPGM
SET CMSTYPE HT
CP SET IMSG OFF
*
* Erase the EASYTRIEVE PLUS program source
*
ERASE &1 &9
*
* Release any sequential, VSAM, SQL/DS or other database disk(s)
*
RELEASE nnn ( DET
*   ...add more RELEASEs here as needed..
*
* Release the OS EASYTRIEVE PLUS minidisk, reset TXTLIBs, and return
*
-EXIT
RELEASE &EZTPMOD ( DET
GLOBAL TXTLIB
&EXIT
```

Tailoring EZTPLUSO EXEC

Tailor the EZTPLUSO EXEC to your installation using the symbolic substitution parameters, as follows:

&EZTPVER - The version release number of EASYTRIEVE PLUS to use with EZ/KEY for foreground execution, for example, 52C.

Installation Without DCSS

- &EZTPPGM - The program module name of the main EASYTRIEVE PLUS module to use with EZ/KEY for foreground execution. This parameter is created automatically from &EZTPVER, so you don't have to change it.
- &EZTPLIB - The name of the EASYTRIEVE PLUS TXTLIB to use with EZ/KEY for foreground execution. This parameter is created automatically from &EZTPVER, so you don't have to change it.
- &EZTPUID - The userid of the virtual machine owning the minidisk where EZ/KEY is installed.
- &EZTPCUU - The address of the minidisk in &EZTPUID's directory where EZ/KEY is installed.
- &EZTPVUU - The virtual address that you want the minidisk to be linked as in each EZ/KEY user's virtual machine.
- &EZTPMOD - The CMS mode letter that you want the minidisk to be ACCESSEd as in each EZ/KEY user's virtual machine.
- &EZTPSRT - The name of the SORT TXTLIB, if SORTs are used in EASYTRIEVE PLUS programs using foreground execution with EZ/KEY; otherwise, null (leave blank). The default name provided is SORTLIB. Change this to the appropriate name for your installation.
- &EZTPSQL - The name of the EASYTRIEVE PLUS SQL TXTLIB, if SQL is to be used with foreground execution using EZ/KEY; otherwise, null (leave blank). The recommended name of the TXTLIB provided with EASYTRIEVE PLUS is ETSQLOS. See the *EASYTRIEVE PLUS Installation Guide* for more information.

EZTPLUSD EXEC

The EZTPLUSD EXEC is supplied as the default EXEC by which EZ/KEY invokes DOS EASYTRIEVE PLUS from the Run On-Line foreground execution panel, PROEXECC (=4). Review this EXEC and tailor it to your environment. If your site primarily uses the DOS version of EASYTRIEVE PLUS under CMS, you should rename this EXEC to EZTPLUS to be consistent with the documentation in the *EZ/KEY Administrator's Guide*.

```
&TRACE
*
* EZTPLUSD EXEC:
*
*   LINK to and ACCESS the CMS minidisk or DOS volume
*   where the EASYTRIEVE PLUS DOSLIB (or SYSCLB) is installed.
*   Invoke EASYTRIEVE PLUS via the CMS/DOS FETCH command.
*
* Invoking Parameters:
*
*   &1 = Name of EASYTRIEVE PLUS program
*   &2 = DLBLname of file to be processed by EASYTRIEVE PLUS.
*   &3 =   CMS FILENAME of above file
*   &4 =   CMS FILETYPE of above file.
*   &5 = "B" to Browse the SYSPRINT dataset;
*       "D" to route SYSPRINT to a Disk file;
*       "P" to Print SYSPRINT to your virtual printer;
*       "T" to Type SYSPRINT on your virtual console terminal.
*   &6 = DLBLname of 2nd file to be processed by EASYTRIEVE PLUS.
*   &7 =   CMS Filename of above file.
*   &8 =   CMS Filetype of above file.
*   &9 =   CMS Filetype of above program (usually 'EZT').
```

continued ...

... continued

```
*
* Installation tailorable symbolic definitions:
*
*   &EZTPUID   = USERID that owns the EASYTRIEVE PLUS minidisk
*   &EZTPCUU   = address of minidisk in USERID's directory
*   &EZTPVUU   = virtual address to LINK minidisk as
*   &EZTPMOD   = Filemode to ACCESS the minidisk as
*   &EZTPCLB   = DOSLIB or SYSCLB (must be DOSLIB for SQL/DS under CMS)
*   &EZTPLIB   = Filename of EASYTRIEVE PLUS DOSLIB
*   &EZTPSQL   = Filename of EZTPSQL DOSLIB or null if no SQL/DS support
*   &EZTP1     = level 1 of name of DOS Core Image Library (SYSCLB)
*   &EZTP2     = level 2 of name of DOS Core Image Library (SYSCLB)
*   &EZTP3     = level 3 of name of DOS Core Image Library (SYSCLB)
*   &EZTP4     = level 4 of name of DOS Core Image Library (SYSCLB)
*   &EZTP5     = level 5 of name of DOS Core Image Library (SYSCLB)
*
* Modify the following based on the above definitions:
*   (you may elect to place this EXEC on the 'Y' disk)
*
*
*   &EZTPUID   = userid
*   &EZTPCUU   = cuu
*   &EZTPVUU   = vuu
*   &EZTPMOD   = D
*   &EZTPCLB   = DOSLIB
*   &EZTPLIB   = EZTPDOS
*   &EZTPSRT   = SORTLIB
*   &EZTPSQL   =
*   &EZTP1     = dsn1
*   &EZTP2     = dsn2
*   &EZTP3     = dsn3
*   &EZTP4     = dsn4
*   &EZTP5     = dsn5
*
* LINK and ACCESS minidisk containing EASYTRIEVE PLUS DOSLIB, or
* LINK and ACCESS volume containing EASYTRIEVE PLUS Core Image Library.
*
*   CP LINK &EZTPUID &EZTPCUU &EZTPVUU RR
*   ACCESS &EZTPVUU &EZTPMOD
```

continued ...

... continued

```

*
* Access any sequential, VSAM, SQL/DS or other database disk(s) needed
*
* CP LINK userid mmm nnn RR
* ACCESS nnn m
* ...add additional LINKs and ACCESSes here as needed...
*
* Set up the CMS/DOS Environment
*
* SET DOS ON (VSAM)
*
* Issue all ASSGNs and DLBLs needed by EASYTRIEVE PLUS program
*
LISTFILE &1 &9 * ( STACK
&READ VARS &FN &FT &FM
ASSGN SYSIPT &FM
DLBL IUSYSIN &FM CMS &1 &9 ( SYSIPT PERM
ASSGN SYSLST A
DLBL IUSYSLA A CMS &1 LISTING ( SYSLST
&IF &2 = 2 &SKIP 4
LISTFILE &3 &4 * ( STACK
&READ VARS &FN &FT &FM
ASSGN SYSOO4 &FM
DLBL &2 &FM CMS &3 &4 ( SYSOO4
&IF &6 = 6 &SKIP 4
LISTFILE &7 &8 * ( STACK
&READ VARS &FN &FT &FM
ASSGN SYSOO5 &FM
DLBL &6 &FM CMS &7 &8 ( SYSOO5
*
* ASSGN SYSOO6 A
* DLBL FILEB A CMS FILEB DATA ( SYSOO6
* ...add additional ASSGNs and DLBLs here as needed...
*
* Clear the screen
*
VMFCLEAR
SET CMSTYPE RT
*
* Set up CMS DOSLIB or DOS Core Image Library (SYSCLB)
*
&IF &EZTPCLB = DOSLIB &SKIP 3
ASSGN SYSCLB D

```

continued ...

... continued

```

      DLBL IJVSCL D DSN &EZTP1 &EZTP2 &EZTP3 &EZTP4 &EZTP5 ( SYSCLB
      &SKIP 1
      GLOBAL DOSLIB &EZTPLIB &EZTPSQL &EZTPSRT
*
* Invoke EASYTRIEVE PLUS
*
      FETCH EZTPA00 (ORIGIN 20000 START
      SET DOS OFF
      SET CMSTYPE HT
      CP SET IMSG OFF
*
* Close SPOOL files, clean up
*
      RELEASE &EZTPVUU ( DET
      &IF &5 NE P &SKIP 3
      CP SPOOL PRINTER SYSTEM
      PRINT &1 LISTING *
      ERASE &1 LISTING
      &IF &5 NE T &SKIP 4
      SET CMSTYPE RT
      TYPE &1 LISTING *
      SET CMSTYPE HT
      ERASE &1 LISTING
*
* Erase the EASYTRIEVE PLUS program source
*
      ERASE &1 EZT
*
* Clear CMS DOSLIB
*
      &IF &EZTPCLB = SYSCLB &SKIP 1
      GLOBAL DOSLIB
*
* Release any sequential, VSAM, SQL/DS or other database disk(s)
*
      RELEASE nnn ( DET
      ...add additional RELEASEs here as needed...
*
* Release EASYTRIEVE PLUS residence disk, and return to CMS
*
      RELEASE &EZTPVUU ( DET
      SET CMSTYPE RT
      &EXIT

```

Tailoring EZTPLUSD EXEC

Tailor the EZTPLUSD EXEC to your installation using the symbolic substitution parameters, as follows:

- &EZTPUID - The userid of the virtual machine owning the minidisk where EZ/KEY is installed.
- &EZTPCUU - The address of the minidisk in &EZTPUID's directory where EZ/KEY is installed.
- &EZTPVUU - The virtual address that you want the minidisk to be linked as in each EZ/KEY user's virtual machine.
- &EZTPMOD - The CMS mode letter that you want the minidisk to be ACCESSed as in each EZ/KEY user's virtual machine.
- &EZTPCLB - Specifies whether EASYTRIEVE PLUS is to be executed from a CMS DOSLIB or a DOS/VSE Core Image Library (SYSCLB). Specify DOSLIB for CMS DOSLIBs, or SYSCLB for Core Image Library. You must install EASYTRIEVE PLUS in a CMS DOSLIB to use SQL/DS in your EASYTRIEVE PLUS programs under VM/CMS. You must also do this if your VSE system is VSE/SP 2.1 or above.
- &EZTPLIB - The name of the EASYTRIEVE PLUS DOSLIB to use with EZ/KEY for foreground execution.
- &EZTPSRT - The name of the SORT TXTLIB, if SORTs are used in EASYTRIEVE PLUS programs using foreground execution with EZ/KEY; otherwise, null (leave blank). The default name provided is SORTLIB. Change this to the appropriate name for your installation.
- &EZTPSQL - The name of the EASYTRIEVE PLUS SQL DOSLIB, if SQL is to be used with foreground execution using EZ/KEY; otherwise, null (leave blank). The recommended name of the DOSLIB provided with EASYTRIEVE PLUS is ETSQLDOS. See the *EASYTRIEVE PLUS Installation Guide* for more information.

The following parameters apply only when &EZTPCLB = SYSCLB is specified, indicating that EASYTRIEVE PLUS is to be executed from a DOS/VSE Core Image Library. These parameters identify the DOS resident Core Image Library dataset name:

- &EZTP1 - The first level qualifier of the dataset name of the core image library to be used with foreground execution under EZ/KEY; otherwise, null (leave blank).
- &EZTP2 - The second level qualifier of the dataset name of the core image library to be used with foreground execution under EZ/KEY; otherwise, null (leave blank).
- &EZTP3 - The third level qualifier of the dataset name of the core image library to be used with foreground execution under EZ/KEY; otherwise, null (leave blank).
- &EZTP4 - The fourth level qualifier of the dataset name of the core image library to be used with foreground execution under EZ/KEY; otherwise, null (leave blank).
- &EZTP5 - The fifth level qualifier of the dataset name of the core image library to be used with foreground execution under EZ/KEY; otherwise, null (leave blank).

PIEDIRST EXEC

The PIEDIRST EXEC sorts a directory list of CMS files. It is used by the Library Utility (option 3.2) to generate a member selection list of the files on a CMS minidisk:

```

      &TRACE
*
* PIEDIRST EXEC:
*
*   Creates a sorted directory list of files on a CMS disk
*
      ERASE CMS EXEC A1
      LISTFILE &1 &2 &3 ( LABEL EXEC &4
      &STACK FIFO SORT * 8 20
      &STACK FIFO FILE
      SET CMSTYPE HT
      XEDIT CMS EXEC A1
      SET CMSTYPE RT
      &EXIT

```

Validate CMS Installation

To verify the correct installation of EZ/KEY, type **EZKEY** and press ENTER. You should see the message **EZKEY001 - Creating your Library file**. A panel should be displayed with the name PROSTART in the upper-left corner, as illustrated below:

```
PROSTART ----- Primary Selection Menu ----- EZ/KEY
COMMAND ==>                                     USERID: userid
                                                TIME  : 10:31:51
                                                DATE   : 05/12/88

Select Option ==>

      0  Program Function Key Settings
      1  Edit an EASYTRIEVE PLUS Program
      2  Edit any Member
      3  Library and System Utilities
      4  Run a Program On-Line
      5  Submit a Program for Batch Processing
      6  Report Processing Facility

      T  Tutorial for EZ/KEY

      X  Exit the EZ/KEY System

EZ/KEY, Version X.X
Copyright (c) 1983, 1988 Pansophic Systems, Inc.
```

After validating the EZ/KEY installation, review the *EZ/KEY Administrator's Guide*, especially the subsection that covers establishing CMS user PIELIBs and libraries in Section 4. If you want to set up user PIELIBs, you should do so at this time.

INSTALLATION IN A DCSS***Discontiguous Shared Segments***

All EZ/KEY modules are reentrant, so you may optionally choose to install the base or all of EZ/KEY in a Discontiguous Shared Segment (DCSS). Discontiguous Shared Segments are areas of VM/SP Control Program owned Direct Access Storage that contain, in page format, the contents of part of the virtual address space as it was at some point in time.

Installing EZ/KEY in a DCSS improves the system in the following ways:

- Reduces each CMS EZ/KEY user's virtual storage requirements
- Reduces real storage usage by sharing one copy of reentrant code
- Allows execution of other CMS commands from EZ/KEY
- Streamlines program loading of EZ/KEY in CMS

Virtual Storage

Since Discontiguous Shared Segments are always above the high end of a user's virtual storage, the EZ/KEY code does not have to be loaded into the user's virtual storage. Thus, the virtual machine size required to use EZ/KEY can be significantly smaller. With all of EZ/KEY installed in a DCSS, it is possible to run EZ/KEY in a 512K CMS virtual machine. In a one megabyte virtual machine, EZ/KEY users can edit members much larger than possible without a DCSS.

Real Storage

When a page within a Discontiguous Shared Segment is referenced, all users reference the same physical page. This reduces the demand for real storage and increases the probability that the page is already resident when referenced. Overall system performance is improved because system page space usage and paging of code is reduced when more than one person is using EZ/KEY.

CMS Commands

Without a DCSS, EZ/KEY runs in the user area; attempting to execute a CMS command that runs in the same area causes EZ/KEY to display the message **INVALID SUBSET COMMAND**. When a program running in the CMS user area invokes another program, only transient area commands are allowed; this is called CMS SUBSET mode. By installing EZ/KEY in a DCSS, you permit your users to execute any CMS commands or EXECs, without concern for whether they are CMS SUBSET commands.

Program Loading

Significant reductions in program load overhead may be achieved by placing EZ/KEY in a Discontiguous Shared Segment. When EZ/KEY is installed in a DCSS, the EZ/KEY program code segment is mapped into the virtual address space by CP and then branched to, eliminating the need to locate and read the EZKEYMOD MODULE from a CMS mini-disk. Each user's performance is improved since MODULE loading is eliminated.

Dynamic Program Loading

Without a DCSS or with the small DCSS, EZ/KEY loads program modules when they are needed and deletes them when they are not. This process is called dynamic program loading. With the large DCSS option, all of the modules are resident in the DCSS, so EZ/KEY does not have to dynamically load any routines. This can significantly reduce the number of I/O operations and CPU cycles needed to operate EZ/KEY.

Foreground Execution

You can obtain significant reductions in foreground execution overhead by installing EZ/KEY in a DCSS. When EZ/KEY is executing from within a DCSS, it can invoke EASYTRIEVE PLUS directly to perform the foreground execution function without requiring large amounts of additional virtual storage. This eliminates having to exit and re-enter EZ/KEY each time an EASYTRIEVE PLUS program is run online.

DCSS Installation Overview

The following tasks must be done to gain the benefits of a DCSS:

- Plan the layout of users' address spaces with discontinuous shared segments attached.
- Plan the layout of DASD space on CP-owned volumes to contain the shared segments. Allow for growth when reserving blocks of space.
- Prepare the DASD to contain the shared segments by reserving space in the CP directory. Use the CP Format/Allocate utility to format the DASD and allocate it as PERM, if necessary.
- Define the shared segments to CP using the NAMESYS macro in DMKSNT.
- Generate a new CP nucleus that includes the new definitions.
- Execute the EZKEYSEG EXEC to save the EZ/KEY code into the DCSS.

DCSS Security Considerations

When planning to install EZ/KEY in a Discontinuous Shared Segment, keep in mind that the SAVESYS command used to write information into the DCSS is a privileged CP command (class E). You must have a privilege class of E (or higher) to complete the installation of EZ/KEY in a DCSS.

DCSS Installation Space Requirements

The EZ/KEY DCSS installation process requires a 3,500 block minidisk whose minimum blocksize is 1,024. This can be a temporary minidisk since the files are not needed after completing the DCSS installation. The TEXT decks, EZKEY TXTLIB, and CMS EXECs for generating the shared segment are placed on this minidisk during the DCSS generation. After installing the DCSS, you can erase these files to free up this disk space. However, you may wish to keep the modified DMKSNT ASSEMBLE and the tailored EZKEYSEG EXEC on a permanent minidisk for future use and reference.

Load the DCSS Data From the Distribution Tape

Have the VM operator:

1. ATTACH a 1600 bpi tape drive to your USERID as 181
2. MOUNT the EZ/KEY installation tape
3. READY the tape drive

Issue the following CMS commands to transfer the EZ/KEY DCSS data from the tape to the EZ/KEY DCSS residence minidisk:

```
TAPE REW
TAPE FSF 4
TAPE LOAD * * fm
```

where *fm* is the filemode of the EZ/KEY DCSS residence minidisk.

Following a successful load of the EZ/KEY DCSS materials, the following files should exist on the EZ/KEY DCSS minidisk:

File Name	File Type	Description
EZKEY	TXTLIB	CMS text library for LOADING the EZ/KEY DCSS
*****	TEXT	Approximately 250 object decks for LOADING the DCSS
EZKEYMAP	EXEC	EXEC to edit the LOAD MAP
EZKEYSEG	EXEC	EXEC to LOAD and SAVESYS the small EZ/KEY DCSS
EZKEYSEG	LARGEEXEC	EXEC to LOAD and SAVESYS the large EZ/KEY DCSS
EZKEYLNK	EXEC	EXEC to recreate EZKEYMOD MODULE from object decks
DMKSNTSE	ASSEMBLE	Model NAMESYS macro defining the small EZ/KEY DCSS
DMKSNTLE	ASSEMBLE	Model NAMESYS macro defining the large EZ/KEY DCSS
LOADMAP	EZT	EASYTRIEVE PLUS program to edit the LOAD MAP

Defining the DCSS

Definitions of Shared Segments (DCSS) are contained in CP in the System Name Table – DMKSNT. You define a DCSS to CP by coding NAMESYS macros in your DMKSNT ASSEMBLE file that define these named objects to CP.

DMKSNTSE and DMKSNTLE ASSEMBLE

Modify DMKSNT to include EZKEY as a *named system*; this must be done by a VM systems programmer (refer to the *IBM VM/SP System Programmer's Guide* (SC19-6203)). The following DMKSNTSE and DMKSNTLE ASSEMBLE files are included on the second file of the distribution tape. You should tailor one of these files to your specific environment and include it in your DMKSNT ASSEMBLE file. Which file you tailor depends on whether you want to install the small or large DCSS option:

```

*****
*
*   EZKEY           CONTAINS PANSOPHIC'S EZ/KEY BASE   (SMALL)
*                   ALL SEGMENTS ARE SHARED
*   5568K           570000           1392           87-90
*
*****
*
*   THE SPACE FOR EZ/KEY IS ALLOCATED ON VM3380 AS FOLLOWS:
*   CYL 172, PAGE 001 TO CYL 172, PAGE 065 (065 PAGES)
*                   TOTAL = 65 PAGES
*
*****
SPACE
EZKEY  NAMESYS SYSNAME=EZKEY,
        SYSPGM=(1392-1457),
        SYSHRSG=(87-90),
        SYSVOL=VM3380,
        SYSSTRT=(017,1),
        SYSPGCT=65,
        SYSSIZE=256K,
        PROTECT=ON,
        SYSCYL=,
        VSYRES=,
        VSYADR=IGNORE
C
C
C
C
C
C
C
C
C
C

```

```

*****
*
*   EZKEY           CONTAINS PANSOPHIC'S EZ/KEY BASE (LARGE)
*   ALL SEGMENTS ARE SHARED
*   5568K           570000           1392           87-110
*
*****
*
*   THE SPACE FOR EZ/KEY IS ALLOCATED ON VM3380 AS FOLLOWS:
*   CYL 172, PAGE 001 TO CYL 172, PAGE 150 (150 PAGES)
*   CYL 173, PAGE 001 TO CYL 173, PAGE 150 (150 PAGES)
*   CYL 174, PAGE 001 TO CYL 174, PAGE 085 (085 PAGES)
*   TOTAL = 385 PAGES
*
*****
*
*   SPACE
EZKEY  NAMESYS SYSNAME=EZKEY,
        SYSPGM=(1392-1775),
        SYSHRSG=(87-110),
        SYSVOL=VM3380,
        SYSSTRT=(0172,1),
        SYSPGCT=384,
        SYSSIZE=1536K,
        PROTECT=ON,
        SYSCYL=,
        VSYSRES=,
        VSYSADR=IGNORE

```

Planning is required in locating the DCSS within the user's 16-megabyte address space. Systems that may be used concurrently must be assigned nonoverlapping addresses, while preserving as much as possible of the total addressing range of 16 megabytes for large applications.

EZ/KEY executes either from a DCSS or from a MODULE and determines from which to execute at runtime. Other software such as VSE/VSAM executes only from a DCSS.

Examine the parameters of the NAMESYS macro used within DMKSNT to define each Saved System or DCSS.

Name the Segment***SYSNAME***

The **SYSNAME** parameter identifies the shared segment to CP. An example of this parameter is:

SYSNAME=EZKEY

You may choose any name for the DCSS, but **EZKEY** is recommended. The following procedures allow for multiple segments so that you can provide alternate segments at several different virtual addresses (alternate segments can also simplify maintenance and testing).

EZ/KEY requires a minimum of 256K of contiguous virtual storage for the small shared segment, and 1536K for the large shared segment option.

Determine Origin Address

Determine what part of the virtual machine's address space is to be occupied by the DCSS. Determine which systems will be used concurrently. This is usually accomplished by determining the load addresses for the various systems and the virtual storage they will occupy.

For example, the suggested load address for the **EZKEY** shared segment is **X'570000'**, and the segment extends through **X'5B0000'** for the small DCSS, and through **X'6F0000'** for the large DCSS option. These addresses are conveyed to CP through the **SYSPGNM** parameter. Shared segments are identified with the **SYSHRSG** parameter.

SYSPGNM

SYSPGNM specifies the pages, or range of pages, to be saved and restored in the virtual address space. Its format is:

SYSPGNM=(nn,nn-nn,...)

where *nn* is a page number, in decimal, to be saved and restored, and *nn-nn* is an inclusive range of page numbers to be saved and restored. To calculate a page number from a load address, divide by 4K (**X'1000'**) and convert to decimal.

For the EZKEY example:

<u>SMALL DCSS OPTION:</u>	<u>LARGE DCSS OPTION:</u>
X'570000'	X'570000'
----- = X'570' = 1392	----- = X'570' = 1392
X'1000'	X'1000'
X'5B0000'	X'6F0000'
----- = X'5B0' = 1456	----- = X'6F0' = 1776
X'1000'	X'1000'

so code:

SYSPGM=(1392-1455)

SYSPGM=(1392-1775)

SYSHRSG

SYSHRSG identifies individual 64K segments of the saved system or DCSS that can be shared. Its format is:

SYSSHRSG=(nn,nn,...)

where *nn* is a segment number, in decimal, of a saved segment that is to be shared.

To calculate the segment number from the load address, divide by 64K (X'10000') and convert to decimal. In the example for EZKEY:

<u>SMALL DCSS OPTION:</u>	<u>LARGE DCSS OPTION:</u>
X'570000'	X'570000'
----- = X'57' = 87	----- = X'57' = 87
X'10000'	X'10000'
X'5B0000'	X'6F0000'
----- = X'5B' = 91	----- = X'6F' = 111
X'10000'	X'10000'

so code:

SYSHRSG=(87-90)

SYSHRSG=(87-110)

Allocate DASD Space

Identify the DASD space to contain the EZ/KEY DCSS. Be careful that DASD space intended for the storage of one shared segment does not overlap another shared segment's allocation.

Make sure that the volume to contain the EZ/KEY DCSS is CP-owned (i.e., listed in the SYSOWN macro of DMKSYS). Reserve the allocated space with a dummy minidisk. Run the CP Format/Allocate program to format the cylinders to be used and allocate them as PERM.

Three operands must be specified: SYSVOL, SYSSTRT, and SYSPGCT.

SYSVOL

SYSVOL identifies the volume to contain the image of the saved segments. It must be a CP-owned volume (i.e., have been specified in the SYSOWN macro of DMKSYS). For example:

```
SYSVOL=VMSPR3
```

SYSSTRT

SYSSTRT locates the area that contains the DCSS on the volume identified by SYSVOL. For count-key-data format devices (e.g., 3350, 3380), the format is:

```
SYSSTRT=(ccc,ppp)
```

where *ccc* indicates the starting cylinder number and *ppp* the page number within the cylinder. For example:

```
SYSSTRT=(11,1)
```

For FBA devices (3310 or 3370) the format is:

```
SYSSTRT=pppppp
```

where *pppppp* identifies the first page (NOT block) allocated to this DCSS. For example:

```
SYSSTRT=10520
```

SYSPGCT

SYSPGCT specifies the total number of pages to be saved or restored for this saved system or DCSS. Its format is:

SYSPGCT=nn

where *nn* is the number of pages to be saved or restored to the virtual machine's address space. This should equal the number of pages identified by the SYSPGNM operand above, and if omitted, will be calculated by the NAMESYS macro during assembly of DMKSNT.

This count does NOT include the extra page used by CP to save other information about the virtual machine such as storage keys. Be sure to account for this extra page in allocating DASD space for the shared segment. No checks are performed during the assembly of DMKSNT to insure against allocation of insufficient DASD space or overlapping extents. Be careful that DASD space intended for the storage of one shared segment does not overlap another shared segment's allocation.

SYSSIZE

The SYSSIZE parameter specifies the size of the shared segment to CP. An example of this parameter is:

<u>SMALL DCSS OPTION:</u>	<u>LARGE DCSS OPTION:</u>
SYSSIZE=256K	SYSPGNM=1536K

PROTECT

The EZ/KEY shared segment should be protected from alteration, so code **PROTECT=ON**, or allow this operand to default.

Specify Remaining Operands

The last set of operands specifies the location of the virtual system residence device. These operands are: SYSCYL or SYSBLOK, VSYSRES, and VSYSADR.

Since EZ/KEY is not an IPLable system, these operands must be specified as follows:

SYSCYL=,
VSYSRES=,
VSYSADR=IGNORE

Generate a New DMKSNT

Insert the new NAMESYS macro contained in DMKSNTSE or DMKSNTLE ASSEMBLE into your DMKSNT ASSEMBLE file, assemble it to generate a new TEXT deck, and build a new CP nucleus that contains the new definition.

Select the EZKEYSEG EXEC

To install just the base module of EZ/KEY in a DCSS (the small DCSS option), use the EZKEYSEG EXEC. It is shown on the following pages.

To install all of EZ/KEY in a DCSS (the large DCSS option), rename EZKEYSEG EXEC to EZKEYSEG SMALL, and EZKEYSEG LARGEEXEC to EZKEYSEG EXEC. The EZKEYSEG LARGEEXEC is shown immediately after the EZKEYSEG EXEC. Follow the instructions that are included with the EZKEYSEG LARGEEXEC to tailor it for your environment.

EZKEYSEG EXEC

Use the EZKEYSEG EXEC if you want to install only the base EZ/KEY module in a DCSS. This exec is located in the second file of the distribution tape. When using this exec, you must tailor it to your specific environment:

```

      &TRACE
*
* EZKEYSEG EXEC:
*
*   LOAD and SAVE a small DCSS named "EZKEY" for EZ/KEY.
*
*   This EXEC must be run by a VM USERID with:
*   - VM System Programmer privilege class E and
*   - at least 7 megabytes of virtual storage.
*   If &EZKEYORG is increased, then more than 7 Megabytes of
*   virtual storage may be required to run this EXEC.
*
* Invoking parameters: (optional)
*
*   &1 = override name of the shared segment (genned in DMKSNT)
*   &2 = override Origin of the EZKEY segment in virtual storage
*
* System default values symbolic definitions: (do not change)
*
*   &EZKEYTXT = CMS Filename of the EZKEY TXTLIB.
*   &EZKEYTXT = EZKEY
*
* Installation tailorable symbolic definitions: (installation defaults)
*
*   &EZKEYID  = USERID owning minidisk containing EZKEY DCSS data
*   &EZKEYCUU = address of DCSS minidisk in USERID's directory
*   &EZKEYVUU = virtual address to LINK DCSS minidisk to
*   &EZKEYMOD = CMS Filemode to ACCESS the DCSS minidisk as
*   &EZKEYORG = default Origin of EZKEY segment in virtual storage
*   &EZKEYSEG = default name of shared segment (genned in DMKSNT)

```

continued ...

... continued

```

*
* Specify installation default values based on the above definitions:
*
    &EZKEYID   = userid
    &EZKEYCUU  = cuu
    &EZKEYVUU  = 499
    &EZKEYMOD  = H
    &EZKEYORG  = 570000
    &EZKEYSEG  = EZKEY
*
* Apply any parameter overrides to the installation's default values:
*
    &IF &N LT 2 &SKIP 1
        &EZKEYORG = &2
    &IF &N LT 1 &SKIP 1
        &EZKEYSEG = &1
*
* LINK and ACCESS the minidisk containing the EZKEY TEXT decks:
*
    CP LINK &EZKEYID &EZKEYCUU &EZKEYVUU RR
    ACCESS &EZKEYVUU &EZKEYMOD
*
* Set up the GLOBAL TXTLIB for LOADING the modules:
*
    GLOBAL TXTLIB &EZKEYTXT
    &IF &RETCODE = 0 &SKIP 2
        &TYPE GLOBAL TXTLIB error: &EZKEYTXT TXTLIB not found
        &EXIT 99
*
* LOAD the CSECTs into virtual storage in this order for better paging
*
    &TYPE Loading Modules ...
    SET CMSTYPE HT
    LOAD PIEDMAIN ( CLEAR NOAUTO NOLIBE ORIGIN &EZKEYORG
*
* Module list of CMS INCLUDE commands (do not change):
*
    INCLUDE PIEEZKEY ( NOAUTO NOLIBE )
    INCLUDE PIEDLFMT ( SAME )
    INCLUDE PIEDLLOD ( SAME )
    .
    .
    .

```

continued ...

... continued

```
INCLUDE PIEVPMGR ( SAME )
INCLUDE PQSMATCH ( SAME )
* INCLUDE PIEXITO1 ( SAME )
SET CMSTYPE RT
INCLUDE DQSZKOOO ( AUTO LIBE MAP RESET PASCALVS )
*
* Generate the module:
*
    CP SAVESYS &EZKEYSEG
*
* Edit the LOAD MAP:
*
    &TYPE
    &TYPE      Issue "CP DEFINE STORAGE 2M", and then "IPL CMS".
    &TYPE      Then, run the EZKEYMAP EXEC to edit the load map
    &TYPE      just created.  Invoke it as follows:
    &TYPE
    &TYPE      EZKEYMAP &EZKEYSEG
    &TYPE
*
* Release the EZ/KEY minidisk and exit:
*
    RELEASE &EZKEYMOD ( DET
    &EXIT 0
```

EZKEYSEG LARGEEXEC

Use the EZKEYSEG LARGEEXEC if you want to install all of EZ/KEY in a DCSS (the large DCSS option). It is located on the second file of the distribution tape. When using this exec, you must tailor it to your specific environment:

```

      &TRACE
*
* EZKEYSEG LARGEEXEC:
*
*   LOAD and SAVE the large DCSS named "EZKEY" for EZ/KEY.
*
*   This .EXEC must be run by a VM USERID with:
*     - VM System Programmer privilege class E and
*     - at least 8 megabytes of virtual storage.
*   If &EZKEYORG is increased, then more than 8 megabytes of
*     virtual storage may be required to run this EXEC.
*
* Invoking parameters: (optional)
*
*   &1 = override name of the shared segment (genned in DMKSNT)
*   &2 = override Origin of the EZKEY segment in virtual storage
*
* System default values symbolic definitions: (do not change)
*
*   &EZKEYTXT = CMS Filename of the EZKEY TXTLIB
*   &EZKEYTXT = EZKEY
*
* Installation tailorable symbolic definitions: (installation defaults)
*
*   &EZKEYID   = USERID owning minidisk containing EZKEY DCSS data
*   &EZKEYCUU  = address of DCSS minidisk in USERID's directory
*   &EZKEYVUU  = virtual address to LINK DCSS minidisk to
*   &EZKEYMOD  = CMS Filemode to ACCESS the DCSS minidisk as
*   &EZKEYORG  = default Origin of EZKEY segment in virtual storage
*   &EZKEYSEG  = default name of shared segment (genned in DMKSNT)
*
* Specify installation default values based on the above definitions:
*
*   &EZKEYID   = userid
*   &EZKEYCUU  = cuu
*   &EZKEYVUU  = 499
*   &EZKEYMOD  = H
*   &EZKEYORG  = 57000C
*   &EZKEYSEG  = EZKEY

```

continued ...

... continued

```

*
* Apply any parameter overrides to the installation's default values:
*
    &IF &N LT 2 &SKIP 1
        &EZKEYORG = &2
    &IF &N LT 1 &SKIP 1
        &EZKEYSEG = &1
*
* LINK and ACCESS the minidisk containing the EZKEY TEXT decks:
*
    CP LINK &EZKEYID &EZKEYCUU &EZKEYVUU WR
    ACCESS &EZKEYVUU &EZKEYMOD
*
* Rename PIESSFIX for the large DCSS option:
*
    RENAME PIESSFIX TEXT &EZKEYMOD = SMALTEXT =
    RENAME PIESSFIX LARGTEXT &EZKEYMOD = TEXT =
*
* Increase the default size of the CMS loader tables:
*
    SET LDRTBLS 8
*
* Set up the GLOBAL TXTLIB for LOADING the modules:
*
    GLOBAL TXTLIB &EZKEYTXT
    &IF &RETCODE = 0 &SKIP 2
        &TYPE GLOBAL TXTLIB error: &EZKEYTXT TXTLIB not found
        &EXIT 99
*
* LOAD the CSECTs into virtual storage in this order for better paging:
*
    &TYPE Loading Modules ...
    SET CMSTYPE HT
    LOAD PIEDMAIN ( CLEAR NOAUTO NOLIBE ORIGIN &EZKEYORG
*
* Module list of CMS INCLUDE commands: (do not change)
*
    INCLUDE PIEEZKEY ( NOAUTO NOLIBE )
    INCLUDE PIEDLFMT ( SAME )
    INCLUDE PIEDLLOD ( SAME )
    INCLUDE PIEDMINT ( SAME )
    .
    .
    .

```

continued ...

... continued

```
INCLUDE PQDMNOPT ( SAME )
INCLUDE PQDMNXYZ ( SAME )
INCLUDE PQSSSWP ( SAME )
SET CMSTYPE RT
INCLUDE PQSMATCH ( AUTO LIBE MAP RESET PASCALVS )
*
* Save EZ/KEY in the DCSS:
*
    CP SAVESYS &EZKEYSEG
*
* Rename PIESSEFIX for the small DCSS option:
*
    RENAME PIESSEFIX TEXT &EZKEYMOD = LARGTEXT =
    RENAME PIESSEFIX SMALTEXT &EZKEYMOD = TEXT =
*
* Edit the LOAD MAP:
*
    &TYPE
    &TYPE      Issue "CP DEFINE STORAGE 2M", and then "IPL CMS".
    &TYPE      Then, run the EZKEYMAP EXEC to edit the load map
    &TYPE      just created.  Invoke it as follows:
    &TYPE
    &TYPE      EZKEYMAP &EZKEYSEG
    &TYPE
*
* Release the EZ/KEY minidisk and exit:
*
    RELEASE &EZKEYMOD ( DET
    &EXIT 0
```

EZKEYMAP EXEC

The EZKEYMAP EXEC is used to edit the LOAD MAP produced by the EZKEYSEG EXEC. This EXEC invokes the LOADMAP EZT program, shown in the next section, using the EZTPLUS EXEC:

```

      &TRACE
*
* EZKEYMAP EXEC:
*
*   Edit the LOAD MAP after running EZKEYSEG or EZKEYLNK EXEC.
*
*   This EXEC must be run by a VM USERID with a normal
*   virtual storage size, if using the DOS version of
*   EASYTRIEVE PLUS under CMS/DOS.
*
*
* Invoking Parameters: (optional)
*
*   &1 = override name of the segment or module
*
* Installation Tailorable Symbolic Definitions: (installation defaults)
*
*   &EZKEYID = USERID owning minidisk containing EZ/KEY
*   &EZKEYCUU = address of minidisk in USERID's directory
*   &EZKEYVUU = virtual address to LINK minidisk to
*   &EZKEYMOD = CMS Filemode to ACCESS the minidisk as
*   &EZKEYPAS = CP write password to access minidisk (if needed)
*   &EZKEYSEG = default name of shared segment (genned in DMKSNT)
*
* Specify Installation default values based on the above definitions:
*
*   &EZKEYID = userid
*   &EZKEYCUU = cuu
*   &EZKEYVUU = 499
*   &EZKEYMOD = H
*   &EZKEYPAS = PASSWORD
*   &EZKEY   = EZKEY
*
* Apply any Parameter Overrides to the Installation's default values:
*
*   &IF &N LT 1 &SKIP 1
*       &EZKEY = &1

```

continued ...

... continued

```
*
* LINK and ACCESS the minidisk containing EZ/KEY
*
  CP LINK &EZKEYID &EZKEYCUU &EZKEYVUU WR &EZKEYPAS
  ACCESS &EZKEYVUU &EZKEYMOD
*
* Edit the LOAD MAP
*
  ERASE &EZKEY MAP A
  EXEC EZTPLUS LOADMAP OLDMAP LOAD MAP P NEWMAP &EZKEY MAP EZT
  &IF &RC EQ 0 COPY &EZKEY MAP A = = &EZKEYMOD
  &IF &RC EQ 0 ERASE LOAD MAP A
*
* Print the LOAD MAP
*
  PRINT &EZKEY MAP A
  &IF &RC EQ 0 ERASE &EZKEY MAP A
*
* Release the minidisk and exit
*
  RELEASE &EZKEYMOD ( DET
  &EXIT 0
```

LOADMAP EZT

The LOADMAP EZT program edits the LOAD MAP resulting from loading EZ/KEY into a DCSS. Due to the way CSECTs are loaded in a particular order for improved paging, the LOAD MAP generated by CMS contains many error messages which are later resolved. The following EASYTRIEVE PLUS program removes the unnecessary messages, leaving only the meaningful module map information. This program is called by the EZKEYSEG EXEC, which uses the EZTPLUS EXEC to invoke it:

```
FILE OLDMAP F(100)
  INREC 1 80 A
  NAME 2 8 A
  AREA 10 4 A
  ADDR 14 6 A

FILE NEWMAP F(80)
- OUTREC 1 80 A

JOB INPUT(OLDMAP) NAME(EDIT-LOAD-MAP)
  IF INREC SPACES
    GOTO JOB
  END-IF
  IF AREA = ' SD ' , ' '
    IF ADDR SPACES
      GOTO JOB
    ELSE
      OUTREC = INREC
      PUT NEWMAP
    END-IF
  END-IF
```

Alternate Segment Name

To use a different name for the EZKEY shared segment, edit the CMS file EZKEYSEG TEXT. Near the end of the text deck are the following character constants:

```
EZKEY  EZKEY2  EZKEY3  EZKEY4  EZKEY5  EZKEY6  EZKEY7  EZKEYMOD
```

Use the CHANGE command to change the first "EZKEY " to the desired name. FILE the TEXT deck and issue the following CMS commands to regenerate the EZKEYSEG MODULE:

```
LOAD  EZKEYSEG ( ORIGIN TRANS
GENMOD EZKEYSEG MODULE h ( STR ALL
```

where *h* is the filemode of the minidisk for the EZKEYSEG MODULE.

Alternate Segment Addresses

If you wish to install multiple segments at different virtual addresses, name the segments EZKEY2, EZKEY3, etc., as shown above.

EZKEYSEG is the shared segment loader routine for EZ/KEY. It provides for up to seven different segments to be attached at different virtual addresses, depending on the size of the user's virtual machine. EZKEYSEG attempts to connect each segment in turn, from left to right. Therefore, these segment names must correspond with segment origins from the lowest to highest virtual address. This ensures that EZKEYSEG always attaches a segment at the lowest possible address just above the user's virtual address space.

The last name in the list is used to find the EZKEYMOD MODULE when no shared segment can be attached or none are available. This name must not be the name of any VM named system or shared segment name you have installed in your VM system. If you rename EZKEYMOD MODULE, you must change EZKEYMOD in the TEXT to match, and regenerate the EZKEYSEG MODULE, as described above under "Alternate Segment Name."

To install multiple segments with different names from those provided, change the names, as explained above under "Alternate Segment Name," to the names you wish to use. These names must be specified and assembled in DMKSNT and generated into your CP nucleus.

LOAD and SAVE the EZ/KEY Shared Segment

Issue the following CMS EXEC to LOAD and SAVESYS EZ/KEY into the EZKEY named shared segment:

```
EZKEYSEG
```

At this point, the following messages are displayed on your console:

```
DEV nnn DOES NOT EXIST
h (nnn) R/O
Loading Modules ...
DUPLICATE IDENTIFIER 'AMPZTGET'
DUPLICATE IDENTIFIER 'AMPZTPUT'
SYSTEM SAVED
```

```
Issue "CP DEFINE STORAGE 2M", and then "IPL CMS".
Then, run the EZKEYMAP EXEC to edit the load map
just created. Invoke it as follows:
```

```
EZKEYMAP EZKEY
```

```
DASD nnn DETACHED
R;
```

You have just generated the EZ/KEY DCSS. From this moment forward, all users issuing the EZKEY EXEC execute from the shared segment.

If you generate additional segments, choose a different origin address, or choose a different segment name. Use the following syntax:

```
EZKEYSEG segname origin
```

Installation in a DCSS

For example, you install two segments, EZKEY and EZKEY2, at virtual addresses 600000 and 800000, respectively. You should issue the following CMS EXECs to LOAD and SAVESYS EZ/KEY into each EZKEY named shared segment:

```
EZKEYSEG EZKEY 600000
```

The above messages are displayed on your console.

```
EZKEYSEG EZKEY2 800000
```

Again, the above messages are displayed on your console.

Validate DCSS Installation

To verify the correct installation of EZ/KEY in a DCSS, type EZKEY and press ENTER. A panel should be displayed, with the name PROSTART in the upper left-corner, as illustrated below:

```
PROSTART ----- Primary Selection Menu ----- EZ/KEY
COMMAND ==>                                     USERID: userid
                                                TIME  : 10:31:51
                                                DATE   : 05/12/88

Select Option ==>

          0  Program Function Key Settings
          1  Edit an EASYTRIEVE PLUS Program
          2  Edit any Member
          3  Library and System Utilities
          4  Run a Program On-Line
          5  Submit a Program for Batch Processing
          6  Report Processing Facility

          T  Tutorial for EZ/KEY

          X  Exit the EZ/KEY System

EZ/KEY, Version X.X
Copyright (c) 1983, 1988 Pansophic Systems, Inc.
```

Press PA1 to enter CP. The screen should clear, and you should see **CP READ** displayed in the lower right corner of the screen.

Enter **D PSW** and press ENTER. Examine the address in the PSW. It should be somewhere within the range specified for your DCSS. If it is, EZ/KEY is now executing in a DCSS.

Enter a **B** (for begin) and press ENTER. The PROSTART panel should re-appear after a **MORE...** or **HOLDING** message.

Press the END PF key to exit EZ/KEY.

If it appears that you are not running in a DCSS, do the following:

1. Enter a **B** (for begin) and press ENTER. The PROSTART panel should re-appear after a **MORE...** or **HOLDING** message.
2. Press PF3 (END) to exit EZ/KEY.
3. Send yourself a message, such as **MSG * HELLO**. This will cause the screen to go into HOLDING status before displaying PROSTART.
4. Run through this procedure again, watching for any messages that are displayed just before the PROSTART panel appears. These messages indicate why EZ/KEY is unable to run in a DCSS. They are documented under "EZKSEGnn MESSAGES" at the end of this section.
5. Use the message explanations to figure out what is happening.

DCSS Errors

Errors encountered when saving the EZ/KEY DCSS may be caused by improperly formatted DASD. They are corrected by rerunning the CP Format/Allocate program.

Errors encountered during the use of the EZ/KEY DCSS may be caused by:

- A mismatch between the SYSPGNM specification and the load address used when saving the system
- Insufficient DASD space for the DCSS
- A system which was not saved before you attempted to use it

The program EZKEYSEG module, the EZ/KEY shared segment loader, detects many of these common errors and issues appropriate messages to assist the installer in problem determination. See "EZKSEGnn Messages" at the end of this section for a list of these messages and their explanations.

EASYTRIEVE PLUS UNDER EZ/KEY

In order to use the DIRECT execution option of EZ/KEY, you may need to install EZ/KEY in a Discontiguous Shared Segment (DCSS).

See Section 3 of the *EZ/KEY Administrator's Guide*, specifically the subsection on tuning parameters and foreground execution, for further details regarding DIRECT execution.

If you are running the DOS version of EASYTRIEVE PLUS under CMS, you must install EZ/KEY in a DCSS in order to use DIRECT execution mode.

If you are running the OS version of EASYTRIEVE PLUS under CMS, you must install EZ/KEY in a DCSS, or generate a new version of the EASYTRIEVE PLUS main program, EZTPnnn MODULE (where *nnn* is the version of EASYTRIEVE PLUS, e.g., 50C) for use with EZ/KEY. To generate a new EZTPnnn MODULE, perform the following steps:

1. LINK to and ACCESS the minidisk containing EASYTRIEVE PLUS. Be sure to LINK to the minidisk with write capability.
2. Generate a new EZTPnnn MODULE by entering the following CMS commands:

GLOBAL TXTLIB EZTPnnn

LOAD EZTPA00 (ORIGIN TRANS

GENMOD EZTPnnn MODULE *m* (FROM A000OS STR

where *m* is the filemode used in step 1 to ACCESS the EASYTRIEVE PLUS residence minidisk.

*EASYTRIEVE PLUS VSAM Macro Libraries***EASYTRIEVE PLUS VSAM MACRO LIBRARIES**

To define EASYTRIEVE PLUS Macro libraries to EZ/KEY under CMS, the following statements must be added to the EZKEY EXEC, just after the label -INIT:

```
*
* Set up the CMS DOS environment:
*
    SET DOS ON (VSAM
    SET DOSPART 1024K
*
* Reset the CMS VSAM environment:
*
    EXECOS
*
* Link to and access all needed VSE packs:
*
    CP LINK VSEIPD 431 431 RR
    ACC 431 D
    CP LINK VSEIPD 432 432 RR
    ACC 432 P
    CP LINK VSEIPD 433 433 RR
    ACC 433 Q
*
* Define the VSAM Master Catalog:
*
    ASSGN SYSCAT D
    DLBL IJSYSCT CLEAR
    DLBL IJSYSCT D DSN VSE1 VSAM MASTER CATALOG ( VSAM SYSCAT PERM
*
* Define the default VSAM User Catalog:
*
    ASSGN SYSO09 P
    DLBL IJSYSUC CLEAR
    DLBL IJSYSUC P DSN USERCAT PERSONEL ( VSAM SYSO09 PERM
*
* Define any additional VSAM User Catalogs:
*
    ASSGN SYSO10 Q
    DLBL ACCNTNG CLEAR
    DLBL ACCNTNG Q DSN USERCAT ACCNTNG ( VSAM SYSO10 PERM
*
```

continued ...

EASYTRIEVE PLUS VSAM Macro Libraries

... continued

```

* Define the VSAM datasets to be accessed:
*
  ASSGN SYSO11 P
  DLBL PANDD1 CLEAR
  DLBL PANDD1 P DSN PERSONEL EZTP MACLIB ( VSAM SYSO11 PERM
*
  ASSGN SYSO12 Q
  DLBL PANDD2 CLEAR
  DLBL PANDD2 Q DSN ACCNTNG EZTP MACLIB ( VSAM SYSO12 CAT ACCNTNG PERM
*

```

The EASYTRIEVE PLUS Macro libraries must be CONNECTed and ENABLEd within EZ/KEY. In this example, you would use PANDD1 and PANDD2 as the External Name to CONNECT the EASYTRIEVE PLUS Macro libraries to EZ/KEY. Type EZTMAC for the LIBSPACE type, and the access mode is READ.

Also add the following statements to the EZ/KEY EXEC, immediately after the label -OUT:

```

-OUT
*
* Clean up the CMS DOS environment:
*
  DLBL PANDD1 CLEAR
  DLBL PANDD2 CLEAR
  DLBL IJSYSUC CLEAR
  DLBL IJSYSCT CLEAR
*
* Release and detach any VSE packs:
*
  RELEASE 431 ( DET
  RELEASE 432 ( DET
  RELEASE 433 ( DET

```

As explained in the *EASYTRIEVE PLUS Installation Guide*, to use EZTMAC as the EASYTRIEVE PLUS macro library, specify MACRO=VSAM as an operand of the EZTPOPT options macro.

This facility requires CMS/VSAM support to be generated in your VM system. If EZ/KEY is installed in a DCSS, the address of the DCSS must not conflict with the CMSVSAM, CMSDOS, and CMSAMS segments.

EASYTRIEVE PLUS VSAM Macro Libraries

The System Administrator must use the External LIBSPACE Utility (option 3.a.4) to CONNECT each EASYTRIEVE PLUS Macro library to EZ/KEY. For further information, see Section 4 of the *EZ/KEY Administrator's Guide*.

INSTALLING EZ/KEY UNDER ISPF/PDF

You can install EZ/KEY under ISPF to appear on the ISPF Primary Option Menu, ISR@PRIM. To do this, make the following changes:

1. Modify the ISR@PRIM panel to display the EZ/KEY choice.
2. In the PROC section of the ISPF panel, invoke the EZ/KEY EXEC.

The following example shows EZ/KEY installed as option E:

```
%----- ISPF/PDF PRIMARY OPTION MENU -----
%OPTION ==> _ZCMD
%
%      1 +BROWSE          ...
%      2 +EDIT           ...
%      3 +UTILITIES      ...
%
%
%      E +EZ/KEY          Productivity Aid for EASYTRIEVE PLUS
%
%
%
%      )INIT
%      ...
%      )PROC
%          &ZSEL = TRANS( TRUNC(&ZCMD, '.')
%                      1, 'PGM(ISRBRO)'
%                      2, 'PGM(ISREDIT)'
%                      3, 'PANEL(ISRUTIL)'
%                      .
%                      .
%                      E, 'CMD(EZKEY) NEWPOOL'
%                      .
%                      .
%
%      )END
```

Exhibit 2.1: Sample ISR@PRIM Panel

Due to the combined storage requirements of ISPF and EZ/KEY, increase the size of your region by at least 1.5 megabytes.

EZ/KEY SQL SUPPORT

File 3 on the EZ/KEY Installation Tape for CMS contains the PANSOPHIC SQL Interface modules needed to access SQL/DS from EZ/KEY under CMS. The SQL Interface support for EZ/KEY can be executed under CMS using OS simulation from a TXTLIB, provided you have SQL/DS running under VM.

SQL Interface Space Requirements

The EZ/KEY SQL installation process requires a 2,000 block minidisk whose minimum blocksize is 1,024. This can be a temporary minidisk since the files are not needed after completing the SQL installation. The TEXT decks, EZKEYSQL TXTLIB, and CMS EXECs for generating the shared segment are placed on this minidisk during the SQL Installation. After installing the SQL Interface, you can erase these files to free up this disk space. However, you may wish to save the modified DQSMCMD and DQSMMTB ASSEMBLE and the tailored EZKEYSQL EXEC on a permanent minidisk for future use and reference.

EZ/KEY SQL Interface Virtual Memory Requirements

Approximately 250K of additional memory is used when the first SQL statement is encountered, in order to load all of the SQL Interface modules into virtual storage. Each additional SQL statement requires approximately 100 bytes of virtual storage for work areas.

Due to the additional storage required by the SQL Interface, you may need to increase the virtual machine size by at least 512K for CMS users who use EZ/KEY.

SQL/DS Storage Requirements

The EZ/KEY SQL Interface under CMS invokes SQL/DS routines to perform syntax checking of SQL statements embedded in your EASYTRIEVE PLUS programs. Therefore, additional virtual storage is required by SQL/DS for processing.

SQL/DS User Virtual Machine Size

To run SQL/DS under CMS, you must increase the size of each user's virtual machine by approximately 1 megabyte. For example, suppose your EZ/KEY users currently have 1 megabyte virtual machines and you want to increase them to 2 megabytes. If you have the DIRMAINT program product, you can allow your EZ/KEY users to change their own directory entry to specify their virtual machine size by instructing them to enter:

DIRM STOR 2M

For further information on virtual storage requirements for SQL/DS, refer to the IBM publication *SQL/Data System Planning and Administration for VM/SP* (SH24-5043).

Install SQL/DS in a DCSS

To save additional virtual storage in each user's virtual machine, you can install SQL/DS in a DCSS. The space normally required to load all of the SQL/DS program modules into virtual storage in each user's virtual machine is eliminated because the DCSS copy of the program code is shared between all users using SQL/DS under CMS. Note that if you install EZ/KEY in a DCSS, the EZ/KEY DCSS must not overlap any part of the SQL/DS DCSS. For further information on installing SQL/DS in a DCSS, refer to the IBM publication *SQL/Data System Installation for VM/SP* (SH24-5044).

Load the SQL Interface Data from the Distribution Tape

Have the VM operator:

1. ATTACH a 1600 bpi tape drive to your USERID as 181
2. MOUNT the EZ/KEY installation tape
3. READY the tape drive

Issue the following CMS commands to transfer the EZ/KEY SQL data from the tape to the EZ/KEY SQL residence minidisk:

```
TAPE REW  
TAPE FSF 7  
TAPE LOAD * * fm
```

where *fm* is the file-mode of the EZ/KEY SQL residence minidisk.

Following a successful load of the EZ/KEY SQL materials, the following files should exist on the EZ/KEY SQL minidisk:

<u>File Name</u>	<u>File Type</u>	<u>Description</u>
DQSMCMD	ASMSQLDS	EZKEY SQL Command Processor Source
DQSMMTB	ASMSQLDS	EZKEY SQL Error Message Table Builder Source
EZKEYSQL	EXEC	CMS EXEC to install EZ/KEY SQL Interface code
EZKEYSQL	TXTLIB	CMS text library containing SQL Interface code

Installing the EZ/KEY SQL Interface

The EZKEYSQL EXEC is provided in File 3 of the EZ/KEY Installation Tape. To install PANSOPHIC's SQL Interface in a TXTLIB for use with EZ/KEY, review the instructions in this section, tailor the EZKEYSQL EXEC, and then run the EZKEYSQL EXEC, as described next under "Installing CMS SQL Support."

SQL/DS Help Tables

During installation of the SQL Interface, you must identify the name of the SQL/DS Help Tables and the UserID and password for the SQLDBA UserID to Pansophic's SQL Interface. This change must be made in both the SQL Interface Message Table Builder program, DQSMMTB ASMSQLDS, and the SQL Interface Command Processor program, DQSMCMD ASMSQLDS, as illustrated below.

Use an editor (such as XEDIT or ISPF) to modify DQSMMTB ASMSQLDS and DQSMCMD ASMSQLDS. Specify the correct name of the SQL/DS Help Tables and the UserID and password for the SQLDBA UserID.

To the right of lines that include SQLDBA is the string `***-x-***`, where *x* is a letter that corresponds to one of the notes listed below. These notes describe the changes you must make in order to use the programs. These note identifiers (`***-x-***`) appear only in this documentation, not in the actual files.

- a. The DQSMMTB program contains the name of the SQL/DS Help Table, SQLDBA.SYSTEXT2. If your site changed the name of this table during installation of SQL/DS, change the name specified in the program, where indicated. Search for *SQLDBA.SYSTEXT2*.
- b. The DQSMMTB program contains the UserID and password of the SQLDBA UserID. Search for *SQLDBA*. Change the UserID and password for the SQLDBA UserID to the correct UserID and password for use at your site.
- c. The DQSMCMDB program contains the name of the SQL/DS Help Tables, SQLDBA.SYSTEXT1 and SQLDBA.SYSTEXT2. If your site changed the name of these tables during installation of SQL/DS, change the names specified in the program, where indicated. Search for *SQLDBA.SYSTEXT*.

DQSMMTB ASMSQLDS

```

*****
*
* THIS PROGRAM READS THE IBM SQL/DS HELP TEXT TABLES, EXTRACTS
.
.
MESSAGE DS      OH
        EXEC SQL
                DECLARE C1 CURSOR FOR
                        SELECT ITEM, "SQL/DS HELP"
                        FROM  SQLDBA.SYSTEXT2
                        WHERE ITEM >= :WKITEM
.
.
WKITEM  DC      H'19099'
USERID  DC      CL8'SQLDBA '
PASSWORD DC     CL8'SQLDBA '
        DC      C'CMDMSG'
.
.
.

```

X
X
X
-a- X

-b-
-b-

DQSMCMD ASMSQLDS

```

*****
*
* THIS MODULE PROCESSES ALL SQL COMMANDS THAT ARE VALID FOR BOTH
* THE COMPILE AND EXECUTION PHASE.  THE CALLING MODULE MUST CREATE
.
.
.
MESSAGE DS OH
EXEC SQL
    DECLARE C1 CURSOR FOR
        SELECT "SQL/DS HELP"
        FROM SQLDBA.SYSTEXT1 TEXT1,
        SQLDBA.SYSTEXT2 TEXT2
        WHERE TGPIC = :CMDMSTPC AND
        TEXT1.ITEM = TEXT2.ITEM

```

INSTALLING CMS SQL SUPPORT**EZKEYSQL EXEC**

The EZKEYSQL EXEC is provided on File 3 of the Installation Tape. You should tailor it to your specific environment, as described below:

```
&TRACE
* EZKEYSQL EXEC:
*
*   This EXEC builds the EZKEYSQL TXTLIB for the SQL/DS Interface
*   on the EZ/KEY residence minidisk.
*
*   This EXEC invokes the IBM supplied SQLINIT EXEC in order to
*   copy ARISISBT MODULE and ARISRMBT MODULE to your A-disk.
*
* Installation tailorable symbolic definitions:
*
*   &EZKEYUID = USERID that owns the EZ/KEY minidisk
*   &EZKEYCUU = address of EZ/KEY minidisk in USERID's directory
*   &EZKEYVUU = virtual address to LINK the EZ/KEY minidisk to
*   &EZKEYMOD = CMS Filemode to ACCESS the EZ/KEY minidisk as
*   &SQLDSUID = USERID that owns the SQL/DS minidisk
*   &SQLDSCUU = address of SQL/DS minidisk in USERID's directory
*   &SQLDSVUU = virtual address to LINK the SQL/DS minidisk to
*   &SQLDSMOD = CMS Filemode to ACCESS the SQL/DS minidisk as
*   &SQLPSWRD = password for the SQLDBA userid
*   &SQLDBASE = the name of your installation's SQL/DS database
*
* Modify the following based on the above definitions:
*
*   &EZKEYUID = userid
*   &EZKEYCUU = cuu
*   &EZKEYVUU = vuu
*   &EZKEYMOD = fm
*   &SQLDSUID = userid
*   &SQLDSCUU = cuu
*   &SQLDSVUU = vuu
*   &SQLDSMOD = fm
*   &SQLPSWRD = password
*   &SQLDBASE = dbname
```

continued ...

... continued

```

*
* LINK and ACCESS the EZ/KEY minidisk:
*
  CP LINK &EZKEYUID &EZKEYCUU &EZKEYVUU WR
  ACCESS &EZKEYVUU &EZKEYMOD
*
* Link and access SQL/DS residence disk, and invoke SQLINIT:
*
  CP LINK &SQLDSUID &SQLDSCUU &SQLDSVUU RR
  ACCESS &SQLDSVUU &SQLDSMOD
  EXEC SQLINIT DBNAME(&SQLDBASE )
*
* Verify access to IBM SQL/DS modules and SQL/CMS minidisk:
*
  STATE ARISISBT MODULE *
  &IF &RC NE 0 &GOTO -ERROR1
  STATE ARISRBMT MODULE *
  &IF &RC NE 0 &GOTO -ERROR1
  STATE ARIRVSTC TEXT *
  &IF &RC NE 0 &GOTO -ERROR1
*
* Update the EZKEYSQL TXTLIB with the IBM module ARIRVSTC:
*
  TXTLIB ADD EZKEYSQL ARIRVSTC
*
* Preprocess Pansophic's SQL/DS Message Extract program and assemble
* the output from the preprocessor:
*
  &PROGNAME = DQSMMTB
  &PREPNAME = DQEKMTB
  &PROGTYPE = ASSEMBLE
  &ASMPARMS = SYSPARM(CMS)
  &CALL -SQLPREP
*
* Run the DQSMMTB program to create DQMTBLST ASSEMBLE source:
*
  ERASE DQMTBLST ASSEMBLE A
  ERASE DQMTBPRT PRINT A
  GLOBAL TXTLIB EZKEYSQL
  LOAD DQSMMTB ( START
*
* Check for error messages from the DQMTBLST program:
*
  STATE DQMTBPRT PRINT A
  &IF &RC = 0 &GOTO -ERROR4

```

continued ...

... continued

```

*
* Assemble the DQMTBLST ASSEMBLE source program:
*
    &PROGNAME = DQMTBLST
    &PROGTYPE = ASSEMBLE
    &ASMPARMS =
    &CALL -ASSEMBLE
*
* Update the EZKEYSQL TXTLIB with the DQMTBLST object just created:
*
    TXTLIB ADD EZKEYSQL DQMTBLST
*
* Preprocess Pansophic's SQL/DS Command Processor
* and assemble the output from the preprocessor:
*
    &PROGNAME = DQSMCMD
    &PREPNAME = DQEKCMD
    &PROGTYPE = ASMSQLDS
    &ASMPARMS =
    &CALL -SQLPREP
*
* Update the EZKEYSQL TXTLIB:
*
    TXTLIB ADD EZKEYSQL DQSMCMD
*
* Copy the EZKEYSQL TXTLIB to the EZ/KEY residence disk:
*
    ERASE EZKEYSQL TXTLIB &EZKEYMOD
    COPY EZKEYSQL TXTLIB A = = &EZKEYMOD ( REPLACE
    &IF &RC = 0 ERASE EZKEYSQL TXTLIB A
*
* Copy ASSEMBLE source programs to the EZ/KEY residence disk:
*
    COPY DQSMMTB ASSEMBLE A = = &EZKEYMOD ( REPLACE
    COPY DQSMCMD ASSEMBLE A = = &EZKEYMOD ( REPLACE
    COPY DQMTBLST ASSEMBLE A = = &EZKEYMOD ( REPLACE
-END
    RELEASE &EZKEYMOD ( DET
    RELEASE &SQLDSMOD ( DET
    FILEDEF * CLEAR
    &EXIT

```

continued ...

... continued

```

*
* Preprocess a program and assemble the output from the preprocessor:
*
-SQLPREP
    &EQUAL = =
    &NAME = &CONCAT OF (PREPNAME &EQUAL &PREPNAME.,KEEP,NOPRINT,
    &USER = &CONCAT OF USERID &EQUAL SQLDBA/ &SQLPSWRD )
    &STR1 = &CONCAT OF &NAME &USER
    &STR2 = &CONCAT OF SYSPRINT( &PROGNAME ) &BLANK
    &STR3 = &CONCAT OF SYSIN( &PROGNAME &BLANK ASMSQLDS &BLANK *)
    &STR4 = &CONCAT OF SYSPUNCH( &PROGNAME &BLANK ASSEMBLE)
    &STR5 = &CONCAT OF &STR1 &BLANK &STR2 &STR3 &STR4
    EXEC SQLPREP ASM PREPPARM &STR5
    &IF &RC NE 0 &GOTO -ERROR2
*
* Assemble a program:
*
-ASSEMBLE
    FILEDEF * CLEAR
    GLOBAL MACLIB DMSSP CMSLIB
    ASSEMBLE &PROGNAME ( &ASMPARMS
    &IF &RC NE 0 &GOTO -ERROR3
    &RETURN
-ERROR1
    &TYPE You do not have access to the CMS minidisks containing the
    &TYPE required SQL/DS modules ARISISBT, ARISRMBT or ARIRVSTC TEXT.
    &TYPE Modify the Symbolic parameters in this EXEC (EZKEYSQL EXEC)
    &TYPE to specify the SQL/DS minidisk, and reinvoke this EXEC.
    &EXIT 1
-ERROR2
    &TYPE Preprocessing of &PROGNAME ASSEMBLE has failed for some reason.
    &TYPE This could be due to an incorrect password, or due to errors
    &TYPE in the source program. Examine the file &PROGNAME LISTPREP to
    &TYPE determine the cause of the error.
    &EXIT 2
-ERROR3
    &TYPE Assembly of &PROGNAME &PROGTYPE has failed for some reason.
    &TYPE Examine the file &PROGNAME LISTING to determine the cause of
    &TYPE the error. Contact Pansophic Customer Service for assistance
    &EXIT 3
-ERROR4
    &TYPE Execution of &PROGNAME has failed for the following reasons:
    TYPE DOMTBPRP PRINT A
    &EXIT 4

```

Tailoring EZKEYSQL EXEC

Tailor the EZKEYSQL EXEC to your installation using the symbolic substitution parameters, as follows:

- &EZKEYUID - The userid of the virtual machine owning the minidisk where EZ/KEY is installed.
- &EZKEYCUU - The address of the minidisk in &EZKEYUID's directory where EZ/KEY is installed.
- &EZKEYVUU - The virtual address that you want the minidisk to be linked as in each EZ/KEY user's virtual machine.
- &EZKEYMOD - The CMS mode letter that you want the minidisk to be ACCESSed as in each EZ/KEY user's virtual machine.
- &SQLDSUID - The userid of the virtual machine owning the minidisk where SQL/DS is installed.
- &SQLDSCUU - The address of the minidisk in &SQLDSUID's directory where SQL/DS is installed.
- &SQLDSVUU - The virtual address that you want the minidisk to be linked as in each EZ/KEY user's virtual machine.
- &SQLDSMOD - The CMS mode letter that you want the minidisk to be ACCESSed as in each EZ/KEY user's virtual machine.
- &SQLPSWRD - The password for the SQLDBA userid, needed to preprocess the PANSOPHIC SQL Interface Command Processor module.
- &SQLDBASE - The name of your SQL/DS database at your installation. SQL/DS must be initialized for this user prior to use by EZ/KEY. This is accomplished by invoking the SQLINIT EXEC provided by IBM. Consult the SQL/DS support person at your installation for the correct database name to use.

Running EZKEYSQL EXEC

You can now execute the installation EXEC by entering:

EZKEYSQL

When the EZKEYSQL EXEC completes, four additional CMS files exist on the EZ/KEY residence minidisk:

```
DQSMCMD  ASSEMBLE
DQSMMTB  ASSEMBLE
DQMTBLST ASSEMBLE
EZKEYSQL TXTLIB
```

The TXTLIB contains the additional modules needed by EZ/KEY to access SQL/DS using the PANSOPHIC SQL Interface.

The EZKEYSQL TXTLIB requires approximately 150 1K minidisk blocks.

You should keep the three ASSEMBLE files in case you need to reinstall SQL support.

You can erase all other files created by the EZKEYSQL EXEC after installation of SQL support is complete.

Additional SQL/DS Requirements

Following the execution of the EZKEYSQL EXEC, you must invoke ISQL (an SQL/DS utility), CONNECT using the SQLDBA userid, and issue the command:

GRANT RUN ON DQEKCMD TO PUBLIC

USING SQL WITH EZ/KEY

To execute EZ/KEY with the SQL Interface, you must specify the name of the SQL TXTLIB in the EZKEY EXEC. To do this, specify:

```
&EZKEYSQL = EZKEYSQL
```

Execution Using EZ/KEY SQL Interface

After you have tailored the EZKEYSQL EXEC, the EZKEYSQL EXEC has run successfully, and you have issued the GRANT RUN ON DQEKCMD TO PUBLIC command as described earlier in this section, modify the EZKEY EXEC for SQL/DS so that EZ/KEY users can use the PANSOPHC SQL Interface to SQL/DS. This allows EZ/KEY users to syntax-check SQL commands embedded in their EASYTRIEVE PLUS programs.

SQL/DS USERIDS

The EZ/KEY SQL Interface under CMS invokes SQL/DS to perform the syntax checking of the SQL statements embedded in your EASYTRIEVE PLUS programs.

EZ/KEY passes a userid and password to the SQL Interface; these are based on what is coded on the PARM statement USERID parameter in the EASYTRIEVE PLUS program. The PARM USERID parameter is required for EASYTRIEVE PLUS to identify the appropriate userid to SQL/DS when the batch job runs. EZ/KEY users must be aware of this when coding SQL statements in their EASYTRIEVE PLUS programs. All security and access checking performed by SQL/DS is based on this userid, not the VM userid.

Your SQL/DS administrator must GRANT authority to access appropriate tables within SQL/DS to each userid, as required. You could make the SQL/DS userids match the VM userids at your site, eliminating a possible source of confusion for your EZ/KEY users.

OPTIONAL IDMS/IDD INSTALLATION

Your site must have IDMS/IDD installed on the VM/CMS system where EZ/KEY is running if EZ/KEY is to access the IDD from CMS. Refer to the following Cullinet publications for more detailed information on installing IDMS/IDD in CMS:

- *Cullinet System Software Integrated Installation Guide*
- *Cullinet System Software System Generation*

Installing EZ/KEY IDMS/IDD Support

The EZKEYIDD EXEC creates the EZKEYIDD TXTLIB, which contains the EZKYIDMS module needed by EZ/KEY to access the IDMS IDD Interface.

You should modify the EZKEYIDD EXEC to specify site-dependent information and then run it, as described in "Tailoring EZKEYIDD EXEC" and "Running EZKEYIDD EXEC" later in this section.

EZKEYIDD Installation EXEC

```
&TRACE
*
* EZKEYIDD EXEC:
*
*   This EXEC generates the EZKEYIDD TXTLIB needed to use the
*   IDMS/IDD interface from within EZ/KEY.
*
* Installation Tailorable symbolic definitions:
*
*   &EZKEYUID = USERID that owns the EZ/KEY minidisk
*   &EZKEYCUU = address of EZ/KEY minidisk in USERID's directory
*   &EZKEYVUU = virtual address to LINK the EZ/KEY minidisk to
*   &EZKEYMOD = CMS Filemode to ACCESS the EZ/KEY minidisk as
*   &IDMSUID  = USERID that owns the IDMS minidisk
*   &IDMSCUU  = address of IDMS minidisk in USERID's directory
*   &IDMSVUU  = virtual address to LINK the IDMS minidisk to
*   &IDMSMOD  = CMS Filemode to ACCESS the IDMS minidisk as
*   &IDMSLIB  = CMS Filename of the IDMS TXTLIB
*
* Modify the following based on the above definitions:
*
*   &EZKEYUID = userid
*   &EZKEYCUU = cuu
*   &EZKEYVUU = vuu
*   &EZKEYMOD = fm
*   &IDMSUID  = userid
*   &IDMSCUU  = cuu
*   &IDMSVUU  = vuu
*   &IDMSMOD  = fm
*   &IDMSLIB  = idmslib
*
* LINK and ACCESS the required minidisks
*
*   CP LINK &EZKEYUID &EZKEYCUU &EZKEYVUU WR
*   ACCESS &EZKEYVUU &EZKEYMOD
*
*   CP LINK &IDMSUID &IDMSCUU &IDMSVUU RR
*   ACCESS &IDMSVUU &IDMSMOD
```

continued ...

... continued

```
*
* copy the required object decks to a temporary CMS TEXT file
*
  FILEDEF INMOVE DISK &IDMSLIB TXTLIB &IDMSMOD ( MEMBER IDMS
  FILEDEF OUTMOVE DISK EZKYIDMS TEXT A ( DISP MOD
  MOVEFILE
  FILEDEF INMOVE DISK &IDMSLIB TXTLIB &IDMSMOD ( MEMBER IDMSINTB
  FILEDEF OUTMOVE DISK EZKYIDMS TEXT A ( DISP MOD
  MOVEFILE
  FILEDEF INMOVE DISK &IDMSLIB TXTLIB &IDMSMOD ( MEMBER IDMSCANC
  FILEDEF OUTMOVE DISK EZKYIDMS TEXT A ( DISP MOD
  MOVEFILE
*
* generate the TXTLIB and move it to the EZ/KEY residence minidisk
*
  TXTLIB GEN EZKEYIDD EZKYIDMS
  COPY EZKEYIDD TXTLIB A = &EZKEYMOD ( REP
*
* clean up temporary files
*
  ERASE EZKEYIDD TXTLIB A
  ERASE EZKYIDMS TEXT A
*
* RELEASE minidisks and return
*
  RELEASE &EZKEYMOD ( DET
  RELEASE &IDMSMOD ( DET
  &EXIT
```

Tailoring EZKEYIDD EXEC

The EZKEYIDD EXEC can be found on the minidisk where EZ/KEY resides. A copy is in the EXAMPLE PIELIB.

Tailor the EZKEYIDD EXEC to your installation using the symbolic substitution parameters, as follows:

- &EZKEYUID - The userid of the virtual machine owning the minidisk where EZ/KEY is installed.
- &EZKEYCUU - The address of the minidisk in &EZKEYUID's directory where EZ/KEY is installed.
- &EZKEYVUU - The virtual address that you want the minidisk to be linked as in your virtual machine while running this EXEC.
- &EZKEYMOD - The CMS mode letter that you want the minidisk to be ACCESSEd as while running this EXEC.
- &IDMSUID - The userid of the virtual machine owning the minidisk where IDMS is installed.
- &IDMSCUU - The address of the minidisk in &IDMSUID's directory where IDMS is installed.
- &IDMSVUU - The virtual address that you want the minidisk to be linked as in your virtual machine while running this EXEC.
- &IDMSMOD - The CMS mode letter that you want the minidisk to be ACCESSEd as while running this EXEC.
- &IDMSLIB - The name of the IDMS TXTLIB where the IDMS object modules IDMS, IDMSINTB, and IDMSCANC reside.

Running EZKEYIDD EXEC

After tailoring the EZKEYIDD EXEC, run it by entering:

EZKEYIDD

This EXEC should run with no errors, and should indicate a zero return code.

ADDITIONAL IDMS CONSIDERATIONS***Central Version Mode Operation***

For EZ/KEY to access the IDD using Central Version, your site must have IDMS-CV (Central Version) installed in an MVS virtual machine, a VSE virtual machine, or a CMS virtual machine, and the IDMS-CV must run on the VM/SP system where EZ/KEY is running.

To permit EZ/KEY users to access the IDD using Central Mode, add the following FILEDEF statement to the EZKEY EXEC:

```
FILEDEF SYSCTL DISK idms control fm
```

where *idms control* is the CMS filename and filetype of the control file for IDMS, and *fm* is the filemode where this control file is located.

If you want EZ/KEY to use Central Version Mode only, specify **CENTRAL=ONLY** in the IDMSOPTI macro instruction, assemble the macro invocation, and link the resulting IDMSOPTI object module into the EZKYIDMS module, as described below.

IDMSOPTI Generation

The following example illustrates how to code the IDMSOPTI macro to specify that execution must always use Central Version Mode:

```
IDMSOPTI SVC=255,                                X
          CVNUM=0,                                X
          CENTRAL=ONLY,                            X
          SYSCTL=SYSCTL,                          X
          NODENAM=,                                X
          DBNAME=                                  X
END
```

The CENTRAL=ONLY parameter indicates that only Central Version Mode is supported. The SVC, CVNUM, SYSCTL, NODENAM, and DBNAME parameters may be different for your installation. Consult your IDMS Data Base Administrator or a systems programmer who installed IDMS for assistance.

Linking IDMSOPTI with EZKYIDMS

To include IDMSOPTI in the EZKYIDMS module, do the following:

1. Insert the statements below into the EZKEYIDD EXEC just before the comment **generate the TXTLIB and move it to the EZ/KEY residence disk**. See "EZKEYIDD Installation EXEC" earlier in this section for a copy of the EXEC:

```
FILEDEF INMOVE DISK IDMSOPTI TEXT *  
FILEDEF OUTMOVE DISK EZKYIDMS TEXT A ( DISP MOD  
MOVEFILE
```

2. Run the modified EZKEYIDD EXEC to generate a new EZKYIDMS module that contains IDMSOPTI in a new EZKEYIDD TXTLIB.

For more information on using the IDMSOPTI macro instruction and for detailed information on operating IDMS-CV in CMS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- CMS Considerations

Accessing an MVS Central Version from CMS

You must have a version of the IDMS SVC module generated by the IDMSMSVC macro for use with EZ/KEY under CMS. The macro must specify VMCF=YES.

IDMSMSVC Generation

The following example illustrates how to specify the IDMSMSVC macro to produce a VMCF-capable version of the IDMS SVC module for MVS:

```
IDMSMSVC SVC=255,                                X
      MAXCVNO=4,                                  X
      DC=NO,                                       X
      SDS=NO,                                       X
      APF=NO,                                       X
      SVCXLEN=0,                                   X
      SVCXIT=NO,                                   X
      VMCF=YES                                     X
END
```

The VMCF=YES parameter indicates that the Virtual Machine Communication Facility is supported. The SVC, MAXCVNO, DC, SDS, APF, SVCXLEN, and SVCXIT parameters may be different for your installation. Consult your IDMS Data Base Administrator or a systems programmer who installed IDMS for assistance.

For more information regarding use of the IDMSMSVC macro instruction, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations
- CMS Considerations

Accessing a VSE Central Version from CMS

You must have a version of the IDMS SVC module generated by the IDMSMSVC macro for use with EZ/KEY under CMS. The macro must specify VMCF=YES.

IDMS\$SVC Generation

The following example illustrates how to specify the IDMS\$SVC macro to produce a VMCF-capable version of the IDMS SVC module for VSE:

```
IDMS$SVC SVC=255,           X
      MAXCVNO=4,             X
      OLDSVC=NO,             X
      VSE=YES,               X
      SDS=NO,                X
      DC=NO,                 X
      VMCF=YES               X
END
```

The VSE=YES parameter indicates VSE Release 1. Specify VSE=2 for VSE/SP Release 2 and above. The VMCF=YES parameter indicates that the Virtual Machine Communication Facility is supported. The SVC, MAXCVNO, OLDSVC, SDS, and DC parameters may be different for your installation. Consult your IDMS Data Base Administrator or a systems programmer who installed IDMS for assistance.

For more information regarding use of the IDMS\$SVC macro instruction, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- DOS/VS(E) Considerations
- CMS Considerations

Local Mode Operation

Local Mode Operation permits EZ/KEY users to access the IDMS/IDD without requiring Central Version on the CMS system where EZ/KEY runs. However, this also requires significantly more virtual storage for all of the IDMS modules that are loaded into the user's virtual storage. Therefore, you should increase the size of each user's virtual machine accordingly.

To permit EZ/KEY users to access the IDD using Local Mode, do the following:

1. Add CP LINK and ACCESS statements to the EZKEY EXEC. Link to and access the minidisks where the Cullinet modules and TXTLIBs or LOADLIBs reside; this provides access to all of the IDMS modules needed to execute IDMS in Local Mode. Also, link to and access the minidisks where the IDD database files reside; this provides access to the IDD databases to be accessed by EZ/KEY.
2. Add the name of the Cullinet TXTLIB to the GLOBAL TXTLIB statement in the EZKEY EXEC.
3. Add the following FILEDEF statements to the EZKEY EXEC:

```
FILEDEF SYSJRNL DISK idms journal fm
FILEDEF dbase1 DISK dbase1 database fm
...
...
FILEDEF dbaseN DISK dbaseN database fm
```

where *dbase1* is the DDNAME, and *dbase1 database* is the CMS filename and filetype of the IDMS database that contains an IDD dictionary. Consult your IDMS Data Base Administrator for assistance.

For more information regarding use of the IDMS\$SVC macro instruction, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- DOS/VS(E) Considerations
- CMS Considerations

IDMS OS Foreground Execution Considerations**Central Version Mode OS Foreground Operation**

For EASYTRIEVE PLUS to access the IDD using Central Version, your site must have IDMS-CV (Central Version) installed in an MVS virtual machine, a VSE virtual machine, or a CMS virtual machine, and the IDMS-CV must be on the VM/SP system where EZ/KEY is running.

To permit EZ/KEY users to access the IDD and IDMS databases from the OS version of EASYTRIEVE PLUS running under CMS using Central Mode, add the following FILEDEF statement to the EZTPLUS EXEC used for Foreground Execution of EASYTRIEVE PLUS under EZ/KEY:

```
FILEDEF SYSCTL DISK idms control fm
```

where *idms control* is the CMS filename and filetype of the control file for IDMS, and *fm* is the filemode where this control file is located.

For more detailed information on operating IDMS-CV in CMS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- DOS/VS(E) Considerations
- MVS Considerations
- CMS Considerations

Local Mode OS Foreground Operation

Local Mode Operation permits EZ/KEY users to access the IDMS/IDD without requiring Central Version on the CMS system where EZ/KEY runs. However, this also requires significantly more virtual storage for all of the IDMS modules that are loaded into the user's virtual storage. Adjust the size of each user's virtual machine accordingly.

To permit EZ/KEY users to access the IDD and IDMS databases from the OS version of EASYTRIEVE PLUS running under CMS using Central Mode, do the following:

1. Add CP LINK and ACCESS statements to the EZTPLUS EXEC used for Foreground Execution of EASYTRIEVE PLUS from EZ/KEY. Link to and access the minidisks where the Cullinet modules and TXTLIBs or LOADLIBs reside; this provides access to all of the IDMS modules needed to execute IDMS in Local Mode. Also, link to and access the minidisks where the IDMS database files reside; this provides access to all of the databases to be accessed by EASYTRIEVE PLUS.
2. Add the name of the Cullinet TXTLIB to the GLOBAL TXTLIB statement in the EZTPLUS EXEC.
3. Add the following FILEDEF statements to the EZTPLUS EXEC:

```
FILEDEF SYSJRNL DISK idms journal fm
FILEDEF dbase1 DISK dbase1 database fm
...
...
FILEDEF dbaseN DISK dbaseN database fm
```

where *dbase1* is the DDNAME, and *dbase1 database* is the CMS filename and filetype of an IDMS database, and *fm* is the CMS filemode where the database resides. There must be a FILEDEF in the EZTPLUS EXEC for each database that may be accessed from EASYTRIEVE PLUS. Consult your IDMS Data Base Administrator for assistance.

For more detailed information on operating IDMS-CV in CMS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- CMS Considerations

IDMS DOS Foreground Execution Considerations***Central Version Mode DOS Foreground Operation***

For EZ/KEY to access the IDD using Central Version, your site must have IDMS-CV (Central Version) installed in an MVS virtual machine, a VSE virtual machine, or a CMS virtual machine, and the IDMS-CV must be on the VM/SP system where EZ/KEY is running.

To permit EZ/KEY users to access the IDD using Central Mode, add the following statement to the EZTPLUS EXEC:

```
SET UPSI b
```

where *b* is the UPSI switch defined to indicate Central Version Mode, as specified in the IDMSOPTI module when IDMS was installed.

Refer to the following Cullinet publication for more detailed information on running IDMS applications in Local Mode under CMS and for information on the IDMSOPTI macro instruction: *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- CMS Considerations

Local Mode DOS Foreground Operation

To permit EZ/KEY users to access the IDD and IDMS databases from EASYTRIEVE PLUS running under CMS using Local Mode, you must install the OS version of EASYTRIEVE PLUS under CMS. Cullinet's IDMS-CMS is an OS based product, and does not run in the CMS-DOS environment. See the subsection "Local Mode OS Foreground Operation" earlier in this section for details on running EASYTRIEVE PLUS OS under CMS with IDMS.

For more detailed information on operating IDMS-CV in CMS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- CMS Considerations

IDMS OS Background Execution Considerations**Central Version Mode OS Background Operation**

If EASYTRIEVE PLUS is to access the IDD through Central Version, your site must have IDMS-CV (Central Version) installed on the MVS system where batch EASYTRIEVE PLUS runs. To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Central Mode under OS, you must identify the IDMS control file for use by EASYTRIEVE PLUS programs.

To do this, add the following DD statement to the EZTPLUS Header JCL for Background Execution in EZ/KEY:

```
//SYSCTL DD DSN=your.idms.control.file,DISP=SHR
```

where *your.idms.control.file* is the name of the control file for IDMS. Consult your IDMS Data Base Administrator for assistance. Refer to the *EASYTRIEVE PLUS Installation Guide* for more information.

For more detailed information on operating IDMS-CV in MVS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

Local Mode OS Background Operation

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs under OS that access IDMS and IDD using Local Mode, add the following DD statements to the EZTPLUS Header JCL for Background Execution in EZ/KEY:

1. Identify the load libraries where the Cullinet IDMS modules reside; this provides access to the modules needed to execute IDMS in Local Mode:

```
//STEPLIB DD DSN=your.EZTP.load.library,DISP=SHR
//          DD DSN=your.IDMS.load.library,DISP=SHR
```

where *your.IDMS.load.library* is the name of the load module library for IDMS.

Additional IDMS Considerations

2. Identify the IDMS journal file for use by EASYTRIEVE PLUS programs:

```
//SYSJRNL DD DSN=your.idms.journal.file,  
//          DISP=(,KEEP),UNIT=TAPE
```

where *your.idms.journal.file* is the name of a journal file for IDMS. Note that you can either allocate this dataset to disk or specify a DUMMY dataset.

3. Identify all of the databases accessed by EASYTRIEVE PLUS programs:

```
//dbfile1 DD DSN=your.idms.database.file1,DISP=SHR
```

```
//dbfileN DD DSN=your.idms.database.fileN,DISP=SHR
```

where *dbfileN* and *your.idms.database.fileN* are the DDname and dataset name, respectively, for each database file that may be accessed by your EASYTRIEVE PLUS programs.

There must be a DD statement in the EZTPLUS JCL for each database that is accessed by any EASYTRIEVE PLUS program running in Local Mode. Consult your IDMS Data Base Administrator for assistance. Refer to the *EASYTRIEVE PLUS Installation Guide* for more information.

For more detailed information on running IDMS applications in Local Mode under CMS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- CMS Considerations

IDMS DOS Background Execution Considerations**Central Version Mode DOS Background Operation**

If EASYTRIEVE PLUS is to access the IDD through Central Version, your site must have IDMS-CV (Central Version) installed on the DOS/VSE system where EASYTRIEVE PLUS runs.

For more detailed information on operating IDMS-CV in DOS/VSE, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- DOS/VS(E) Considerations

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Central Mode under DOS, add the following statement to the EZTPLUS Header JCL for Background Execution in EZ/KEY:

```
// UPSI b
```

where *b* is the appropriate UPSI switch, if one was specified in the IDMSOPTI module that was linked with EZTPA00 when EASYTRIEVE PLUS was installed. Consult your IDMS Data Base Administrator for assistance.

Refer to the *EASYTRIEVE PLUS Installation Guide* for more information about installing EASYTRIEVE PLUS with IDMS support.

Local Mode DOS Background Operation

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Local Mode under DOS, add the following statements to the EZTPLUS Header JCL for Background Execution in EZ/KEY:

1. Identify the phase libraries where the Cullinet IDMS phases reside; this provides access to the phases needed to execute IDMS in Local Mode:

```
// DLBL IDMS,'your.IDMS.phase.library'  
// EXTENT ,volser  
// LIBDEF CL,SEARCH=(EZTP,IDMS)
```

where *your.IDMS.phase.library* is the name of the phase or core image library for IDMS, and *volser* is the label of the disk volume where the phase library is located.

NOTE: For VSE/SP 2.1, change *CL* to **PHASE** on the LIBDEF statement in the above illustration.

2. Identify the IDMS journal file for use by EASYTRIEVE PLUS programs. Use either:

```
// DLBL SYSJRNL,'your.idms.journal.file'  
// EXTENT ,volser
```

or

```
// TLBL SYSJRNL,'your.idms.journal.file',,volser,,f
```

where *your.idms.journal.file* is the name of the journal file for IDMS, *volser* is the label of the disk volume where this journal file is located, and *f* is the file number if the journal file is on tape.

Additional IDMS Considerations

3. Identify all of the databases accessed by EASYTRIEVE PLUS programs:

```
// DLBL dbase1,'your.idms.database.file1',0,SD  
// EXTENT ,volser
```

```
// DLBL dbaseN,'your.idms.database.fileN',0,SD  
// EXTENT ,volser
```

where *dbaseN* and *your.idms.database.fileN* are the DLBL name and dataset name, respectively, and *volser* is the label of the disk volume where this database file is located.

There must be DLBL and EXTENT statements in the EZTPLUS JCL for each database that is accessed by any EASYTRIEVE PLUS program running in Local Mode. Consult your IDMS Data Base Administrator for assistance.

Refer to the *EASYTRIEVE PLUS Installation Guide* for more information about installing EASYTRIEVE PLUS with IDMS support.

For more detailed information on running IDMS applications in Local Mode under DOS/VSE, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- DOS/VS(E) Considerations

EZKSEGNN MESSAGES

The following messages may be issued by the program EZKEYSEG MODULE, the EZ/KEY Shared Segment Loader. A brief explanation of the probable causes is included below:

Message Number	Description
EZKSEG01	Invoking EZKEYMOD MODULE EZKEYSEG is not able to attach EZ/KEY in a DCSS. EZ/KEY is executed in the CMS user area.
EZKSEG02	Recursive Call to EZKEYSEG MODULE EZKEYSEG detected that EZ/KEY is already in the CMS user area.
EZKSEG03	Insufficient storage for EZ/KEY Not enough virtual storage is available to initialize EZ/KEY. DEFINE STORAGE 1M (or larger), re-IPL CMS, and try again.
EZKSEG04	Error loading EZKEY shared segment EZKEYSEG found that EZKEY DCSS is not yet generated into CP, the virtual machine size overlaps the EZKEY DCSS start address, or another DCSS was already loaded in same address range.
EZKSEG05	EZKEY DCSS not initialized The EZKEY DCSS has been generated into CP, but it has not yet been initialized. See the procedure in this section on running the EZKEYSEG EXEC to install EZ/KEY in a DCSS.

Message Number	Description
EZKSEG06	<p>EZKEY DCSS does not contain EZ/KEY</p> <p>The EZKEY DCSS has been generated into CP, and it contains something, but not EZ/KEY. See the procedure in this section for running the EZKEYSEG EXEC to install EZ/KEY in a DCSS.</p>
EZKSEG07	<p>EZKEYMOD MODULE not found</p> <p>EZKEYMOD MODULE was not found on any ACCESSED CMS minidisk. The EZKEY EXEC must LINK to and ACCESS the correct minidisk.</p>
EZKSEG08	<p>Invalid CMS SUBSET Command</p> <p>A user issued 'CMS EZKEY' on the command line of EZ/KEY. This is not supported.</p>
EZKSEG09	<p>Insufficient storage for EZKEYSEG</p> <p>Not enough virtual storage was available to initialize EZKEYSEG. DEFINE STORAGE 1M (or larger), re-IPL CMS, and try again.</p>

SECTION 3 TSO INSTALLATION

Introduction	3-1
Storage Estimates	3-1
Reinstalling EZ/KEY under TSO	3-3
Reinstallation Procedure	3-3
Installation Files	3-4
Memory Requirements	3-5
TSO Disk Space Requirements	3-5
Loading Installation JCL	3-6
JCL Notes	3-6
Installation JCL	3-7
Tailoring TSO Installation JCL	3-8
EZKEY CLIST	3-10
Tailoring EZKEY CLIST	3-12
EZTPLUS CLIST	3-15
Tailoring EZTPLUS CLIST	3-17
EZ/KEY Foreground Execution CLIST Considerations	3-18
CLISTs in a SYSPROC Library	3-18
Instructions for Placing CLISTs in a SYSPROC Library	3-18
CLISTs in a PIELIB Library	3-19
Instructions for Placing CLISTs in a PIELIB Library	3-20
Choosing the Size of PIELIBs	3-21
Number of Blocks	3-21
Size of Each Block	3-21
Optional TSO Installation Procedures	3-22
Installation as a TSO Command Processor	3-22
Customizing EZ/KEY TSO Allowed Commands	3-23
Customizing EZ/KEY TSO Protected Commands	3-24
Installation in the MVS Link Pack Area	3-26
Installation in an MVS/XA Environment	3-26
Validate TSO Installation	3-27
Using VSAM PIELIBs with EZ/KEY	3-28
EASYTRIEVE PLUS VSAM Macro Libraries	3-29

PANVALET Considerations for TSO	3-30
PANVALET Security for TSO	3-31
PANVALET/TSO Memory Usage	3-32
TSO PANVALET Version 14.0 or Above Considerations	3-33
TSO/TCAM Considerations	3-34
TSO/TCAM Message Handler	3-34
Stage 1 Modifications	3-34
Stage 2 Modifications	3-34
TSO/VTAM Considerations	3-36
Installing EZ/KEY Under ISPF/PDF	3-37
Optional SQL Installation Procedures	3-38
EZ/KEY SQL Interface	3-38
Requirements	3-38
Disk Space Requirements	3-38
Virtual Memory Requirements	3-38
SQL Interface Option Installation	3-38
Loading SQL Installation JCL	3-39
JCL Notes	3-39
SQL Installation JCL (File 6)	3-40
Tailoring SQL Installation JCL	3-43
Optional IDMS/IDD Installation	3-44
IDMS/IDD Installation JCL	3-44
Tailoring IDMS/IDD JCL	3-44
Additional IDMS Considerations	3-45
Central Version Mode Operation	3-45
Local Mode Operation	3-45
IDMS Foreground Execution Considerations	3-46
Central Version Mode Foreground Operation	3-46
Local Mode Foreground Operation	3-47
IDMS Background Execution Considerations	3-48
Central Version Mode Background Operation	3-48
Local Mode Background Operation	3-48
IDMS/IDD Documentation	3-49

INTRODUCTION

This section contains detailed procedures for installing EZ/KEY under TSO. You should also review the *Administrator's Guide* to establish how you intend to operate EZ/KEY at your site.

The following procedures install the EZ/KEY system into a separate load module library. The EZKEY and EZTPLUS CLISTs are installed as members of a PDS. This PDS can then be used as a SYSPROC library. After ensuring that EZ/KEY operates successfully in your environment, you can improve performance and/or usability by:

- Copying the EZKEY and EZTPLUS CLISTs to your SYSPROC library
- Installing EZ/KEY as a TSO Command Processor.

Installing EZ/KEY as a TSO Command Processor allows EZ/KEY users to issue most of the TSO commands (such as ALLOCATE and FREE) during an EZ/KEY session.

Storage Estimates

EZ/editor is an in-storage editor, similar to SPF or ISPF/PDF. The size of the program or data being edited is limited by the availability of virtual storage. EZ/KEY requires a minimum region size of one megabyte when it is executed in the TSO user's address space.

These storage estimates are subject to user activities and installation options; because of these activities and options, you may need additional virtual storage to use EZ/KEY. The following factors can affect virtual storage availability:

- Whether EZ/KEY is installed in the LPA
- Which MVS modules are installed in the LPA
- Whether EZ/KEY is executed under ISPF
- The blocksizes of the PIELIBs
- The number of statements in the member being edited

If EZ/KEY is executed under ISPF/PDF, the region size must be large enough to accommodate both ISPF and EZ/KEY. For example, if ISPF runs in a region size of one megabyte, a region size of two megabytes should accommodate both ISPF and EZ/KEY.

Memory requirements, disk space sizes, JCL, and CLISTs presented in this Section are for planning purposes only. More precise information is provided in the files on the EZ/KEY distribution tape.

REINSTALLING EZ/KEY UNDER TSO

If you have already installed EZ/KEY under TSO, almost everything that you need is already in place. For Version 3.2 of EZ/KEY the CLISTs have been revised.

Reinstallation Procedure

To reinstall EZ/KEY under TSO, follow this procedure:

1. Review the output produced by the installation jobstream when the last version of EZ/KEY was installed. If you cannot locate this information, review the appropriate material in the previous version of the EZ/KEY Installation Guide.
2. Review the new sample CLISTs for EZ/KEY and compare them with your current CLISTs. Note the differences. Make the necessary changes to the new CLISTs for EZ/KEY.
3. Review the new installation JCL. Make any changes needed for your installation's environment.
4. Once you have made the necessary changes, run the new installation procedure, and then verify the results as described in this section under "Verify TSO Installation."

INSTALLATION FILES

The EZ/KEY system for TSO is distributed on a standard labeled tape that contains six files:

- File 1 (DSN=FILE01) - Contains the Job Control Language (JCL) in IEBGENER format, to install EZ/KEY. This file is blocked 80/3120.
- File 2 (DSN=FILE02) - Contains the EZKEY module in IEBCOPY format. The original load library is blocked 6144.
- File 3 (DSN=FILE03) - Contains the EZKEY and EZTPLUS CLISTs, the assembler language source for PIESSTSA and PIESSTSN, and the EZ/REPORT User's Guide, in IEBCOPY format. The original source library is blocked 80/3120.
- File 4 (DSN=FILE04) - Contains the SYSTEM PIELIB in IEBGENER format. This file is unblocked 4089.
- File 5 (DSN=FILE05) - Contains the EXAMPLE PIELIB in IEBGENER format. This file is unblocked 2041.
- File 6 (DSN=FILE06) - Contains the JCL and object code in IEBGENER format, to install the SQL Interface for EZ/KEY. This file is blocked 80/3120.

Memory Requirements

The region size should usually be increased by 1024K for the EZ/KEY modules and by an additional amount that depends on what users are doing with EZ/KEY at the time. Use the following formula to compute your user memory requirements:

48K (initial requirements to sign on)
plus 200 bytes times number of lines being edited

EZ/KEY is an in-storage editor. One hundred bytes of storage are required per line being edited, plus approximately 100 bytes per line while using EZ/checker.

For example: you anticipate that the largest member to be edited by any user is 1000 lines:

$$\begin{array}{r}
 1000 \times 200 = 200 \text{ K} \\
 + 48 \text{ K} \\
 \hline
 248 \text{ K}
 \end{array}$$

If PANVALET is used with EZ/KEY, allow an additional 120K for the PANVALET module, 10K for each connected PANVALET library, and 10K for each enabled PANVALET library.

TSO Disk Space Requirements

The approximate disk space required for TSO EZ/KEY is:

Name	Disk Blocks	Dataset Blocksize	3350 Tracks
-----	-----	-----	-----
EZ/KEY load library	100	6144	20
EZ/KEY source library	50	3120	10
SYSTEM PIELIB	1500	4089	300
EXAMPLE PIELIB	400	2041	50

Loading Installation JCL

File 1 on the installation tape contains the JCL you need to install EZ/KEY. The following JCL copies file 1 from the installation tape to a sequential data set. This JCL must be modified as described below.

```
//jobname1 JOB accounting.info ***-1-***
//INSTJCL EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=FILE01,UNIT=TAPE,DISP=OLD,
// LABEL=(1,SL),VOL=SER=volser ***-2-***
//SYSUT2 DD DSN=userid.EZKEY.JCL, ***-3-***
// DISP=(,CATLG,DELETE),
// DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
// UNIT=SYSDA,VOL=volser, ***-4-***
// SPACE=(3120,(100,25),RLSE)
```

JCL Notes

At the right of some of the JCL lines shown above is the character string *****-n-*****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in these notes:

1. Modify the JOB name and accounting information as needed.
2. Specify the volser of the EZ/KEY Installation Tape.
3. Specify the dataset name to use for the sequential dataset to contain the EZ/KEY Installation JCL, or modify this statement to place the JCL into a member of a PDS of your choice.
4. Specify the volser of the disk to contain the above dataset.

INSTALLATION JCL

The following JCL is on the first file of the EZ/KEY distribution tape. You must load this file onto your system and edit it. Items that you should change are identified by ***-n-*** to the right:

```
//jobname JOB accounting.info
//EZKINST PROC EZKPREF=prefix,
//          EZKTAP='123456',
//          EZKVOL='SER=volser'
//*****
//* COPY THE EZ/KEY SOURCE AND LOAD LIBRARIES
//*****
//COPY      EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD UNIT=SYSDA,SPACE=(80,(120,112))
//SYSUT4 DD UNIT=SYSDA,SPACE=(80,(15,7))
//FILE02 DD DSN=FILE02,UNIT=TAPE,DISP=(OLD,PASS),
//          LABEL=2,VOL=(PRIVATE,RETAIN,SER=&EZKTAP)
//FILE03 DD DSN=FILE03,UNIT=TAPE,DISP=(OLD,PASS),
//          LABEL=3,VOL=REF=*.COPY.FILE02
//LOAD      DD DSN=&EZKPREF..EZKEY.LOAD,DISP=(,CATLG),VOL=&EZKVOL,
//          UNIT=SYSDA,SPACE=(6144,(250,50,10),RLSE)
//SOURCE DD DSN=&EZKPREF..EZKEY.SOURCE,DISP=(,CATLG),VOL=&EZKVOL,
//          UNIT=SYSDA,SPACE=(3120,(50,50,10),RLSE)
//SYSIN DD DUMMY
//*****
//* COPY THE SYSTEM PIE LIBRARY
//*****
//SYSTEM EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=FILE04,UNIT=TAPE,DISP=(OLD,PASS),
//          VOL=REF=*.COPY.FILE02,LABEL=4
//SYSUT2 DD DSN=&EZKPREF..SYSTEM.PIELIB,
//          DISP=(,CATLG,DELETE),UNIT=SYSDA,VOL=&EZKVOL,
//          SPACE=(4089,(1500,150),,CONTIG,ROUND)
//SYSIN DD DUMMY
```

continued ...

... continued

```

//*****
//*   COPY THE EXAMPLE PIE LIBRARY
//*****
//EXAMPLE EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=FILE05,UNIT=TAPE,DISP=(OLD,KEEP),
//          VOL=REF=*.COPY.FILE02,LABEL=5
//SYSUT2 DD DSN=&EZKPREF..EXAMPLE.PIELIB,
//          DISP=(,CATLG,DELETE),UNIT=SYSDA,VOL=&EZKVOL,
//          SPACE=(2041,(400,4),,CONTIG,ROUND)
//SYSIN DD DUMMY
//          PEND
//EZKINSTL EXEC EZKINST
//COPY.SYSIN DD *
//          COPY INDD=((FILE02,R)),OUTDD=LOAD
//          COPY INDD=((FILE03,R)),OUTDD=SOURCE
//          /*
//          //

```

Tailoring TSO Installation JCL

At the right of some of the JCL lines shown above is the character string *****-n-*****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in these notes:

1. Modify the JOB name and accounting information as needed.
2. Specify the dataset name prefix for EZ/KEY datasets.
3. Specify the volume serial of the EZ/KEY distribution tape.
4. Specify the volume serial of the disk pack where you want the EZ/KEY datasets to reside.

Installation JCL

You should copy the EZKEY and EZTPLUS CLISTs from the SOURCE library to your SYSPROC library so that users can invoke EZ/KEY by entering:

EZKEY or ***%EZKEY***

Otherwise, EZ/KEY must be invoked by entering:

EX 'prefix.EZKEY.CLIST'

The CLISTs are distributed with no sequence numbers so they can be copied directly into a SYSPROC library.

EZKEY CLIST

The EZKEY CLIST (copied from the third file of the distribution tape) is contained in the EZKEY.SOURCE library. The CLIST's member name is EZKEY. The CLIST is shown below:

```

PROC O PARM() PREFIX(NOPREFIX)
CONTROL NOFLUSH
/*-----*/
/* Invoke EZ/KEY as a TSO Command Processor or as a CALLED program. */
/* */
/*   For Foreground Execution, */
/*   execute EASYTRIEVE PLUS via '&EZKPREF..PIETEMP.CLIST', */
/*   cycle back and reinvoke EZKEY. */
/* */
/* Invoking Parameters: */
/* */
/*   PARM   = Initial Menu Selection value */
/*           (e.g., 3.3 = go directly to the Copy/Move Panel) */
/* */
/*   PREFIX = to override default PREFIX for TSO Libspace */
/* */
/* Installation Tailorable symbolic definitions: */
/* */
/*   &EZKPREF = PREFIX for EZ/KEY user TSO sequential datasets */
/* */
/*   &EZKQUAL = prefix for your installation's EZ/KEY datasets */
/* */
/*   &EZKMODE = 'CALL' to invoke EZ/KEY via CALL, */
/*             'CP'   to invoke EZ/KEY as a Command Processor */
/* */
/*   &EZKAUTO = 'YES'  permits anyone to use EZ/KEY automatically */
/*             'NO'   System Administrator must create new users */
/* */
/*   &EZKBLKSZ = blocksize to use for new BDAM userid.PIELIBs */
/* */
/*   &EZKSPACE = initial allocation for new BDAM userid.PIELIBs */
/* */

```

continued ...

.... continued

```

/*      &EZKALLOC = 'CYL'  allocate space in cylinders      */
/*      'TRK'      allocate space in tracks                 */
/*      'BLK'      allocate space in blocks                 */
/*      */
/*      &EZKCLASS = SYSOUT class for EZ/KEY printed output  */
/*      */
/* Modify the following 9 lines based on the above definitions: */
/*-----*/
      IF &PREFIX = NOPREFIX THEN SET &EZKPREF = &PREFIX
      ELSE SET &EZKPREF = &SYSPREF

      SET &EZKQUAL = prefix
      SET &EZKMODE = CALL
      SET &EZKAUTO = YES
      SET &EZKBLKSZ = 2048
      SET &EZKSPACE = 1
      SET &EZKALLOC = CYL
      SET &EZKCLASS = X
INIT: FREE F(&SYSUID, SYSTEM, EXAMPLE, INPUT, OUTPUT, EZKPRT, EZKPUN)
      FREE ATTRLIST(EZKEYPRT)
      ALLOC F(SYSTEM) DA('&EZKQUAL..SYSTEM.PIELIB') SHR
      ALLOC F(EXAMPLE) DA('&EZKQUAL..EXAMPLE.PIELIB') SHR
      ALLOC F(&SYSUID) DA('&SYSUID..PIELIB') SHR
      IF &LASTCC NE 0 THEN DO
        IF &EZKAUTO = YES THEN DO
          FREE ATTRLIST(EZKLIB)
          ATTRIB EZKLIB RECFM(F) LRECL(&EZKBLKSZ) BLKSIZE(&EZKBLKSZ)
          ALLOC F(EZKUSER) DA('&SYSUID..PIELIB') NEW +
            SPACE(&EZKSPACE) &EZKALLOC USING(EZKLIB)
          OPENFILE EZKUSER OUTPUT
          CLOSFILE EZKUSER
          FREE F(EZKUSER)
          FREE ATTRLIST(EZKLIB)
          ALLOC F(&SYSUID) DA('&SYSUID..PIELIB') SHR
          END
        ELSE DO
          WRITE EZKEY008 - User not authorized to use EZ/KEY.
          EXIT
          END
        END
      END

```

continued ...

... continued

```

      ALLOC F(INPUT)  DA(*)
      ALLOC F(OUTPUT) DA(*)
      ATTRIB EZKEYPRT RECFM(F A) LRECL(133)
      ALLOC F(EZKPRT) SYSOUT(&EZKCLASS) USING(EZKEYPRT)
      ALLOC F(EZKPUN) DUMMY
      IF &EZKMODE = CP THEN EZKEYTSO &PARM.:&EZKPREF
      ELSE CALL 'EZKQUAL..EZKEY.LOAD(EZKEYTSO)' '&PARM.:&EZKPREF'
      IF &LASTCC = 4 THEN DO
        EXEC 'EZKPREF..PIETEMP.CLIST'
        DELETE 'EZKPREF..PIETEMP.CLIST'
        WRITE
        WRITE Press ENTER to continue...
        READ
        SET &PARM = 4
        GOTO INIT
      END
      WRITE EZKEYO18 - EZ/KEY Completed
      EXIT

```

Tailoring EZKEY CLIST

Tailor the EZKEY CLIST to your installation using the symbolic substitution parameters, as follows:

&EZKPREF - The high-level qualifier to be used to prefix all sequential datasets created dynamically on behalf of the user by EZ/KEY, and used to copy sequential datasets from TSO into EZ/KEY. The default is &SYSPREF, the TSO user's PROFILE PREFIX, or the userid (&SYSUID) if PROFILE NOPREFIX is in effect.

The optional symbolic parameter PREFIX may be used to temporarily override the default prefix when invoking EZ/KEY. For example, user SMITH wants to enter EZ/KEY, but wishes to access userid JONES' datasets using the EZ/KEY TSO library. User SMITH should enter **EZKEY PREFIX(JONES)** from TSO. This stays in effect only for the duration of the EZ/KEY session.

EZKEY CLIST

If you have a security package such as RACF installed and you require a multiple-level prefix, such as *groupid.userid.dataset.name*, you can override the default by specifying "ELSE SET &EZKPREF = &SYSPREF.&SYSUID" in the EZKEY CLIST and setting your TSO PROFILE PREFIX to your groupid.

- &EZKQUAL - The high-level qualifier to be used to prefix all EZ/KEY datasets created when you installed EZ/KEY.
- &EZKMODE - The *mode* in which EZ/KEY operates: as a CALLED program, or as a Command Processor (CP). The default is CALL because it is easier to install and test initially. After successfully installing and testing EZ/KEY, you may wish to change this to CP. See "Installation as a TSO Command Processor" later in this section.
- &EZKAUTO - The method that EZ/KEY uses to create user PIELIBs: either "automatically" (YES) or not (NO). When &EZKAUTO = YES, EZ/KEY automatically allocates and initializes a PIELIB for each user the first time that user signs onto EZ/KEY. If you want to control who may use EZ/KEY, set &EZKAUTO to NO. A system administrator must then create a PIELIB for each user in order for that user to be able to sign on to EZ/KEY.
- &EZKBLKSZ - The blocksize that EZ/KEY uses when creating user PIELIBs. The default is 2048. Details about selecting an appropriate blocksize for PIELIBs can be found under "Choosing the Size of PIELIBs" later in this section. The valid range currently supported for PIELIBs is 505 to 4096 bytes.
- &EZKSPACE - The number of units EZ/KEY uses when creating user PIELIBs. The default is 1. Details about selecting an appropriate amount of space for user PIELIBs can be found under "Choosing the Size of PIELIBs" later in this section. The units are described next.

EZKEY CLIST

- &EZKALLOC - The units in which to allocate space when creating user PIELIBs. The default is CYL. You may specify CYL, TRK or BLK. Details about selecting an appropriate amount of space for user PIELIBs can be found under "Choosing the Size of PIELIBs" later in this section.
- &EZKCLASS - The SYSOUT class to use for EZ/KEY printed output. The default is X. You must specify a SYSOUT class that is valid for your installation. Consult your JES2 or JES3 system programmer for further information.

EZTPLUS CLIST

The EZTPLUS CLIST (copied from the third file of the distribution tape) is contained in the EZKEY.SOURCE library. Its member name is EZTPLUS. The CLIST is shown below:

```

PROC 7 PGM DDNAM DSNAM TYP DDNAM2 DSNAM2 PGM TYP
CONTROL NOFLUSH NOMSG
/*-----*
/* Invoke EASYTRIEVE PLUS as a CALLED program in the foreground.      *
/*                                                                    *
/*   This CLIST is invoked via '&EZKPREF..PIETEMP.CLIST',              *
/*   which is invoked by the EZKEY CLIST.                             *
/*                                                                    *
/* Invoking Parameters:                                              *
/*                                                                    *
/*   PGM      = the name of the EASYTRIEVE PLUS program to execute   *
/*                                                                    *
/*   DDNAM     = the DDNAME of the primary input file for the above  *
/*                                                                    *
/*   DSNAM     = the DSNAM of the primary input file for the above   *
/*                                                                    *
/*   TYP       = the type of output: 'B' for browse,                 *
/*               'D' for disk dataset,                               *
/*               'P' for printer, or                                 *
/*               'T' for terminal.                                     *
/*                                                                    *
/*   DDNAM2    = the DDNAME of the second input file for the above   *
/*                                                                    *
/*   DSNAM2    = the DSNAM of the second input file for the above    *
/*                                                                    *
/*   PGM TYP   = the type of the program input file, usually 'EZT'  *
/*                                                                    *
/* Installation Tailorable symbolic definitions:                     *
/*                                                                    *
/*   &EZKPREF  = PREFIX for EZ/KEY TSO sequential datasets           *
/*               (must be the same as &EZKPREF in EZKEY.CLIST)        *
/*                                                                    *
/*   &EZTPDSN  = DSNAM of installation's EASYTRIEVE PLUS library     *
/*                                                                    *
/*   &EZTCLASS = sysout class for EASYTRIEVE PLUS printed output     *
/*                                                                    *
/* modify the following 3 lines based on the above definitions:      *
/*-----*

```

continued ...

... continued

```

SET &EZKPREF = &SYSPREF
SET &EZTPDSN = your.EZTPLUS.LOADLIB
SET &EZTCLASS = A
FREE F(SYSIN, SYSPRINT, SYSOUT, EZTVFM)
IF      &TYP = T THEN ALLOC F(SYSPRINT) DA(*)
ELSE IF &TYP = P THEN ALLOC F(SYSPRINT) SYSOUT(&EZTCLASS)
ELSE IF &TYP = B | &TYP = D THEN DO
  DELETE '&EZKPREF..&PGM..LIST'
  FREE ATTRLIST(EZKLATTR)
  ATTR EZKLATTR RECFM(F B A) LRECL(133) BLKSIZE(3990)
  ALLOC F(SYSPRINT) DA('&EZKPREF..&PGM..LIST') NEW +
    SPACE(20 100) BLOCK(3990) USING(EZKLATTR)
END
ELSE ALLOC F(SYSPRINT) DA(*)
IF &DDNAM ^= 2 THEN DO
  IF &DSNAM = 3 THEN SET &DSNAM = NULLFILE
  FREE F(&DDNAM)
  ALLOC F(&DDNAM) DA(&DSNAM) SHR
END
IF &DDNAM2 ^= 6 THEN DO
  IF &DSNAM2 = 7 THEN SET &DSNAM2 = NULLFILE
  FREE F(&DDNAM2)
  ALLOC F(&DDNAM2) DA(&DSNAM2) SHR
END
ALLOC F(SYSOUT) DA(*)
ALLOC F(EZTVFM) TR SP(5 5)
IF &PGMTYP = 9 THEN SET &PGMTYP = EZT
IF &PGM = 1 THEN ALLOC F(SYSIN) DUMMY
  ELSE ALLOC F(SYSIN) DA('&EZKPREF..&PGM..&PGMTYP') SHR
CALL '&EZTPDSN.(EZTPA00)' ''
FREE F(SYSIN, SYSPRINT, SYSOUT, EZTVFM)
IF &DDNAM ^= 2 THEN FREE F(&DDNAM)
IF &DDNAM2 ^= 6 THEN FREE F(&DDNAM2)
IF &PGM ^= 1 THEN DELETE '&EZKPREF..&PGM..&PGMTYP'
EXIT

```

Tailoring EZTPLUS CLIST

Tailor the EZTPLUS CLIST to your installation using the symbolic substitution parameters, as follows:

- &EZKPREF** - The high-level qualifier to be used to prefix all sequential datasets created dynamically on behalf of the user by EZ/KEY, and used to copy sequential datasets from TSO into EZ/KEY. This must be the same qualifier as specified in the EZKEY CLIST.
- &EZTPDSN** - The dataset name where the EASYTRIEVE PLUS program modules reside.
- &EZTCLASS** - The SYSOUT class to use for EZ/KEY printed output. The default is A. You must specify a SYSOUT class that is valid for your installation. Consult your JES2 or JES3 system programmer for further information.

EZ/KEY FOREGROUND EXECUTION CLIST CONSIDERATIONS

The following discussion pertains to all of the CLISTs that you must tailor for use with EZ/KEY Foreground Execution.

When tailoring any CLISTs for use with EZ/KEY Foreground Execution, you can store these CLISTs in one of several places for use by all of your EZ/KEY users.

CLISTs in a SYSPROC Library

One alternative is to place the tailored CLISTs in a SYSPROC library (partitioned dataset) that all EZ/KEY users have access to and have your EZ/KEY users specify % for the CLIST Type. Modify the EZKEY.CLIST to FREE F(SYSPROC) and then ALLOC F(SYSPROC) to the appropriate dataset(s). If the appropriate dataset is not already in the SYSPROC concatenation, or if there is no SYSPROC allocated in their LOGON PROC, change the users LOGON PROC to add the appropriate library to their SYSPROC concatenation.

When a user specifies the name of the CLIST to use for foreground execution and specifies % for the CLIST Type, EZ/KEY assumes that the CLIST is available in the currently allocated SYSPROC concatenation. If this is not the case, TSO error messages are displayed on the user's terminal.

Instructions for Placing CLISTs in a SYSPROC Library

To implement this alternative, use the following steps to tailor the CLISTs:

1. Copy the CLIST into the SYSPROC library where you want the CLIST to permanently reside, using ISPF/PDF utility 3.3, or an equivalent utility.
2. Edit the CLIST you want to tailor in the SYSPROC library, using an editor such as ISPF/PDF or TSO EDIT.
3. After making your changes, save the CLIST back into the SYSPROC library where you want the CLIST to permanently reside.

CLISTs in a PIELIB Library

The other choice is to place the tailored CLISTs in a common library, such as EXAMPLE PIELIB, and tell your EZ/KEY users to specify *EX* for CLIST Type (the default).

When a user specifies the name of the CLIST to use for foreground execution and specifies *EX* as the CLIST type, EZ/KEY searches all enabled PIELIB LIBSPACEs for a CLIST by that name. If EZ/KEY finds a CLIST by that name in any enabled PIELIB LIBSPACE, EZ/KEY copies that CLIST to the LIBSPACE named TSO, which is a sequential dataset named 'prefix.clistname.CLIST', and then executes that copy of the CLIST. If EZ/KEY cannot find a CLIST by that name, EZ/KEY assumes that the CLIST is available in a sequential dataset with the dataset name *prefix.clistname.CLIST*. If this is not the case, TSO error messages are displayed on the user's terminal. This maintains compatibility with previous releases of EZ/KEY.

To use this alternative, your EZ/KEY users must have EXAMPLE PIELIB (or another common PIELIB) enabled. This way, these CLISTs are always available to your EZ/KEY users for use with foreground execution. You can accomplish this in one of two ways:

- Instruct your users to enable EXAMPLE (or another common library) using utility option 3.2.

Or

- Use the Report Administrator Utility functions to have EZ/KEY automatically enable EXAMPLE for all or selected users. See Section 7 of the *EZ/KEY Administrator's Guide* for detailed instructions on how to do this. Specifically, see the subsection on establishing default CONNECT and LIBCHAIN members.

Instructions for Placing CLISTs in a PIELIB Library

To implement this alternative, use the following steps to tailor the CLISTs:

1. Disable EXAMPLE PIELIB using utility option 3.2.
2. Select all of the CLISTs in EXAMPLE by placing **S** next to **EXAMPLE**, and typing **CLIST** under **Member Type** on the PROULSM panel.
3. Edit the CLIST you want to tailor by typing **ED** next to the member name.
4. After making your changes, press the END PF key. This saves the member in your library.
5. Use utility option 3.a.3 to CONNECT EXAMPLE PIELIB with CREATE Access Rights. Fill in **EXAMPLE** for **LIBSPACE Name**, **CREATE** for **Access Rights**, and press ENTER.
6. Use utility option 3.3 to COPY the member(s) you modified from your library to the EXAMPLE library. Specify **YES** for the Replace option.
7. Use utility option 3.2 to enable EXAMPLE if you want to have it normally enabled for your userid.
8. Optionally, you can use the Report Administrator functions of EZ/REPORT to enable EXAMPLE automatically for all or selected users.

You can use a library other than EXAMPLE PIELIB if you have one or more *common* PIELIBs for different groups of users in your environment. To do this, just substitute the name of your common library for **EXAMPLE** in the above instructions.

In this way, you can set up different versions of the Foreground Execution CLISTs to satisfy varying requirements of different departments while maintaining standard naming conventions. For example, you can have a different EZTPLUS.CLIST for each department, and you can instruct all EZ/KEY users to specify the name EZTPLUS as the name to use for Foreground Execution. Each user uses the correct EZTPLUS.CLIST for the department that they belong to.

CHOOSING THE SIZE OF PIELIBS***Number of Blocks***

Select the number of blocks in the LIBSPACE (minimum of 100). This should be large enough for future expansion. A LIBSPACE cannot be expanded easily.

Size of Each Block

Enter the number of bytes for each block. EZ/KEY uses 10 bytes of each block for internal pointers. The blocksize should approximate the average size of members in the LIBSPACE. Large blocksizes waste disk space if most members are small; small blocksizes incur unnecessary I/O if most members are larger than one block.

For PIEVSAM, the blocksize must be the CISIZE minus 7 bytes (for VSAM overhead). 505, 1017, 2041, or 4089 are good choices.

PIEBDAM PIELIBs on FBA devices should have blocksizes equal to or an exact multiple of 512. PIEBDAM PIELIBs on non-FBA devices should have blocksizes equal to or slightly less than one fourth of the DASD track length. Consult the IBM Reference Manuals for Direct Access Storage (e.g., GA26-1638) or Reference Summary (e.g., GX20-1983) for help in choosing an optimal blocksize.

Each record must be wholly contained in one block. The maximum recordsize is 10 characters less than the PIELIB blocksize (no spanned records).

Each block contains records from only one member. The LIBSPACE blocksize should be large enough to accommodate the largest record but small enough to avoid wasted space at the end of the block (505 - 4096 is the allowed range).

Each member must be wholly contained in one LIBSPACE. The size of the LIBSPACE (blocksize, number of blocks) must be large enough for the largest member.

One LIBSPACE must contain all members within a library.

With a blocksize of 1024, a 100-block LIBSPACE can hold 10 members with 200 80-byte records reach (approximately 2,000 80-byte records or about 160,000 characters).

OPTIONAL TSO INSTALLATION PROCEDURES

Installation as a TSO Command Processor

Installing EZ/KEY as a TSO Command Processor allows EZ/KEY users to issue a subset of TSO commands during an EZ/KEY session by entering ***TSO command parameters*** on the command line of any EZ/KEY panel. When EZ/KEY is not installed as a TSO Command Processor, the message **TSO NOT ALLOWED** is displayed if an attempt is made to execute a TSO command or CLIST.

To install EZ/KEY as a TSO Command Processor, edit the EZKEY CLIST and change the statement:

```
SET &EZKMODE = CALL
```

to:

```
SET &EZKMODE = CP
```

Then do *one* of the following:

- Copy the EZKEYTSO and EZKYLAM modules to a link-list library. (If you install SQL support, also copy the DQSEKCF, DQSEKCI, DQSEKCR, DQSEKCS, and DQSEKCT modules to a link-list library. If you install IDMS support, also copy the EZKYIDMS module to a link-list library.)

Or

- Include the name of the EZ/KEY load library in the LNKLISTxx member of SYS1.PARMLIB.

Or

- Add the EZ/KEY load library as a STEPLIB in each EZ/KEY user's LOGON PROC.

EZ/KEY is not designed to run APF authorized under MVS. Attempting to execute an authorized TSO command may result in a 106 or 047 abend and the TSO Terminal Monitor Program (TMP) displays a **READY** message. The user should press ENTER to allow EZ/KEY to continue from the point of failure. If recovery is successful, EZ/KEY displays a message such as **SYSTEM 047 USER 000** and the EZ/KEY session continues executing. If the TMP issues another **READY** message, EZ/KEY was unable to recover from the abend and the user must reinvoked EZ/KEY.

Customizing EZ/KEY TSO Allowed Commands

For security or integrity reasons, you may want EZ/KEY users to be able to only issue certain TSO commands from within EZ/KEY. To permit you to specify which commands to allow, EZ/KEY contains a table of allowed commands.

The assembler language source code for the PIESSTSA CSECT is contained in the member PIESSTSA. This member is in the SOURCE library copied from file 3 of the distribution tape. The list of commands is a sequence of 8-byte entries. Each entry contains a left-justified command name, padded with blanks. An entry of 8 spaces indicates the end of the list. Any TSO abbreviations must also be in the list. Extra blank entries are provided at the end of the list so that you can add command names using AMASPZAP. The commands in the default PIESSTSA have been tested, but may not necessarily operate correctly in all environments due to changes in IBM code. You should review the command names in PIESSTSA to ensure that they are appropriate for your environment.

The PIESSTSA CSECT link-edited into EZ/KEY contains a list of the TSO commands that EZ/KEY allows to be executed. If the command is not found in this list, the message **xxxxxxx NOT ALLOWED** is displayed, where **xxxxxxx** is the command the user entered. The commands contained in PIESSTSA are:

DC	CL8'	'	FORCE USE OF PIESSTSP
DC	CL8'ALLOC	'	
DC	CL8'ALLOCATE	'	
DC	CL8'ATTR	'	
DC	CL8'ATTRIB	'	
DC	CL8'EX	'	
DC	CL8'EXEC	'	
DC	CL8'FREE	'	
DC	CL8'LISTA	'	
DC	CL8'LISTALC	'	
DC	CL8'LISTBC	'	
DC	CL8'LISTD	'	
DC	CL8'LISTDS	'	
DC	CL8'OUT	'	
DC	CL8'OUTPUT	'	
DC	CL8'REN	'	
DC	CL8'RENAME	'	
DC	CL8'SEND	'	
DC	CL8'	'	ALL BLANKS INDICATES
DC	CL8'	'	END OF TABLE

Optional TSO Installation Procedures

The following illustrates the JCL to link-edit your customized PIESSTSA into EZ/KEY:

```
//jobname JOB (accounting.info),MSGLEVEL=1,
//          CLASS=A,MSGCLASS=A
//LKED      EXEC PGM=IEWL,PARM='LIST,LET,NCAL,XREF,MAP,RENT'
//SYSPRINT DD SYSOUT=*
//SYSMOD    DD DSN=your.EZKEY.LOAD,DISP=SHR
//SYSLIB    DD DSN=your.EZKEY.LOAD,DISP=SHR
//OBJECT    DD DSN=your.PIESSTSA.OBJ,DISP=OLD
//SYSLIN    DD *
            ORDER EZKEYTCP
            ORDER PIEDMAIN
            INCLUDE OBJECT
            INCLUDE SYSLIB(EZKEYTSO)
            ENTRY EZKEYTCP
            NAME EZKEYTSO(R)
/*
//
```

The following messages are normal and expected when link-editing EZ/KEY for TSO:

```
IEWO241 AMPZTPUT
IEWO241 AMPZTGET
```

Customizing EZ/KEY TSO Protected Commands

For security or integrity reasons, you may not want EZ/KEY users to be able to issue certain TSO commands from within EZ/KEY. To allow you to specify which commands to inhibit, EZ/KEY contains a table of protected commands.

The assembler language source code for the PIESSTSP CSECT is contained in the member PIESSTSP. This member is in the SOURCE library copied from file 3 of the distribution tape. The list of commands is a sequence of 8-byte entries. Each entry contains a left-justified command name, padded with blanks. An entry of 8 spaces indicates the end of the list. Any TSO abbreviations must also be in the list. Extra blank entries are provided at the end of the list so that you can add command names using AMASPZAP. The commands in the default PIESSTSP have been provided to prevent accidental invocation of EZ/KEY from within EZ/KEY. Do not remove or change these entries. You should review the names in PIESSTSP to ensure that they are appropriate for your environment.

Optional TSO Installation Procedures

The PIESSTSP CSECT link-edited into EZ/KEY contains a list of the TSO commands that EZ/KEY does not allow to be executed. If the command is found in this list, the message **xxxxxxx NOT ALLOWED** is displayed, where **xxxxxxx** is the command the user entered. The commands contained in PIESSTSP are:

```

DC    CL8'EZKEY      '
DC    CL8'EZKEYTSO'
DC    CL8'EZKYLAM    '
DC    CL8'FGPAN23    '
DC    CL8'           '    ALL BLANKS INDICATES END OF TABLE
DC    CL8'           '
DC    CL8'           '    EXTRA SLOTS FOR INSTALLATION USE
DC    CL8'           '
DC    CL8'           '
DC    CL8'           '
DC    CL8'           '
DC    CL8'           '
DC    CL8'           '
DC    CL8'           '
DC    CL8'           '
DC    CL8'           '
DC    CL8'           '

```

The following illustrates the JCL to link-edit your customized PIESSTSP into EZ/KEY:

```

//jobname JOB (accounting.info),MSGLEVEL=1,
//          CLASS=A,MSGCLASS=A
//LKED      EXEC PGM=IEWL,PARM='LIST,LET,NCAL,XREF,MAP,RENT'
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=your.EZKEY.LOAD,DISP=SHR
//SYSLIB DD DSN=your.EZKEY.LOAD,DISP=SHR
//OBJECT DD DSN=your.PIESSTSP.OBJ,DISP=OLD
//SYSLIN DD *
          ORDER EZKEYTCP
          ORDER PIEDMAIN
          INCLUDE OBJECT
          INCLUDE SYSLIB(EZKEYTSO)
          ENTRY EZKEYTCP
          NAME EZKEYTSO(R)
/*
//

```

Optional TSO Installation Procedures

The following messages are normal and expected when link-editing EZ/KEY for TSO:

```
IEWO241 AMPZTPUT  
IEWO241 AMPZTGET
```

Installation in the MVS Link Pack Area

After installing EZ/KEY as a TSO Command Processor, you can put the EZKEYTSO load module in the MVS Link Pack Area (LPA). This reduces each TSO user's virtual storage requirements by approximately 1024K, but also reduces the available private area of all address spaces by an equivalent amount.

To install EZ/KEY in the LPA:

1. Copy the EZKEYTSO module to SYS1.LPALIB or to another library specified in the LPA concatenation.
2. IPL MVS with a CLPA.

Refer to the IBM *MVS SPL: Initialization and Tuning Guide* (GC28-1029) for more information.

Installation in an MVS/XA Environment

EZ/KEY is designed to run in 24-bit addressing mode and to reside in a 24-bit address space. If you put EZKEYTSO in the MVS/XA Link Pack Area, ensure that it is link-edited with RMODE=24 so that they will reside below the 16-megabyte line.

Refer to the following IBM publications for more information about operation of MVS/370 programs in an MVS/XA environment:

- *MVS/XA Conversion Notebook* (GC28-1143)
- *MVS/XA SPL: Initialization and Tuning Guide* (GC28-1149)
- *MVS/XA SPL: 31-Bit Addressing* (GC28-1158)

VALIDATE TSO INSTALLATION

To verify the correct installation of EZ/KEY, type EZKEY and press ENTER. You can suppress the extraneous IEC031I and IEC161I messages by issuing the TSO command **PROFILE NOWTPMSG** before invoking EZ/KEY. A panel should be displayed, with the name **PROSTART** in the upper left-corner, as illustrated below:

```
PROSTART ----- Primary Selection Menu ----- EZ/KEY
COMMAND ==>                                     USERID: userid
                                                TIME  : 10:31:51
                                                DATE   : 05/12/88

Select Option ==>

          0  Program Function Key Settings
          1  Edit an EASYTRIEVE PLUS Program
          2  Edit any Member
          3  Library and System Utilities
          4  Run a Program On-Line
          5  Submit a Program for Batch Processing
          6  Report Processing Facility

          T  Tutorial for EZ/KEY

          X  Exit the EZ/KEY System

EZ/KEY, Version X.X
Copyright (c) 1983, 1988 Pansophic Systems, Inc.
```

After validating the EZ/KEY installation, review Section 4 of the *EZ/KEY Administrator's Guide*, especially the subsection that covers establishing TSO user PIELIBs and libraries. If you want to set up user PIELIBs, you should do so at this time.

USING VSAM PIELIBS WITH EZ/KEY

EZ/KEY can use VSAM as an alternative to BDAM for user PIELIBs. If you want all EZ/KEY userid.PIELIBs to be VSAM, do the following:

1. Sign on to EZ/KEY to create a BDAM PIELIB for yourself. Use your userid to create all of the VSAM userid.PIELIBs.
2. Set &EZKAUTO to NO in the EZKEY.CLIST. This prevents other users from signing onto EZ/KEY and creating their own BDAM userid.PIELIBs.
3. Use IDCAMS to allocate each VSAM userid.PIELIB. For example, assume you select a blocksize of 2041 and an initial allocation of 200 blocks for each EZ/KEY user:

```
DEFINE CLUSTER (NAME(userid.PIELIB) VOLUMES(volser) RECORDS(200 20) -  
                CISZ(2048) SHR(2 3) RECORDSIZE(2041 2041) NUMBERED)
```

4. Allocate the newly created userid.PIELIB to your TSO session:

```
ALLOCATE FILE(userid) DATASET('userid.PIELIB') SHR
```

5. Use EZ/KEY utility option 3.a.3 to FORMAT the new PIELIB. Specify the userid as the LIBSPACE Name, PIEVSAM as the LIBSPACE Type, and CREATE for Access Rights. Erase the External Name field. Specify YES for Format. Press ENTER. For the Number of Blocks, specify the same number that you specified for the primary extent on the RECORDS parameter of the DEFINE CLUSTER (in the example above, 200). For the Size of Each Block, specify the same number that you specified in the RECORDSIZE parameter of the DEFINE CLUSTER (in the example above, 2041). Press ENTER. You should receive the message **Format OK** in the upper right corner of the screen. Press END (PF3).

Repeat steps 3 through 5 for each PIELIB to be created. Refer to Section 4 of the *EZ/KEY Administrator's Guide* for more detailed instructions for setting up userid.PIELIBs.

EASYTRIEVE PLUS VSAM MACRO LIBRARIES

EZ/KEY can read members from one or more EASYTRIEVE PLUS VSAM Macro libraries. For each library, an ALLOC statement similar to:

```
ALLOC FI(PANDD) DA('your.eztmacro.library') SHR
```

must be in the EZKEY.CLIST, or a DD statement similar to:

```
//PANDD DD DISP=SHR,DSN=your.eztmacro.library
```

must be in the user's logon PROC.

As explained in the *EASYTRIEVE PLUS Installation Guide*, to use VSAM as the EASYTRIEVE PLUS macro library, specify **MACRO=VSAM** as an operand of the EZTPOPT options macro.

Each library must be CONNECTed and ENABLEd to EZ/KEY. The System Administrator must use the External LIBSPACE Utility (option 3.a.4) to CONNECT each EASYTRIEVE PLUS Macro library to EZ/KEY. In the example above, you would use PANDD as the External Name.

PANVALET CONSIDERATIONS FOR TSO

EZ/KEY can read members from one or more PANVALET Release 11.0 or above libraries. Each library must be identified in the EZKEY CLIST (or LOGON PROC) and must be CONNECTed and ENABLEd to EZ/KEY.

For each library, a statement similar to:

```
ALLOC F(PANDD1) DA('your.PANVALET.library') SHR
```

must be in the EZKEY CLIST, or a DD statement similar to:

```
//PANDD1 DD DISP=SHR,DSN=your.PANVALET.library
```

must be in each EZ/KEY TSO user's LOGON PROC.

As explained in the *EASYTRIEVE PLUS Installation Guide*, to use PANVALET as the EASYTRIEVE PLUS macro library, specify MACRO=PAN as an operand of the EZTPOPT options macro. PANDD, the default value of the MACDDN option of EZTPOPT, is the prefix of DDnames for EASYTRIEVE PLUS' macro library. Multiple PANVALET libraries can be used in EASYTRIEVE PLUS; the DDnames for the PANVALET libraries must be PANDD1, PANDD2, ..., PANDDn, where *n* is the value of MAC#LIB on the EZTPOPT macro. PANDD1 should be used as the DDname and External LIBSPACE Name for the PANVALET library to comply with the naming conventions of EASYTRIEVE PLUS.

The System Administrator must use the External LIBSPACE Utility (option 3.a.4) to CONNECT each PANVALET library to EZ/KEY.

When the first PANVALET library is CONNECTed to EZ/KEY, a LOAD is issued for the module EZKYLAM, the access method for PANVALET (this module is distributed and installed as part of the EZ/KEY system for TSO). EZKYLAM issues a LOAD for FGPN23, the options module for PANVALET. FGPN23 is a component of your PANVALET system; it is produced by assembling and link-editing the PVOPT macro.

PANVALET Considerations for TSO

FGPAN23 must be located in *one* of the following load module libraries:

- The same library specified in the CALL statement in the EZKEY CLIST if &EZKMODE = CALL

Or

- A STEPLIB in the users' LOGON PROC

Or

- A LINKLIST library

PANVALET Security for TSO

There are no special EZ/KEY considerations when you do not use any of the PANVALET security features.

If the PANVALET library is protected with a library control code, and the ++PRINT or ++WRITE command is suppressed for this library, you must specify this control code as the External Password. The control code should be the same as specified on the ++CONTROL statement in batch PANVALET.

If a PANVALET member that has an access code is invoked as a macro in an EASYTRIEVE PLUS program, you must precede the %macro-name statement by an ACCESS 'access-code' statement in the program.

To edit an access-protected member, use the Copy/Move Utility (option 3.3) to copy the member from the PANVALET library to your PIELIB. Specify the member access code as the Password of the FROM: member.

During an edit session, you can retrieve an access-protected PANVALET member by entering *COPY* (with no operands) on the COMMAND line. This invokes the COPY/MOVE Utility panel on which you can specify the member to be retrieved and its access code.

PANVALET/TSO Memory Usage

Approximately 100K of additional memory is used when the first PANVALET library is CONNECTed; each additional PANVALET library requires approximately 10K:

EZKYLAM module	85K
FGPAN23 module	1K
Each CONNECTed Library	10K

TSO PANVALET Version 14.0 or Above Considerations

If you are using PANVALET Version 14.0 or above, use the System Tuning Parameter panel (PROULTPS) to inform EZ/KEY to use the PANVALET Interface module, PANMODI, to access PANVALET libraries instead of EZKYLAM.

Type **3.a.7** on the PROSTART panel to obtain the System Tuning Parameter panel (PROULTPS). On PROULTPS, type PANMODI in the PANVALET Module Name field.

So that EZ/KEY can locate the PANMODI module, it must be in a library that is available to EZ/KEY. To make PANMODI available, you can do *one* of the following:

- Add the PANVALET load library as a STEPLIB in each EZ/KEY user's LOGON PROC

Or

- Include the name of the PANVALET load library in the LNKSTxx member of SYS1.PARMLIB

TSO/TCAM CONSIDERATIONS

This topic applies only to TSO/TCAM sites. Skip this topic if IBM's SPF or ISPF program products are installed in your system or if you do not plan to run EZ/KEY under TSO/TCAM. The following information is derived from the IBM publication *ISPF/MVS Installation and Customization* (SC34-2084).

TSO/TCAM Message Handler

To run EZ/KEY under TSO/TCAM, you should make the following minor modifications to the standard IBM TSO/TCAM message handler for TSO. To install these changes:

1. Modify Stage 1
2. Assemble Stage 1 to produce the Stage 2 source
3. Modify the resulting Stage 2 source
4. Assemble Stage 2
5. Link edit the output of Stage 2

You should also review the TSO/TCAM (TIOC) buffer control parameters to ensure that adequate buffering is available for EZ/KEY.

Stage 1 Modifications

If the FULLSCR=YES operand was not originally specified during MCP generation, add FULLSCR=YES to the LINEGRP macro and to any LISTTA macros which describe terminals on which EZ/KEY will be used. Reassemble the Stage 1 source to produce the Stage 2 source.

Stage 2 Modifications

Add EXPFLS=YES as an operand on the FULLSCR macro in the input section of the Stage 2 source. If TRIGGER= is an operand of the FULLSCR macro, then instead of adding the EXPFLS=YES operand, insert the following statement immediately after the existing FULLSCR macro:

```
FULLSCR EXPFLS=YES
```

In the output section of the message handler, insert the following statement between the OUTBUF macro and the CODE macro:

```
OUTBUF FULLSCR EXPFLS=YES CODE
```


Additional Stage 2 Considerations

Before assembling your newly created Stage 2 MCP, review the values specified for BUFSIZE, CUTOFF, and LNUNITS.

BUFSIZE

BUFSIZE must be at least 2100 on both the TERMINAL and DCB macros for each terminal on which EZ/KEY will be used. BUFSIZE should also be an even multiple of the KEYLEN operand value specified on the INTRO macro.

CUTOFF

The value for the CUTOFF macro must be at least as large as indicated in the table below:

Screen Size	3270 Model	CUTOFF Value
-----	-----	-----
1920	2	2048
2560	3	3072
3440	4	3072
3562	5	3072

Screen sizes larger than 1920 may be used by ACF/TCAM, but not by TCAM 10.

LNUNITS

LNUNITS on the INTRO macro must be at least as large as the sum of (BUFSIZE/KEYLEN) for all TCAM terminals (including non-EZ/KEY terminals). For example, if your installation has:

- 8 terminals with BUFSIZE=2100
- 3 terminals with BUFSIZE=400
- KEYLEN=100

then set LNUNITS = $(8 * 2100/100) + (3 * 400/100) = 180$

TSO/VTAM CONSIDERATIONS

This topic applies only to TSO/VTAM sites. Skip this topic if you do not plan to run EZ/KEY under TSO/VTAM. The following information is derived from the IBM publication *ISPF/MVS Installation and Customization* (SC34-2084).

SCRSIZE=1920 must be specified on the TSOKEY00 member in PARMLIB. (Specify a screen size of 1920 even if you have larger screens.)

TSO/NCP/VTAM sites should ensure that the NCP and VTAM parameters are sufficient for running EZ/KEY. The following values should be adequate for 1920-byte screens.

NCP HOST macro:

UNITSZ = 384

VTAM Parameters:

IOBUF = 384

PPBUF = 384 (applies only to VTAM 2)

BUFLIM = 7 (applies only to VTAM 2)

BUFACT = 1 (applies only to VTAM 2)

Screen sizes larger than 1920 bytes are supported by ACF/VTAM, but are not supported by VTAM 2.

INSTALLING EZ/KEY UNDER ISPF/PDF

You can install EZ/KEY under ISPF to appear as an option on the ISPF Primary Option Menu, ISR@PRIM. To do this, make the following changes:

1. Modify the ISR@PRIM panel to display the EZ/KEY choice.
2. In the PROC section of the ISPF panel, invoke the EZ/KEY CLIST.

The following example shows EZ/KEY installed as option E:

```
%----- ISPF/PDF PRIMARY OPTION MENU -----
%OPTION ==> _ZCMD
%
%      1  +BROWSE          ...
%      2  +EDIT            ...
%      3  +UTILITIES       ...
%
%      E  +EZ/KEY          Productivity Aid for EASYTRIEVE PLUS
%
%
%INIT
...
)PROC
  &ZSEL = TRANS( TRUNC(&ZCMD, '.')
    1, 'PGM(ISRBRO)'
    2, 'PGM(ISREDIT)'
    3, 'PANEL(ISRUTIL)'
    .
    .
    .
    E, 'CMD(EXEC ''prefix.EZKEY.CLIST'') NEWPOOL'
    .
    .
    .
)END
```

Exhibit 3.1: Sample ISR@PRIM Panel

Due to the combined storage requirements of ISPF and EZ/KEY, increase the size of your region by at least 1.5 megabytes.

OPTIONAL SQL INSTALLATION PROCEDURES***EZ/KEY SQL Interface***

When you install the SQL option for TSO EZ/KEY, your users can use the PANSOPHIC SQL Interface to DB2. This allows them to syntax-check SQL commands embedded in their EASYTRIEVE PLUS programs.

Requirements

The SQL Option for TSO EZ/KEY operates on all MVS systems that support IBM's DB2. If you do not have DB2 installed, then you cannot use the EZ/KEY SQL support.

Disk Space Requirements

The EZ/KEY SQL Interface should be installed in the same load module library where EZ/KEY is installed, specified in the JCL as **your.EZKEY.LOAD**.

An additional 250K bytes is required in your.EZKEY.LOAD library for permanent residence of the SQL Interface modules.

Virtual Memory Requirements

Approximately 250K of additional memory is used to load all of the SQL Interface modules into virtual storage when the first SQL statement is encountered. Each additional SQL statement requires approximately 100 bytes of virtual storage for work areas.

Due to the additional storage required, you may need to increase the default LOGON SIZE by at least 512K for TSO users who use EZ/KEY.

SQL Interface Option Installation

Running the job to install the EZ/KEY SQL Interface transfers the EZ/KEY SQL Interface modules from the installation tape to your EZ/KEY load module library.

Loading SQL Installation JCL

File 6 on the installation tape contains the JCL you need to install the SQL Interface. The following JCL copies file 6 from the installation tape to a sequential data set. This JCL must be modified as described below:

```
//jobname1 JOB accounting.info          ***-1-***
//DB2JCL   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=FILE06,UNIT=TAPE,DISP=OLD,
//          LABEL=(6,SL),VOL=SER=volser          ***-2-***
//SYSUT2   DD DSN=user.id.EZKEYDB2.JCL,          ***-3-***
//          DISP=(,CATLG,DELETE),
//          DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//          UNIT=SYSDA,VOL=volser,              ***-4-***
//          SPACE=(3120,(100,25),RLSE)
```

JCL Notes

At the right of some JCL lines shown above is the character string ***-*n****, where *n* is a number corresponding to one of the notes below. Modify the JCL as described in the notes:

1. Modify the JOB name and accounting information as needed.
2. Specify the volser of the EZ/KEY Installation Tape.
3. Specify the dataset name to use for the sequential dataset to contain the EZ/KEY SQL installation JCL, or modify this statement to place the JCL into a member of a PDS of your choice.
4. Specify the volser of the disk to contain the above dataset.

SQL Installation JCL (File 6)

The JCL contained in file 6 on the installation tape is shown below. Use it to install the SQL Interface for EZ/KEY. First, review the JCL. Then modify it as described in the notes that follow:

```
//jobname1 JOB accounting.info,USER=userid          ***-1-***
//*****
//**          INSTALL EZ/KEY SQL INTERFACE          **
//*****
//EZKEYSQL PROC DBRMLIB='your.DB2.DBRM.library',      ***-2-***
//              DCLGLIB='your.DB2.DCLGN.library',      ***-3-***
//              DB2LOD1='your.IBM.DB2.loadlib1',        ***-4-***
//              DB2LOD2='your.IBM.DB2.loadlib2',        ***-5-***
//              LOADLIB='your.EZKEY.LOAD',              ***-6-***
//              MEMNAME=DQEKCMD                        ***-7-***
//*****
//**          BUILD OBJECT LIBRARY OF SQL INTERFACE MODULES          **
//*****
//BUILD OBJ EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD  SYSOUT=*
//SYSUT1  DD  DUMMY
//SYSUT2  DD  DSN=&&OBJLIB,DISP=(,PASS),
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//              UNIT=SYSDA,SPACE=(CYL,(3,1,10))
//*****
//**          ASSEMBLE COMMAND PROCESSOR MACRO          **
//*****
//ASMMACR EXEC PGM=IEV9C,REGION=512K,PARM='DECK,RENT,LIST'
//SYSUT1  DD  DSN=&&SYSUT1,UNIT=(SYSDA),SPACE=(CYL,(10,5))
//SYSLIN  DD  DSN=&&LOADSET,DISP=(NEW,PASS),
//              UNIT=SYSDA,SPACE=(TRK,(10,5)),
//              DCB=BLKSIZE=3120
//SYSPUNCH DD  DSN=&&ASM,DISP=(NEW,PASS),
//              UNIT=SYSDA,SPACE=(CYL,(5,5,0)),
//              DCB=BLKSIZE=800
//SYSPRINT DD  SYSOUT=*,DCB=BLKSIZE=3509,
//              UNIT=(,SEP=(SYSUT1,SYSPUNCH))
```

continued ...

... continued

```

//*****
//**      DB2 PRECOMPILER      **
//*****
//DB2COMP EXEC PGM=DSNHPC,PARM='HOST(ASM),XREF,SOURCE'
//STEPLIB DD DSN=&DB2L0D1,DISP=SHR
//DBRMLIB DD DSN=&DBRMLIB(&MEMNAME),DISP=SHR
//SYSCIN DD DSN=&&DB2OUT,DISP=(MOD,PASS),
//          UNIT=SYSDA,SPACE=(TRK,(3,3)),
//          DCB=BLKSIZE=800
//SYSLIB DD DSN=&DCLGLIB,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(800,(1000,1000))
//SYSUT2 DD UNIT=SYSDA,SPACE=(800,(1000,1000))
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSIN DD DSN=&&ASM,DISP=(OLD,DELETE)
//*****
//**      ASSEMBLE COMMAND PROCESSOR      **
//*****
//ASMCMDP EXEC PGM=IEV90,REGION=512K,PARM='DECK,LIST,RENT'
//SYSUT1 DD DSN=&&SYSUT1,
//          UNIT=(SYSDA,SEP=SYSLIB),
//          SPACE=(CYL,(10,5))
//SYSLIN DD DSN=&&LOADSET,DISP=(MOD,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(10,5)),
//          DCB=BLKSIZE=3120
//SYSPUNCH DD DSN=&&OBJLIB(OMSMCMD2),DISP=(OLD,PASS)
//SYSPRINT DD SYSOUT=*,DCB=(BLKSIZE=3509),
//          UNIT=(,SEP=(SYSUT1,SYSPUNCH))
//SYSIN DD DSN=&&DB2OUT,UNIT=SYSDA,DISP=(OLD,DELETE)
//*****
//**      LINK-EDIT SQL INTERFACE      **
//*****
//LINKEDIT EXEC PGM=IEWL,PARM='CALL,LET,LIST,MAP,RENT'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIB DD DSN=&&OBJLIB,DISP=(OLD,DELETE,DELETE)
//DB2LIB DD DSN=&DB2L0D1,DISP=SHR
//SYSLMOD DD DSN=&LOADLIB,DISP=OLD

```

continued ...

... continued

```

//*****
//**      INVOKE TSO DSN COMMAND TO DO BIND      **
//*****
//DB2BIND EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=&DB2LOD1,DISP=SHR
//      DD DSN=&DB2LOD2,DISP=SHR
//DBRMLIB DD DISP=SHR,DSN=&DBRMLIB
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//REPORT DD SYSOUT=*
//*
//      PEND
//INSTSQL EXEC EZKEYSQL
//BUILDOBJ.SYSIN DD *
./ ADD NAME=.....
ESD ...
...
/*
//ASMMACR.SYSIN DD *
PRINT OFF
...
      DQSMCMD MAXSTMT=ss,MAXCUR=cc,PLAN=DQSMCMD      ***-7-***
      END
/*
//LINKEDIT.SYSLIN DD *
INCLUDE ...
...
/*
//DB2BIND.SYSTSIN DD *
      DSN SYSTEM(ssid)
      BIND PLAN(DQEKCMD) MEMBER(DQEKCMD) -      ***-8-***
      ACT(REPLACE) ISOLATION(CS)      ***-7-***
      RUN PROGRAM(DSNTIAD) PLAN(DSNTIAD2) -
      LIB('your.IBM.DB2.loadlib2')      ***-5-***
      END
/*
//DB2BIND.SYSIN DD *
      GRANT EXECUTE ON PLAN DQEKCMD TO PUBLIC;      ***-7-***
/*
//

```


Tailoring SQL Installation JCL

At the right of some of the JCL lines shown above is the character string `***-n-***`, where *n* is a number corresponding to one of the notes below. Modify the JCL as described in the notes:

1. Modify the JOB name and accounting information as needed. Change the USER=userid parameter to specify a valid DB2 userid with DBA authority.
2. Change the DBRMLIB= parameter to identify your IBM DB2 DBRM library.
3. Change the DCLGLIB= parameter to identify your IBM DB2 DCLGN library that contains the definition of the SQLCA.
4. Change the DB2LOD1= parameter to identify the IBM DB2 load library where the DB2 module DSNTIAR resides.
5. Change the DB2LOD2= parameter and the LIB(...) parameter to identify the IBM DB2 load library where the DB2 module DSNTIAD resides. Note that at your site this may be the same library referred to in Note #4 above.
6. Change the LOADLIB= parameter to identify the EZ/KEY load library where the SQL Interface is to be installed.
7. Change the MAXSTMT= and MAXCUR= parameters to the same values specified in your EASYTRIEVE PLUS SQL installation job. Change the PLAN=DQSMCMD parameter to PLAN=DQEKCMD, or any other name, as long as the name you choose is different than the name specified in your EASYTRIEVE PLUS SQL installation job. The name recommended for EZ/KEY is DQEKCMD.
8. Change the SSID parameter to specify the DB2 subsystem that EZ/KEY operates with. This should be the same as the one specified in your EASYTRIEVE PLUS SQL installation job.

OPTIONAL IDMS/IDD INSTALLATION***IDMS/IDD Installation JCL***

To use the IDD statement from within EZ/KEY, you must run the following JCL to create the EZKYIDMS module EZ/KEY needs to access the IDMS IDD Interface:

```
//jobname1 JOB (accounting info),REGION=512K          ***-1-***
//LKED      EXEC  PGM=IEWL,REGION=512K,
//          PARM='LIST,LET,MAP,NCAL,RENT'
//SYSUT1    DD   UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSLIB    DD   DISP=SHR,DSN=your.IDMS.RO10.0BULIB    ***-2-***
//SYSLMOD   DD   DISP=SHR,DSN=your.EZKEY.LOAD         ***-3-***
//SYSPRINT  DD   SYSOUT=*
//SYSLIN    DD   *
              INCLUDE SYSLIB(IDMS)
              INCLUDE SYSLIB(IDMSINTB)
              INCLUDE SYSLIB(IDMSCANC)
              ENTRY IDMS
              NAME EZKYIDMS(R)
/*
//
```

Tailoring IDMS/IDD JCL

A copy of the JCL shown above can be found in EXAMPLE.PIELIB in the member named LKIDDTSO JCL.

At the right of some of the JCL lines shown above is the character string ***-*n****, where *n* is a number corresponding to one of the notes below. Modify the JCL as described in the notes:

1. Modify the JOB statement as required for your installation.
2. Specify the name of the dataset in which the IDMS object modules IDMS, IDMSINTB, and IDMSCANC reside.
3. Specify the dataset name of your EZ/KEY load library. This job creates a new module, EZKYIDMS, and adds it to this library. EZ/KEY uses EZKYIDMS to access the IDMS/IDD interface.

After tailoring the JCL above, submit it for execution under MVS. This job should run with no errors and receive a return code of 0.

ADDITIONAL IDMS CONSIDERATIONS***Central Version Mode Operation***

If EZ/KEY is to access the IDD through Central Version, your site must have IDMS-CV (Central Version) installed on the MVS system where EZ/KEY is running.

To permit EZ/KEY users to access the IDD using Central Mode, add the following ALLOCATE statement to the EZKEY.CLIST:

```
ALLOCATE FI(SYSCTL) DA('your.IDMS.control.dataset') SHR
```

For more detailed information on operating IDMS/IDD in MVS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

Local Mode Operation

Local Mode operation permits EZ/KEY users to access the IDMS/IDD without requiring Central Version on the MVS system where EZ/KEY runs. However, this also requires significantly more virtual storage for all of the IDMS modules that are loaded into the user's virtual storage. Therefore, you should increase the SIZE parameter in the UADS for each user or change the REGION= parameter in each user's LOGON PROC accordingly.

To permit EZ/KEY users to access the IDD using Local Mode, complete the following steps.

1. Concatenate the appropriate Cullinet load module libraries to STEPLIB in each TSO EZ/KEY user's LOGON PROC; this provides access to all of the IDMS modules needed to execute IDMS in Local Mode.

Additional IDMS Considerations

2. Add the following ALLOCATE statements to the EZKEY.CLIST (or add the equivalent DD statements to the EZ/KEY user's LOGON PROC):

```
ALLOCATE FI(SYSJRN1) DA('your.IDMS.journal.dataset') OLD  
ALLOCATE FI(dbase1) DA('your.dbase1.dataset.name') OLD
```

```
ALLOCATE FI(dbaseN) DA('your.dbaseN.dataset.name') OLD
```

where *dbase1* is the DDNAME, and *your.dbase1.dataset.name* is the dataset name of the IDMS database that contains an IDD dictionary. Consult your IDMS Data Base Administrator for assistance. Refer to the *EASYTRIEVE PLUS Installation Guide* for more information.

For more detailed information on operating IDMS/IDD in MVS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

IDMS Foreground Execution Considerations***Central Version Mode Foreground Operation***

If EASYTRIEVE PLUS is to access the IDD through Central Version, your site must have IDMS-CV (Central Version) installed on the VM/CMS system where EZ/KEY is running.

To permit EZ/KEY users to access the IDD and IDMS databases from EASYTRIEVE PLUS running under TSO using Central Mode, add the following ALLOCATE statement to the EZTPLUS.CLIST used for Foreground Execution of EASYTRIEVE PLUS under EZ/KEY:

```
ALLOCATE FI(SYSCTL) DA('your.IDMS.control.dataset') SHR
```

Consult your IDMS Data Base Administrator for assistance.

For more detailed information on operating IDMS/IDD in MVS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

Local Mode Foreground Operation

Local Mode Operation permits EZ/KEY users to access the IDMS/IDD without requiring Central Version on the MVS system where EZ/KEY runs. However, this also requires significantly more virtual storage for all of the IDMS modules that are loaded into the user's virtual storage. Therefore, you should either increase the SIZE parameter in the UADS for each user or change the REGION= parameter in each user's LOGON PROC accordingly.

To permit EZ/KEY users to access the IDD and IDMS databases from EASYTRIEVE PLUS running under TSO using Local Mode, complete the following steps:

1. Concatenate the appropriate Cullinet load module libraries to STEPLIB in each TSO EZ/KEY user's LOGON PROC; this provides access to all of the IDMS modules needed to execute IDMS in Local Mode.
2. Add the following ALLOCATE statements to the EZTPLUS.CLIST:

```
ALLOCATE FI(SYSJRN) DA('your.IDMS.journal.dataset') OLD
ALLOCATE FI(dbase1) DA('your.dbase1.dataset.name' ) OLD

ALLOCATE FI(dbaseN) DA('your.dbaseN.dataset.name' ) OLD
```

where *dbase1* is the DDNAME, and *your.dbase1.dataset.name* is the dataset name of an IDMS database. Consult your IDMS Data Base Administrator for assistance.

For more detailed information on operating IDMS/IDD in MVS, refer to Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

IDMS Background Execution Considerations***Central Version Mode Background Operation***

If EASYTRIEVE PLUS is to access the IDD through Central Version, your site must have IDMS-CV (Central Version) installed on the MVS system where batch EASYTRIEVE PLUS runs.

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Central Mode, add the DD statement shown below to the EZTPLUS Header JCL for Background Execution in EZ/KEY.

Identify the IDMS control file for use by EASYTRIEVE PLUS programs:

```
//SYSCTL DD DSN=your.idms.control.file,DISP=SHR
```

where *your.idms.control.file* is the name of the control file for IDMS. Consult your IDMS Data Base Administrator for assistance. Refer to the *EASYTRIEVE PLUS Installation Manual* for more information.

For more detailed information on operating IDMS/IDD in MVS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

Local Mode Background Operation

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Local Mode, add the DD statements shown below to the EZTPLUS Header JCL for Background Execution in EZ/KEY:

1. Identify the load libraries where the Cullinet IDMS modules reside; this provides access to the modules needed to execute IDMS in Local Mode:

```
//STEPLIB DD DSN=your.EZTP.load.library,DISP=SHR  
//          DD DSN=your.IDMS.load.library,DISP=SHR
```

where *your.IDMS.load.library* is the name of the load module library for IDMS.

Additional IDMS Considerations

2. Identify the IDMS journal file for use by EASYTRIEVE PLUS programs:

```
//SYSJRN DD DSN=your.idms.journal.file,DISP=(,KEEP),UNIT=TAPE
```

where *your.idms.journal.file* is the name of a journal file for IDMS. Note that you can allocate this data set to disk or specify a DUMMY data set.

3. Identify all of the databases accessed by EASYTRIEVE PLUS programs:

```
//dbfile1 DD DSN=your.idms.database.file1,DISP=SHR
```

```
//dbfileN DD DSN=your.idms.database.fileN,DISP=SHR
```

where *dbfileN* and *your.idms.database.fileN* are the DDname and dataset name, respectively, for each database file that may be accessed by your EASYTRIEVE PLUS programs.

There must be a DD statement in the EZTPLUS JCL for each database that is accessed by any EASYTRIEVE PLUS program running in Local Mode. Consult your IDMS Data Base Administrator for assistance. Refer to the *EASYTRIEVE PLUS Installation Manual* for more information.

For more detailed information on operating IDMS/IDD in MVS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

IDMS/IDD Documentation

For more information and details on installing and running IDMS/IDD under TSO, refer to the following Cullinet publications:

- *Cullinet System Software, Integrated Installation Guide*
- *Cullinet System Software, System Generation*
- *IDMS-DC/CV System Operations*

(This page intentionally left blank.)

SECTION 4 CICS/OS/VS INSTALLATION

Introduction	4-1
Storage Estimates	4-1
Reinstalling EZ/KEY under CICS/OS/VS	4-2
Reconnect Procedures	4-2
Reinstallation Procedures	4-2
Installation Files	4-4
CICS Release Dependencies	4-4
CICS 1.5 Considerations	4-5
CICSLINK JCL	4-5
Tailoring CICSLINK JCL	4-6
Disk Space Requirements	4-6
Memory Requirements	4-6
Auxilliary Temporary Storage Requirements	4-7
Loading Installation JCL	4-8
JCL Notes	4-8
Installation JCL	4-9
Tailoring CICS/OS/VS Installation JCL	4-11
CICS/OS/VS Installation JCL Notes	4-11
Tailoring CICS/OS/VS Start-Up JCL	4-12
Choosing the Size of PIELIBs	4-14
Number of Blocks	4-14
Size of Each Block	4-14
CICS/OS/VS Table Changes	4-15
CICS 1.7 Considerations	4-17
Alternate Screen Size	4-17
CICS Dynamic Transaction Backout	4-18
CICS Multi-Region Option (MRO)	4-18
CICS Runaway Task Interval	4-19
CICS/OS/VS BDAM PIELIBS	4-20
Installation in an MVS/XA Environment	4-21
Validate CICS Installation	4-22
EASYTRIEVE PLUS VSAM Macro Libraries for CICS/OS/VS	4-23
PDS Macro Libraries for CICS/OS/VS	4-24

CICS/OS/VS Submit to Background	4-25
PANVALET Considerations for CICS/OS/VS	4-28
PANVALET Security for CICS/OS/VS	4-29
PANVALET/CICS Memory Requirements	4-29
CICS/OS/VS PANVALET Version 14.0 or Above Con- siderations	4-29
Multiple EZ/KEY Systems Under One CICS	4-31
Optional SQL Installation Procedures	4-33
EZ/KEY SQL Interface	4-33
SQL Interface Requirements	4-33
EZ/KEY SQL Interface Disk Space Requirements	4-33
EZ/KEY SQL Interface Virtual Memory Requirements ..	4-34
SQL Interface Option Installation	4-34
Loading SQL Installation JCL	4-34
JCL Notes	4-35
SQL Installation JCL (File 7)	4-35
Notes for SQL Interface Installation JCL	4-38
Additional CICS Considerations	4-39
Additional DB2 Considerations	4-40
DB2 Userids	4-42
Optional IDMS/IDD Installation	4-43
IDMS/IDD Installation JCL	4-43
Tailoring IDMS/IDD JCL	4-44
Additional IDMS/IDD Considerations	4-45
IDMS/IDD Interface Storage Usage	4-48
Additional IDMS/IDD CICS Considerations	4-49
IDMSINTC Generation	4-49
IDMSCINT Generation	4-50
IDMS Background Execution Considerations	4-51
Central Version Mode Background Operation	4-51
Local Mode Background Operation	4-51

INTRODUCTION

This section contains detailed procedures for installing EZ/KEY under CICS/OS/VS. You should also review the Administrator's Guide to establish how you intend to operate EZ/KEY at your site.

EZ/KEY can execute as either a conversational or as a pseudo-conversational CICS transaction.

When EZ/KEY executes as a conversational transaction, CICS allocates memory for EZ/KEY when a user signs on. EZ/KEY acquires additional memory as needed, and releases memory when no longer needed. The memory acquired for the basic EZ/KEY functions is freed when the user signs off.

When EZ/KEY executes as a pseudo-conversational transaction, CICS allocates memory for EZ/KEY when a user presses ENTER. EZ/KEY frees memory when it displays the next panel for a user. Some memory is shared by all EZ/KEY users. This memory is freed when the last EZ/KEY user signs off.

Storage Estimates

EZ/editor is an in-storage editor, similar to XEDIT or ISPF/PDF. The size of the program or data being edited is limited by the availability of virtual storage.

These storage estimates are subject to user activities and installation options that may require additional virtual storage in order to use EZ/KEY. The following factors can affect virtual storage availability:

- The number of open files
- The number of CICS programs residing in virtual storage
- The number of CICS transactions (users) currently active
- The number of *conversational* programs currently running

See the section titled "Memory Requirements" for further details on specific memory requirements.

Memory requirements, disk space sizes, JCL, and CICS table entries presented in this Section are for planning purposes only. More precise information is provided in the files on the EZ/KEY distribution tape.

REINSTALLING EZ/KEY UNDER CICS/OS/VS

If you have already installed EZ/KEY under CICS, almost everything you need is already in place. For Version 3.2 of EZ/KEY you must:

- Add a new PCT entry for transaction EZKT
- Add a new PPT entry for program EZKYOPTN
- Remove the PPT entries for programs EZKYSCOM and EZKYUTIL because they are no longer required

Also, if you choose a transaction ID other than EZKE for EZ/KEY, the first three characters of the EZ/KEY (EZKE) transaction must match those of the removal transaction (EZKT).

Reconnect Procedures

In CICS environments, EZ/KEY maintains a SYSTEM CONNECT member in the SYSTEM PIELIB. This member contains the names of all LIBSPACES that are connected to the user community sharing a particular SYSTEM PIELIB.

This CONNECT member is lost when a new release of EZ/KEY is installed from the distribution tape. Have the EZ/KEY System Administrator make a backup copy of the SYSTEM CONNECT member before you install a new release of EZ/KEY. Making a backup copy is described in the *EZ/KEY Administrator's Guide*, in the section covering reconnect procedures.

Reinstallation Procedures

Once the EZ/KEY System Administrator makes a backup copy of the SYSTEM CONNECT member, proceed with the installation as follows:

Review the output produced by the installation jobstream when the last version of EZ/KEY was installed. If you cannot locate this information, review the appropriate material in the previous version of the *EZ/KEY Installation Guide*.

Review the new sample table entries for EZ/KEY and compare them with your current table entries. Note any differences. Make the necessary changes to your CICS tables. If you specified parameters other than those shown in the sample entries, ensure that they are the same as the IBM supplied defaults for those values.

Review the sections of this guide that pertain to the new features described above.

Reinstalling EZ/KEY under CICS/OS/VS

Review the new installation JCL. Make any changes needed for your installation's environment.

Once you have made the necessary changes, run the new installation procedure, and then verify the results as described in this section under "Validate CICS Installation."

INSTALLATION FILES

The EZ/KEY system for CICS/OS/VS is distributed on a standard labeled tape that contains seven files:

- File 1 (DSN=FILE01) - Contains the Job Control Language (JCL) in IEBCGENER format, to install EZ/KEY. This file is blocked 80/3120.
- File 2 (DSN=FILE02) - Contains the EZ/KEY modules in IEBCOPY format.
- File 3 (DSN=FILE03) - Contains the CICS/OS/VS table changes (DCT, FCT, PCT, PPT, RCT) and the EZ/REPORT User's Guide in IEBCOPY format. The original source library is blocked 80/3120.
- File 4 (DSN=FILE04) - Contains the SYSTEM PIELIB in IDCAMS REPRO format.
- File 5 (DSN=FILE05) - Contains the EXAMPLE PIELIB in IDCAMS REPRO format.
- File 6 (DSN=FILE06) - Contains the USER PIELIB in IDCAMS REPRO format.
- File 7 (DSN=FILE07) - Contains a copy of the JCL and object code to install the SQL Interface for EZ/KEY. This file is blocked 80/3120.

CICS Release Dependencies

EZ/KEY operates under CICS 1.6 and later releases.

EZ/KEY is written as a Command Level Assembler language program. To run EZ/KEY under CICS, you must specify **EXEC=YES** in the SIT (or allow it to default) or specify **EXEC=YES** in the SIT overrides in the CICS start-up deck.

See the following IBM publications for further information:

- *CICS/OS/VS Resource Definition (Macro)* (SC33-0149 or SC33-0237) SIT - System Initialization Table (DFHSIT)
- *CICS/OS/VS Installation and Operations Guide* (SC33-0071) System Initialization Override Parameters

CICS 1.5 Considerations

In order to run EZ/KEY under CICS 1.5 you must call PANSOPHC Customer Service to obtain a PTF to apply to EZ/KEY.

If you are running CICS 1.5, you must re-link EZKYZ00 and EZKYZ01 with a copy of DFHEAI from your CICS/OS/VS Release 1.5 system. The CICS LINK JCL to do this is shown below and is provided in the SOURCE library and in EXAMPLE PIELIB provided on the distribution tape.

CICSLINK JCL

```
//jobname JOB accounting.info ***-1-***
//EZKOLNK EXEC PGM=IEWL,REGION=512K,PARM='LET,LIST,MAP,XREF'
//SYSLIB DD DSN=your.IBM.CICS.LOAD,DISP=SHR ***-2-***
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=prefix.EZKEY.LOAD,DISP=SHR ***-3-***
//SYSUT1 UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSLIN DD *
ORDER DFHEAI
ORDER PISSCOO
INCLUDE SYSLIB(DFHEAI)
REPLACE DFHEAIO
INCLUDE SYSLMOD(EZKYZ00)
ENTRY DFHEAI
NAME EZKYZ00(R)
ORDER DFHEAI
ORDER PISSCO1
INCLUDE SYSLIB(DFHEAI)
REPLACE DFHEAIO
INCLUDE SYSLMOD(EZKYZ01)
ENTRY DFHEAI
NAME EZKYZ01(R)
/*
//
```

Tailoring CICS LINK JCL

At the right of some of the JCL lines shown above is the character string `***-n-***`, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB name and accounting information as needed.
2. Specify the dataset name of your IBM CICS load module library.
3. Specify the dataset name prefix for EZ/KEY datasets.

Disk Space Requirements

The approximate disk space required for EZ/KEY is:

Name	Disk Blocks	Blocksize or Control Interval		3350
		Size		Tracks
EZ/KEY load library	350	6144		70
SYSTEM PIELIB	1500	4089		400
USER PIELIB	1000	2041		250
RESTORE PIELIB	1500	4089		400
EXAMPLE PIELIB	400	2041		50

Memory Requirements

Increase the CICS Region Size by 768K for the EZ/KEY main module EZKYZ00 (256K), the program cache (448K), plus the panel cache (48K), and by an additional amount that depends on the number of users who will be using EZ/KEY at the same time and on whether or not EZ/KEY operates in Conversational or Pseudo-conversational mode (the default for CICS).

The program cache limit defaults to 448K and the panel cache limit defaults to 48K for CICS. See the section "Site Options" in the *EZ/KEY Administrator's Guide* for a description of how to change the size of the Program and Panel Caches. If you change the size of either of the cache limits, you may need to adjust your CICS Partition Size accordingly.

When EZ/KEY is executing in Conversational mode, the virtual storage required for each user is needed for the duration of the EZ/KEY session. See the section "Tuning Parameters" in the *EZ/KEY Administrator's*

Guide for a description of how to change the Execute Conversational parameter.

When EZ/KEY is executing in Pseudo-conversational mode, the virtual storage required for each user is needed only for a short duration. Before the next screen is displayed, the per-user data is swapped out to Auxilliary Temporary Storage. CICS dynamic area virtual storage is acquired when the user presses ENTER and is released just before the next screen is displayed.

Use the following formula to compute your user memory requirements:

48K per user (initial requirements to sign on)
plus 200 bytes times the number of lines being edited

EZ/KEY is an in-storage editor, similar to XEDIT or ISPF/PDF. The size of a program or of data being edited is limited by the availability of virtual storage. One hundred bytes of storage are required for each line a user is editing, plus approximately 100 bytes per line when EZ/Checker is being used.

For example, suppose you anticipate that the largest member to be edited by any user is 500 lines in length and that there will be a maximum of ten simultaneous users. To determine the total memory required, perform the following calculations:

$$\begin{array}{r}
 500 \times 200 = 200 \text{ K} \\
 + \quad 48 \text{ K} \\
 \hline
 148 \text{ K per user} \\
 \times \quad 10 \text{ users} \\
 \hline
 1480 \text{ K total additional storage required}
 \end{array}$$

When EZ/KEY is executing in Pseudo-conversational mode, the actual amount of storage needed at any one moment is usually much less than the total shown above. This depends, however, on the actual activities being performed by the users who are signed on and active within EZ/KEY at that time.

Auxilliary Temporary Storage Requirements

You may need to increase the size of Auxilliary Temporary Storage (disk space), depending on the maximum number of users using EZ/KEY at the same time.

Use the formula shown above to determine the additional Auxilliary Temporary Storage needed for EZ/KEY to operate pseudo-conversationally under CICS. For planning purposes, assume that all users may be swapped out to Auxilliary Temporary Storage at the same time.

Loading Installation JCL

File 1 on the installation tape contains the JCL you need to install EZ/KEY. Use the following JCL to copy file 1 from the installation tape to a sequential data set. Modify this JCL as described below:

```
//jobname1 JOB accounting.info ***-1-***
//INSTJCL EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DSN=FILE01,UNIT=TAPE,DISP=OLD,
// LABEL=(1,SL),VOL=SER=volser ***-2-***
//SYSUT2 DD DSN=user.id.EZKEY.JCL, ***-3-***
// DISP=(,CATLG,DELETE),
// DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
// UNIT=SYSDA,VOL=volser, ***-4-***
// SPACE=(3120,(100,25),RLSE)
```

JCL Notes

At the right of some of the JCL lines shown above is the character string *****-n-*****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB name and accounting information as needed.
2. Specify the volser of the EZ/KEY Installation Tape.
3. Specify the dataset name to use for the sequential dataset to contain the EZ/KEY Installation JCL, or modify this statement to place the JCL into a member of a PDS of your choice.
4. Specify the volser of the disk to contain the above dataset.

INSTALLATION JCL

The following JCL is on the first file of the EZ/KEY distribution tape. You must load this file onto your system and edit it. Items that you should change are identified by the character string *****-n-***** to the right:

```
//jobname JOB accounting.info ***-1-***
//*****
//* IEBCOPY EZKEY AND LAMS MODULES
//*****
//COPY EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD UNIT=SYSDA,SPACE=(80,(120,112))
//SYSUT4 DD UNIT=SYSDA,SPACE=(80,(15,7))
//FILE02 DD DSN=FILE02,DISP=(OLD,PASS),LABEL=2,
// UNIT=TAPE,VOL=(PRIVATE,RETAIN,SER=123456) ***-2-***
//FILE03 DD DSN=FILE03,DISP=(OLD,PASS),LABEL=3,
// UNIT=TAPE,VOL=REF=*.COPY.FILE02
//LOAD DD DSN=prefix.EZKEY.LOAD,DISP=(,CATLG), ***-3-***
// UNIT=SYSDA,VOL=SER=volser, ***-4-***
// SPACE=(6144,(350,50,10),RLSE)
//SOURCE DD DSN=prefix.EZKEY.SOURCE,DISP=(,CATLG), ***-3-***
// UNIT=SYSDA,VOL=SER=volser, ***-4-***
// SPACE=(6144,(350,50,10),RLSE)
//SYSIN DD *
COPY INDD=((FILE02,R)),OUTDD=LOAD
COPY INDD=((FILE03,R)),OUTDD=SOURCE
/*
//*****
//* DEFINE SYSTEM, EXAMPLE, USER AND RESTORE PIELIBS
//*****
//IDCAMS EXEC PGM=IDCAMS
//STEP1 DD DSN=your.user.catalog,DISP=SHR ***-5-***
//SYSPRINT DD SYSOUT=*
//DD1 DD DISP=SHR,UNIT=SYSDA,VOL=SER=volser ***-4-***
//FILE04 DD DSN=FILE04,LABEL=4,DISP=(OLD,PASS),
// UNIT=TAPE,VOL=REF=*.COPY.FILE02
//FILE05 DD DSN=FILE05,LABEL=5,DISP=(OLD,PASS),
// UNIT=TAPE,VOL=REF=*.COPY.FILE02
//FILE06 DD DSN=FILE06,LABEL=6,DISP=(OLD,PASS),
// UNIT=TAPE,VOL=REF=*.COPY.FILE02
//SYSIN DD *
```

continued ...

... continued

```

DELETE prefix.SYSTEM.PIELIB                ***-3-***
DEFINE CLUSTER -
    (NAME(prefix.SYSTEM.PIELIB) -          ***-3-***
    VOLUMES(volser) RECORDS(1500 150) -    ***-4-***
    CISZ(4096) SHR(2 3) -
    RECORDSIZE(4089 4089) NUMBERED)

REPRO IFILE(FILE04) OUTDATASET(prefix.SYSTEM.PIELIB) ***-3-***
LISTCAT ENT(prefix.SYSTEM.PIELIB) ALL      ***-3-***

DELETE prefix.EXAMPLE.PIELIB                ***-3-***
DEFINE CLUSTER -
    (NAME(prefix.EXAMPLE.PIELIB) -          ***-3-***
    VOLUMES(volser) RECORDS(400 100) -      ***-4-***
    CISZ(2048) SHR(2 3) -
    RECORDSIZE(2041 2041) NUMBERED)

REPRO IFILE(FILE05) OUTDATASET(prefix.EXAMPLE.PIELIB) ***-3-***
LISTCAT ENT(prefix.EXAMPLE.PIELIB) ALL      ***-3-***

DELETE prefix.USER.PIELIB                   ***-3-***
DEFINE CLUSTER -
    (NAME(prefix.USER.PIELIB) -              ***-3-***
    VOLUMES(volser) RECORDS(nnn nnn) -      ***-4-*** ***-6-***
    CISZ(2048) SHR(2 3) -
    RECORDSIZE(2041 2041) NUMBERED)

REPRO IFILE(FILE06) OUTDATASET(prefix.USER.PIELIB) ***-3-***
LISTCAT ENT(prefix.USER.PIELIB) ALL          ***-3-***

DELETE prefix.RESTORE.PIELIB                ***-3-***
DEFINE CLUSTER -
    (NAME(prefix.RESTORE.PIELIB) -           ***-3-***
    VOLUMES(volser) RECORDS(nnn nnn) -      ***-4-*** ***-6-***
    CISZ(4096) SHR(2 3) -
    RECORDSIZE(4089 4089) NUMBERED)

LISTCAT ENT(prefix.RESTORE.PIELIB) ALL      ***-3-***
/*
//

```

Tailoring CICS/OS/VS Installation JCL

At the right of some of the JCL lines shown above is the character string `***-n-***`, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB name and accounting information as needed.
2. Specify the volume serial of the EZ/KEY distribution tape.
3. Specify the dataset name prefix for EZ/KEY datasets.
4. Specify the volume serial of the disk pack where you want the EZ/KEY datasets to reside.
5. Specify the dataset name of the VSAM user catalog where you want to catalog the EZ/KEY datasets.
6. Specify the number of primary and secondary records for the USER PIELIB and RESTORE PIELIB datasets. When specifying the number of primary records, the minimum recommended number is 200. When specifying the number of secondary records, begin with 100 records per EZ/KEY user. For further information see "Choosing the Size of PIELIBs" later in this section.

CICS/OS/VS Installation JCL Notes

<u>JCL Step</u>	<u>Processing</u>
COPY	IEBCOPY the EZKYZ00, EZKYZ01, EZKYOPTN, and EZKYLAM load modules from FILE02 of the distribution tape to a partitioned data set.
IDCAMS	Invoke IDCAMS to define the VSAM data spaces for the EXAMPLE, RESTORE, SYSTEM and USER PIELIBs.

Change the names on the DEFINE CLUSTER statements to names that comply with your site's conventions. Change the VOLUMES parameter to identify the correct volumes for your site. Do not modify any other parameters of the DEFINE CLUSTER statements for the EXAMPLE and SYSTEM PIELIBs.

On the DEFINE for the RESTORE and USER PIELIBs, you must specify the number of records that these files are to contain. You must specify at least 200 primary records for your USER PIELIB. See "Choosing the Size of PIELIBs" below. The number of RECORDS and the RECORDSIZE parameters for the RESTORE PIELIB should be at least as large as SYSTEM or USER PIELIB, whichever is larger. For

further information, see "Choosing the Size of PIELIBs" later in this section.

NOTE: Do not DELETE and DEFINE the USER PIELIB if you are re-installing EZ/KEY.

Invoke IDCAMS to REPRO the SYSTEM PIELIB from the distribution tape to the SYSTEM PIELIB.

Invoke IDCAMS to REPRO the EXAMPLE PIELIB from the distribution tape to the EXAMPLE PIELIB.

Invoke IDCAMS to REPRO the USER PIELIB from the distribution tape to the VSAM USER PIELIB. The records in the REPROed USER PIELIB contain binary zeros.

NOTE: Do not REPRO the USER PIELIB if you are re-installing EZ/KEY.

Tailoring CICS/OS/VS Start-Up JCL

Add the following JCL to your CICS start-up deck to identify the SYSTEM, USER, EXAMPLE, and RESTORE PIELIBs, and to identify the load module library for EZ/KEY in the DFHRPL concatenation:

- Add the following DD statements to your CICS start-up deck to identify the PIELIBs:

```
//EZKSYS DD DSN=prefix.SYSTEM.PIELIB,DISP=SHR
//EZKUSR DD DSN=prefix.USER.PIELIB,DISP=SHR
//EZKXMP DD DSN=prefix.EXAMPLE.PIELIB,DISP=SHR
//EZKRSTR DD DSN=prefix.RESTORE.PIELIB,DISP=SHR
```

- Add the following DD statement to your CICS start-up deck as a part of the DFHRPL concatenation to identify the EZ/KEY load module library:

```
// DD DSN=prefix.EZKEY.LOAD,DISP=SHR
```

- Add the following DD statement to your CICS start-up deck to permit EZ/KEY users to submit batch jobs to the background:

```
//EZKE DD SYSOUT=(A,INTRDR)
```

Installation JCL

- Add any additional DD statements to your CICS start-up deck to identify any additional datasets to be used by EZ/KEY, such as PANVALET, PDS, or VSAM macro libraries. For example:

```
//PANDD DD DSN=your.EZTPLUS.macrolib,DISP=SHR
```

CHOOSING THE SIZE OF PIELIBS***Number of Blocks***

Select the number of blocks in the LIBSPACE (minimum of 200). This should be large enough for future expansion. A LIBSPACE cannot be expanded easily.

Size of Each Block

Enter the number of bytes for each block. EZ/KEY uses 10 bytes of each block for internal pointers. The blocksize should approximate the average size of members in the LIBSPACE. Large blocksizes waste disk space if most members are small; small blocksizes incur unnecessary I/O if most members are larger than one block.

For PIEBDAM, the blocksize must be the same value specified in the CICS FCT (File Control Table).

For PIEVSAM, the blocksize must be the CISIZE minus 7 bytes (for VSAM overhead). 505, 1017, 2041, or 4089 are good choices.

PIEBDAM PIELIBs on FBA devices should have blocksizes equal to or an exact multiple of 512. PIEBDAM PIELIBs on non-FBA devices should have blocksizes equal to or slightly less than one fourth of the DASD track length. For help in choosing an optimal block size, see either of the following IBM publications: *Direct Access Storage* (GA26-1638) or *Reference Summary* (GX20-1983).

Each record must be wholly contained in one block. The maximum recordsize is 10 characters less than the PIELIB blocksize (no spanned records).

Each block contains records from only one member. The LIBSPACE blocksize should be large enough to accommodate the largest record but small enough to avoid wasted space at the end of the block (505 - 4096 is the allowed range).

Each member must be wholly contained in one LIBSPACE. The size of the LIBSPACE (blocksize, number of blocks) must be large enough for the largest member.

One LIBSPACE must contain all members within a library.

With a blocksize of 1024, a 100-block LIBSPACE can hold 10 members with 200 80-byte records each (approximately 2,000 80-byte records or about 160,000 characters).

CICS/OS/VS TABLE CHANGES

You must incorporate the following changes into your CICS tables and reassemble your FCT, PCT, and PPT:

```
*
* PCT ENTRIES FOR EZ/KEY
*
      DFHPCT TYPE=ENTRY,PROGRAM=EZKYZOO,TRANSID=EZKE,           C
      SPURGE=NO,TPURGE=NO,SCRNSZE=ALTERNATE,                   C
      TWASIZE=42,DTB=NO
      DFHPCT TYPE=ENTRY,PROGRAM=EZKYZO1,TRANSID=EZKT,           C
      SPURGE=NO,TPURGE=NO,TWASIZE=42
*
* PPT ENTRIES FOR EZ/KEY
*
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYZOO,RES=YES
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYZO1
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYOPTN
*
* ADDITIONAL PPT ENTRIES FOR EZ/KEY IF PANVALET IS USED
*
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYLAM
      DFHPPT TYPE=ENTRY,PROGRAM=FGPAN23
      DFHPPT TYPE=ENTRY,PROGRAM=PANMODK
*
* ADDITIONAL PPT ENTRIES FOR EZ/KEY IF IDMS/IDD INTERFACE IS USED
*
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYIDMS
      DFHPPT TYPE=ENTRY,PROGRAM=IDMSINTC
*
* PLTPI ENTRY FOR EZ/KEY IF IDMS/IDD INTERFACE IS USED
*
      DFHPLT TYPE=ENTRY,PROGRAM=IDMSINTC
*
* ADDITIONAL PPT ENTRIES FOR EZ/KEY IF DB2 INTERFACE IS USED
*
      DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCF
      DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCI
      DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCR
      DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCS
      DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCT
```

continued ...

... continued

```

*
* FCT ENTRIES
*
      DFHFCT TYPE=DATASET,DATASET=EZKSYS,ACCMETH=(VSAM,RRDS),      C
          SERVREQ=(BROWSE,GET,NEWREC,PUT,UPDATE),                  C
          BUFND=2,STRNO=1,LOG=NO
      DFHFCT TYPE=DATASET,DATASET=EZKUSR,ACCMETH=(VSAM,RRDS),      C
          SERVREQ=(BROWSE,GET,NEWREC,PUT,UPDATE),                  C
          BUFND=2,STRNO=1,LOG=NO
      DFHFCT TYPE=DATASET,DATASET=EZKRSTR,ACCMETH=(VSAM,RRDS),      C
          SERVREQ=(BROWSE,GET,NEWREC,PUT,UPDATE),                  C
          BUFND=2,STRNO=1,LOG=NO
      DFHFCT TYPE=DATASET,DATASET=EZXEMPL,ACCMETH=(VSAM,RRDS)      C
          SERVREQ=(BROWSE,GET,NEWREC,PUT,UPDATE),                  C
          BUFND=2,STRNO=1,LOG=NO
*
* DCT ENTRIES FOR EZ/KEY
*
      DFHDCT TYPE=SDSCI,DSCNAME=EZKE,TYPEFLE=OUTPUT,                C
          RECFORM=FIXUNB,RECSIZE=80,BLKSIZE=80
*
      DFHDCT TYPE=EXTRA,DESTID=EZKE,DSCNAME=EZKE
*
* RCT ENTRY FOR EZ/KEY IF DB2 INTERFACE IS USED
*
      DSNCRCT TYPE=ENTRY,PLAN=DQEKCMD,TXID=EZKE
*

```

If you specify any parameters not shown in the sample table entries, they should be the same as the IBM supplied default values for those parameters. Otherwise, the results may be unpredictable.

If storage is often overutilized in your CICS system, you may need to specify **SPURGE=YES** on the PCT entry for **EZKE**. However, if CICS purges **EZ/KEY** users, any members being edited by those users are not saved. If **EZ/KEY** users receive the message:

```

      DFH2004 TRANSACTION EZKE PURGED - SYSTEM UNDER STRESS
      PLEASE RESUBMIT

```

the System Administrator must issue the **EZKT** transaction, and reply **YES** to the message:

```

      EZKEY025 - Type YES to remove EZ/KEY

```

before any users sign on to EZ/KEY again. Otherwise, users will receive the message:

EZKEY007 - User already signed on to EZ/KEY

CICS 1.7 Considerations

IF you are installing EZ/KEY in CICS 1.7, you may make the following changes to the FCT table entries for EZKSYS, EZKXMPL, EZKUSR, and EZKRSTR to avoid receiving any warning messages:

- Change:

ACCMETH=(VSAM,RRDS)

to:

ACCMETH=VSAM

- Change:

SERVREQ=(BROWSE,GET,NEWREC,PUT,UPDATE)

to:

SERVREQ=(BROWSE,READ,ADD,UPDATE)

Alternate Screen Size

EZ/KEY uses the alternate screen size defined in the TCT for 3270 terminals, if one is specified. You must ensure that the ALTSCRN= parameter of the DFHTCT macro (or ALTSCREEN for RDO) is correct for all of the terminals that you use with EZ/KEY. If this value is not specified correctly, only part of the screen is displayed, and it does not appear aligned properly. To correct this situation, change the ALTSCRN= parameter for the terminal to the correct size for that particular 3270 model. Alternatively, you can change SCRNSIZE=ALTERNATE to SCRNSIZE=DEFAULT on the PCT entry for EZ/KEY, which forces EZ/KEY to use the default screen size (DEFSCRN=) for all terminals using EZ/KEY, regardless of whether ALTSCRN= is specified for those terminals.

Refer to the following IBM manuals for further information:

- *CICS/VS Resource Definition Guide* (SC33-0149)
- *CICS/VS Resource Definition Online* (SC33-0186)
- *CICS/VS Resource Definition (Macro)* (SC33-0237)

CICS Dynamic Transaction Backout

EZ/KEY supports the use of the CICS Dynamic Transaction Backout (DTB) Facility. To use this facility, change DTB=NO to DTB=YES and specify DTIMOUT=0030 (or larger) on the PCT entry for EZ/KEY.

EZ/KEY performs its own commit processing, and so there is *no* benefit if you specify LOG=YES. If you do specify LOG=YES with EZ/KEY, spurious ASCF abends may occur. These abends are due to deadlocks caused by enques issued automatically by the CICS File Control Program on behalf of each task that updates any of these files.

You must, therefore, specify LOG=NO for all FCT entries used with EZ/KEY.

If you use Resource Definition Online (RDO) to define the transactions and programs for EZ/KEY with the CEDA transaction, there is no way to specify DTB=NO. When RDO is installed, all transactions run with DTB=YES. Therefore, you must specify LOG=NO for all FCT entries used with EZ/KEY.

CICS Multi-Region Option (MRO)

EZ/KEY supports the use of MRO to provide virtual storage constraint relief. To use MRO with EZ/KEY, change all of the DFHPCT TYPE=ENTRYs to TYPE=REMOTE for the primary CICS region (the one that has most of the terminals), and add SYSIDNT=xxxx to identify the remote CICS region in which EZ/KEY executes. Install only the DFHPCT entry in the primary CICS region; do not install the remaining table entries for EZ/KEY in the primary CICS region. Install all table entries (including DFHPCT) in the remote region. This allows EZ/KEY to use the Transaction Routing feature of MRO.

Refer to the IBM *CICS/OS/VS Intercommunication Facilities Guide* (SC33-0133) for further information.

CICS RUNAWAY TASK INTERVAL

EZ/KEY runs as either a *conversational* or *pseudo-conversational* transaction under CICS. Some activities, such as error-checking a large EASYTRIEVE PLUS program, may require more CPU cycles than other CICS applications.

To prevent spurious AICA abends, you may need to increase your Runaway Task Interval. To determine the current setting, enter the transaction:

CEMT INQUIRE RUNAWAY

The following screen is displayed:

INQUIRE RUNAWAY	
STATUS: RESULTS - OVERTYPE TO MODIFY	
MAxtasks(020)	
AMaxtasks(010)	
BMaxtasks(004)	
ATp(003)	-
AKp()	NOT IN SYSTEM
CUshion(016384)	
IOcp(00)	
Time(005000)	
RUnaway(0005000)	
STall(020000)	
SWt()	NOT ON OS
APPLID=CICS	
RESPONSE: NORMAL	TIME: 16.40.35 DATE: 85.037
PF: 1 HELP 3 END	7 SBH 8 SFH 9 MSG 10 SB 11 SF

The default value for CICS (and for this example) is 5000 milliseconds or five seconds. If your current setting is less than 5000 and you are experiencing spurious AICA abends while running EZ/KEY, you should increase your setting's value to at least 5000. To change the value for the duration of this CICS job, overtype the RUnaway(value) with a new value, such as 0005000, for 5 seconds.

CICS Runaway Task Interval

For more information on using the CEMT command, refer to the IBM *CICS/VS Operator's Guide* (SC33-0080).

The value recommended for EZ/KEY is 5 seconds (the IBM-supplied default).

Once you have decided on a Runaway Task Interval suitable for your environment, you may specify it in the CICS System Initialization Table (SIT) using the DFHSIT macro parameter ICVR=nnnnnn.

For more information on specifying the Runaway Task Interval in the SIT, refer to the IBM *CICS/OS/VS System Programmers Reference Manual* (SC33-0069).

CICS/OS/VS BDAM PIELIBS

If you want to use BDAM as the access method for one or more PIELIBs, you must specify RELTYPE=BLK in the FCT entry for each BDAM PIELIB, as shown below:

```
DFHFCT TYPE=DATASET, DATASET=fctname,ACCMETH=BDAM,RELTYPE=BLK, C
      RECFORM=(FIXED,UNBLOCKED),LRECL=nnnn,BLKSIZE=nnnn, C
      SERVREQ=(BROWSE,GET,NEWREC,PUT,UPDATE),LOG=NO |
```

INSTALLATION IN AN MVS/XA ENVIRONMENT

EZ/KEY is designed to run in 24-bit addressing mode and to reside in a 24-bit address space. Ensure that the EZ/KEY modules are link-edited with AMODE=24 and RMODE=24 so that they reside below the 16-megabyte line.

For more information about operation of MVS/370 programs in an MVS/XA environment, refer to the following IBM publications:

- *MVS/XA Conversion Notebook* (GC28-1143)
- *MVS/XA SPL: Initialization and Tuning Guide* (GC28-1149)
- *MVS/XA SPL: 31-Bit Addressing* (GC28-1158)

VALIDATE CICS INSTALLATION

To verify the correct installation of EZ/KEY, enter the following CICS transaction:

ezkey system

and press ENTER.

A panel is displayed, with the name **PROSTART** in the upper left corner:

PROSTART -----	Primary Selection Menu -----	EZ/KEY
COMMAND ==>		USERID: userid
		TIME : 10:31:51
		DATE : 05/12/88
Select Option ==>		
	0	Program Function Key Settings
	1	Edit an EASYTRIEVE PLUS Program
	2	Edit any Member
	3	Library and System Utilities
	4	Run a Program On-Line
	5	Submit a Program for Batch Processing
	6	Report Processing Facility
	T	Tutorial for EZ/KEY
	X	Exit the EZ/KEY System
EZ/KEY, Version X.X		
Copyright (c) 1983, 1988 Pansophic Systems, Inc.		

Immediately after validating the EZ/KEY installation, perform the procedures described in Section 4 of the *EZ/KEY Administrator's Guide*. Specifically, perform the procedures in the subsection that describes establishing CICS USERIDs and libraries.

*EASYTRIEVE PLUS VSAM Macro Libraries for CICS/OS/VS***EASYTRIEVE PLUS VSAM MACRO LIBRARIES FOR CICS/OS/VS**

EZ/KEY can read members from one or more EASYTRIEVE PLUS VSAM Macro libraries. For each library, a DD statement similar to:

```
//PANDD DD DISP=SHR,DSN=your.eztmacro.library
```

must be in the CICS startup JCL. Each macro library must be CONNECTed and ENABLEd to EZ/KEY.

Additional FCT entries are required since CICS File Control OPENs, READs, and CLOSEs are used to access EZTMACRO libraries. Sample FCT entries are shown below:

DFHFCT	TYPE=DATASET,DATASET=PANDD1,SERVREQ=(BROWSE,GET),	C
	BUFND=2,BUFNI=1,STRNO=1,ACCMETH=(VSAM,KSDS)	
DFHFCT	TYPE=DATASET,DATASET=PANDD2,SERVREQ=(BROWSE,GET),	C
	BUFND=2,BUFNI=1,STRNO=1,ACCMETH=(VSAM,KSDS)	
DFHFCT	TYPE=DATASET,DATASET=PANDD3,SERVREQ=(BROWSE,GET),	C
	BUFND=2,BUFNI=1,STRNO=1,ACCMETH=(VSAM,KSDS)	

As explained in the *EASYTRIEVE PLUS Installation Guide*, to use VSAM as the EASYTRIEVE PLUS macro library, specify MACRO=VSAM as an operand of the EZTPOPT options macro.

The System Administrator must use the External LIBSPACE Utility (option 3.a.4) to CONNECT each EZTMAC library to EZ/KEY.

PDS MACRO LIBRARIES FOR CICS/OS/VS

EZ/KEY can read members from one or more OS Partitioned Datasets or PDS libraries. For each PDS library, a DD statement similar to:

```
//PANDD DD DISP=SHR,DSN=your.PDS.library
```

must be in the CICS startup JCL. Each library must be CONNECTed and ENABLEd to EZ/KEY.

FCT entries are not required since OS OPENs, READs, and CLOSEs (with CICS WAITs) are used to access PDS libraries.

As explained in the *EASYTRIEVE PLUS Installation Guide*, to use a PDS as the EASYTRIEVE PLUS macro library, specify MACRO=PDS as an operand of the EZTPOPT options macro.

The System Administrator must use the External LIBSPACE Utility (option 3.a.4) to CONNECT each PDS library to EZ/KEY.

CICS/OS/VS SUBMIT TO BACKGROUND

To permit EZ/KEY users to submit batch jobs to the background, you must insert a JCL statement in the CICS startup JCL, for example:

```
//EZKE      DD SYSOUT=(A,INTRDR)
```

If this JCL statement is not present, or if the EZKE queue dataset is closed to CICS when a user requests a submit operation, the message **Job Queue Not Open** is displayed at the user's terminal.

You may choose to place the JCL for the submitted jobs in a dataset to be submitted when you desire. This may be accomplished as follows:

1. Change the DFHDCT TYPE=SDSCI entry to specify the correct record format and blocksize for the dataset to be used.
2. Change the JCL statement in the CICS startup JCL, for example:

```
//EZKE      DD DSN=your.EZKEY.CICS.JOBQUEUE,DISP=SHR
```

3. When you are ready to actually submit the jobs that have accumulated in the queue dataset to JES, issue:

```
CEMT INQUIRE QUEUE(EZKE)
```

4. Overtyp **Ope** on line 3 with **Clo** to close the dataset to CICS/VS.
5. Next, run a job that submits the contents of this dataset to JES. The JCL illustrated below can be used under MVS with either JES2 or JES3:

```
//SUBMITQ EXEC PGM=IEBGENER
//SYSPRINT DD DUMMY
//SYSUT1 DD DSN=your.EZKEY.CICS.JOBQUEUE,DISP=SHR
//SYSUT2 DD SYSOUT=(A,INTRDR)
//SYSIN DD DUMMY
//CLEARQ EXEC PGM=IEBGENER
//SYSPRINT DD DUMMY,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUT1 DD DUMMY,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUT2 DD DSN=your.CICS.EZKEY.JOBQUEUE,DISP=SHR
//SYSIN DD DUMMY
//
```

6. After the job runs, reissue:

CEMT INQUIRE QUEUE(EZKE)

and overwrite **Clo** with **Ope** to re-open the dataset to CICS/VS.
This allows EZ/KEY users to resume submitting jobs to the queue.

PANVALET CONSIDERATIONS FOR CICS/OS/VSE

EZ/KEY can read members from one or more PANVALET Release 11.0 or above libraries. For each library, a DD statement similar to:

```
//PANDD1 DD DISP=SHR,DSN=your.PANVALET.1library
```

must be in the CICS startup JCL, and each library must be CONNECTed and ENABLEd to EZ/KEY.

FCT entries are not required since OS OPENs, READs, and CLOSEs (with CICS WAITs) are used to access PANVALET libraries.

PPT entries for EZKYLAM and FGPN23 must be made as shown under the previous topic, "CICS/OS/VSE Table Changes".

As explained in the *EASYTRIEVE PLUS Installation Guide*, to use PANVALET as the EASYTRIEVE PLUS macro library, specify MACRO=PAN as an operand of the EZTPOPT options macro. PANDD, the default value of the MACDDN option of EZTPOPT, is the prefix of DDnames for EASYTRIEVE PLUS' macro library. Multiple PANVALET libraries can be used in EASYTRIEVE PLUS; the DDnames for the PANVALET libraries must be PANDD1, PANDD2, ..., PANDDn, where *n* is the value of MAC#LIB on the EZTPOPT macro. PANDD1 should be used as the DDname and External LIBSPACE Name for the PANVALET library to comply with the naming conventions of EASYTRIEVE PLUS.

The System Administrator must use the External LIBSPACE Utility (option 3.a.4) to CONNECT each PANVALET library to EZ/KEY.

When the first PANVALET library is CONNECTed to EZ/KEY, a CICS LOAD is issued for the module EZKYLAM, the access method for PANVALET (this module is distributed and installed as part of the EZ/KEY system for CICS). EZKYLAM issues a CICS LOAD for FGPN23, the options module for PANVALET. FGPN23 is a component of your PANVALET system; it is produced by assembling and link-editing the PVOPT macro. FGPN23 must be located in the DFHRPL concatenation of load libraries.

See the "Reconnect Procedures" and "Reinstallation Procedures" topics earlier in this section for further instructions.

*PANVALET Considerations for CICS/OS/VS****PANVALET Security for CICS/OS/VS***

There are no special EZ/KEY considerations when you do not use any of the PANVALET security features.

If the PANVALET library is protected with a library control code and the ++PRINT or ++WRITE command is suppressed for this library, you must specify this control code as the External Password. The control code should be the same as the code specified on the ++CONTROL statement in batch PANVALET.

If a PANVALET member that has an access code is invoked as a macro in an EASYTRIEVE PLUS program, you must precede the %macro-name statement by an "ACCESS 'access-code'" statement in the program.

To edit an access-protected member, use the Copy/Move Utility (option 3.3) to copy the member from the PANVALET library to your PIELIB. Specify the member access code as the Password of the FROM: member.

During an edit session, you can retrieve an access-protected PANVALET member by entering **COPY** (with no operands) on the COMMAND line. This invokes the COPY/MOVE Utility panel on which you can specify the member to be retrieved and its access code.

PANVALET/CICS Memory Requirements

If PANVALET is used with EZ/KEY, allow an additional 85K for the PANVALET module, 10K for each connected PANVALET library, and 10K for each enabled PANVALET library:

EZKYLAM module	85K
FGPAN23 module	1K
Each CONNECTED Library	10K
Each ENABLEed Library	10K per user

CICS/OS/VS PANVALET Version 14.0 or Above Considerations

If you are using PANVALET VERSION 14.0 or above, use the System Tuning Parameter panel (PROULTPS) to inform EZ/KEY to use the PANVALET Interface module, PANMODK, to access PANVALET libraries instead of EZKYLAM.

PANVALET Considerations for CICS/OS/VS

Type **3.a.7** on the PROSTART panel to obtain the System Tuning Parameter panel ((PROULTPS). On PROULTPS, type PANMODK in the PANVALET Module Name field.

So that EZ/KEY can locate the PANMODK module, the PANVALET load library must be added to the DFHRPL concatenation of load libraries.

The following PPT entry must be added:

```
DFHPPT TYPE=ENTRY, PROGRAM=PANMODK
```

EZ/KEY must execute in Conversational Mode to use PANMODK with EZ/KEY. Before attempting to CONNECT to any PANVALET libraries, and before executing the CICS Reconnect Procedures, sign on as **SYSTEM** and change the **Execute Conversational** System Tuning parameter to **YES**. This parameter indicates that all users execute in Conversational Mode.

MULTIPLE EZ/KEY SYSTEMS UNDER ONE CICS

Different departments have differing security needs. Some departments need complete isolation from users in other departments.

To provide increased security in a CICS environment, several distinct transaction IDs for EZ/KEY can be run under the same CICS system at the same time. These EZ/KEY systems are completely isolated from each other. Each has its own copy of the SYSTEM and EXAMPLE PIELIBs, and one or more USER PIELIBs.

EZ/KEY is normally installed in CICS with a transaction ID of EZKE. To allow multiple EZ/KEY systems to run under a single CICS, each EZ/KEY system:

- Prefixes all CICS ENQ/DEQ Resource names with the 4-character CICS Transaction Identifier (TRANSID)
- Prefixes its CICS Temporary Storage name with the TRANSID
- Uses the TRANSID as the Destination ID for the extra-partition dataset used to submit jobs
- Uses the first three characters of the CICS TRANSID as the first three characters of the FCTnames for SYSTEM (EZKSYS) and EXAMPLE (EZKXMPL) PIELIBs
- Uses the first three characters of the CICS TRANSID as the first three characters of the EZ/KEY Options Module (EZKYOPTN) in the PPT.
- The first three characters of the CICS TRANSID for program EZKYZ01 must be the same as the first three characters of the CICS TRANSID for EZKYZ00.

Suppose you have three departments: Accounting, Distribution, and Manufacturing. You can set up three different TRANSIDs. For example: EZAC for Accounting, EZDI for Distribution, and EZMA for Manufacturing. The removal transactions may be called EZAT, EZDT, and EZMT, respectively. The TRANSIDs must be unique within the first three characters. All three transactions can use the same EZKYZ00, EZKYZ01, and EZKYLAM modules since the EZ/KEY code is fully reentrant.

Separate FCT entries must be made for EZASYS, EZDSYS, EZMSYS, EZAXMPL, EZDXMPL, and EZMXMPL. Separate VSAM files must be allocated for each of these. Initially, the three sets of SYSTEM and

Multiple EZ/KEY Systems Under One CICS

EXAMPLE PIELIBs contain identical data from the Installation Tape.

Separate FCT entries must be made for EZAUSR, EZDUSR, and EZMUSR for the respective USER PIELIBs for each department (or you may use names such as ACCTNG, DISTRIB, and MNFCTNG). There must be a unique FCT entry and a corresponding VSAM dataset to contain each department's USER PIELIB.

You should assign one person as the EZ/KEY System Administrator for each of the departments. You may use ordinary CICS operator sign-on security (or some other CICS security facility) to control access to the various EZ/KEY transactions.

The EZ/KEY System Administrator for each department should sign on with the userid SYSTEM to initialize, format, and connect the USER PIELIBs. For example, one would sign on as EZKA SYSTEM, while another would use EZKD SYSTEM. Refer to the *EZ/KEY Administrator's Guide* for further information.

Once the separate EZ/KEY systems are established, each department is off and running. Each System Administrator can create new users, add new userids and libraries, and perform other maintenance functions independent of any other departments. For example, two different departments isolated as described above might each have a userid called SMITH.

The only time that you need to be involved is when one of the System Administrators wants to add a new PIELIB or macro library. A new PIELIB requires an additional FCT entry. A new external library (such as PANVALET) requires additional JCL to be added to the CICS start-up deck. An EASYTRIEVE PLUS VSAM macro library requires both an additional FCT entry and JCL in the CICS start-up deck.

For even more complete isolation, each of the various transactions could be routed to a different CICS address space, using Multi-Region Option (MRO). Each department's CICS address space would have only the FCT entries needed by that department. This could prevent an industrious System Administrator from connecting to libraries which do not belong to that department.

One note of caution – if more than one of these EZ/KEY systems connect up to a single physical PIELIB, then only one of them may specify CREATE or WRITE access. All others must specify READ access. Otherwise, the integrity of the PIELIB is NOT guaranteed, and most likely will be corrupted.

OPTIONAL SQL INSTALLATION PROCEDURES***EZ/KEY SQL Interface***

When you install the SQL option for CICS, your EZ/KEY users can use the PANSOPHIC SQL Interface to DB2. This allows them to syntax-check SQL commands embedded in their EASYTRIEVE PLUS programs.

SQL Interface Requirements

The SQL Option for EZ/KEY operates on all CICS/OS/VS MVS systems that support IBM's DB2. If you do not have DB2 installed, then you cannot use the EZ/KEY SQL support.

Your site must have the CICS/OS/VS DB2 Attachment Facility installed in the CICS system where EZ/KEY is to be installed. This requires that you add an additional table, the Resource Control Table (RCT), to your CICS system. This is defined using the DSNCRCT macro, which is provided with DB2. See the sample table entries provided with EZ/KEY for an example of how to code the RCT entry for EZ/KEY. For further information on adding RCT entries to your CICS system and for details on coding the DSNCRCT macro, refer to the following IBM publications:

- *IBM DATABASE 2 Installation* (SC26-4084)
- *IBM DATABASE 2 System Planning and Administration Guide* (SC26-4085)

EZ/KEY SQL Interface Disk Space Requirements

The EZ/KEY SQL Interface should be installed in the same load module library where EZ/KEY is installed. Specify this in the JCL as ***your.EZKEY.LOAD***.

The EZ/KEY SQL Interface modules require an additional 250K bytes in ***your.EZKEY.LOAD*** library.

EZ/KEY SQL Interface Virtual Memory Requirements

EZ/KEY uses approximately 90K of additional memory whenever it encounters an SQL statement. It uses the memory to bring the appropriate SQL Interface module into virtual storage and LINK to it. Depending on how many EZ/KEY users are active concurrently, the amount of memory used may actually be more than 90K at any one instant. The maximum storage required when all of the SQL Interface modules are resident in memory at the same time is 250K.

Each SQL statement requires approximately 100 bytes of additional dynamic storage for work areas.

Due to the additional storage requirements of the SQL Interface, you may need to increase the size of your CICS region by at least 512K in addition to the increases already required for normal EZ/KEY installation.

SQL Interface Option Installation

Running the job to install the EZ/KEY SQL Interface transfers the EZ/KEY SQL Interface modules from the installation tape to your EZ/KEY load module library.

Loading SQL Installation JCL

File 7 on the installation tape contains the JCL you need to install the SQL Interface. The following JCL copies file 7 from the installation tape to the sequential data set named userid.EZKEYDB2.JCL. This JCL must be modified as described below:

```
//jobname1 JOB accounting.info                      ***-1-***
//DB2JCL   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=FILE07,UNIT=TAPE,DISP=OLD,
//          LABEL=(7,SL),VOL=SER=volser             ***-2-***
//SYSUT2   DD DSN=userid.EZKEYDB2.JCL,               ***-3-***
//          DISP=(,CATLG,DELETE),
//          DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//          UNIT=SYSDA,VOL=volser,                   ***-4-***
//          SPACE=(3120,(100,25),RLSE)
```

JCL Notes

At the right of some of the JCL lines shown above is the character string *****-n-*****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB name and accounting information as needed.
2. Specify the volser of the EZ/KEY Installation Tape.
3. Specify the dataset name to use for the sequential dataset to contain the EZ/KEY SQL Installation JCL, or modify this statement to place the JCL into a member of a PDS of your choice.
4. Specify the volser of the disk to contain the above dataset.

SQL Installation JCL (File 7)

The JCL contained in file 7 on the distribution tape is shown below. Use it to install the SQL Interface for EZ/KEY. First, review the JCL. Then modify it as described in the notes that follow:

```
//jobname1 JOB accounting.info,USER=userid          ***-1-***
//*****
//**          INSTALL EZ/KEY SQL INTERFACE          **
//*****
//EZKEYSQL PROC DBRMLIB='your.DB2.DBRM.library',      ***-2-***
//          DCLGLIB='your.DB2.DCLGN.library',         ***-3-***
//          DB2LOD1='your.IBM.DB2.loadlib1',          ***-4-***
//          DB2LOD2='your.IBM.DB2.loadlib2',          ***-5-***
//          CICSLIB='your.CICS.LOAD.library',         ***-6-***
//          LOADLIB='your.EZKEY.LOAD',                ***-7-***
//          MEMNAME=DQEKCMD                           ***-8-***
//*****
//**          BUILD OBJECT LIBRARY OF SQL INTERFACE MODULES          **
//*****
//BUILD OBJ EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DUMMY
//SYSUT2 DD DSN=&&OBJLIB,DISP=(,PASS),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//          UNIT=SYSDA,SPACE=(CYL,(3,1,10))
```

continued ...

... continued

```

//*****
//**      ASSEMBLE COMMAND PROCESSOR MACRO      **
//*****
//ASMMACR EXEC PGM=IEV90,REGION=512K,PARM='DECK,RENT,LIST'
//SYSUT1 DD DSN=&&SYSUT1,UNIT=(SYSDA),SPACE=(CYL,(10,5))
//SYSLIN DD DSN=&&LOADSET,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(TRK,(10,5)),
//          DCB=BLKSIZE=3120
//SYSPUNCH DD DSN=&&ASM,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(CYL,(5,5,0)),
//          DCB=BLKSIZE=800
//SYSPRINT DD SYSOUT=*,DCB=BLKSIZE=3509,
//          UNIT=(,SEP=(SYSUT1,SYSPUNCH))
//*****
//**      DB2 PRECOMPILER      **
//*****
//DB2COMP EXEC PGM=DSNHPC,PARM='HOST(ASM),XREF,SOURCE'
//STEPLIB DD DSN=&DB2LOD1,DISP=SHR
//DBRMLIB DD DSN=&DBRMLIB(&MEMNAME),DISP=SHR
//SYSCIN DD DSN=&&DB2OUT,DISP=(MOD,PASS),
//          UNIT=SYSDA,SPACE=(TRK,(3,3)),
//          DCB=BLKSIZE=800
//SYSLIB DD DSN=&DCLGLIB,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(800,(1000,1000))
//SYSUT2 DD UNIT=SYSDA,SPACE=(800,(1000,1000))
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSIN DD DSN=&&ASM,DISP=(OLD,DELETE)
//*****
//**      ASSEMBLE COMMAND PROCESSOR      **
//*****
//ASMMDP EXEC PGM=IEV90,REGION=512K,PARM='DECK,LIST,RENT'
//SYSUT1 DD DSN=&&SYSUT1,
//          UNIT=(SYSDA,SEP=SYSLIB),
//          SPACE=(CYL,(10,5))
//SYSLIN DD DSN=&&LOADSET,DISP=(MOD,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(10,5)),
//          DCB=BLKSIZE=3120
//SYSPUNCH DD DSN=&&OBJLIB(DMSMCM2),DISP=(OLD,PASS)
//SYSPRINT DD SYSOUT=*,DCB=(BLKSIZE=3509),
//          UNIT=(,SEP=(SYSUT1,SYSPUNCH))
//SYSIN DD DSN=&&DB2OUT,UNIT=SYSDA,DISP=(OLD,DELETE)

```

continued ...

... continued

```

//*****
//**      LINK-EDIT SQL INTERFACE      **
//*****
//LINKEDIT EXEC PGM=IEWL,PARM='CALL,LET,LIST,MAP,RENT'
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLIB DD DSN=&&OBJLIB,DISP=(OLD,DELETE,DELETE)
//DB2LIB DD DSN=&DB2LOD1,DISP=SHR
//SYSLMOD DD DSN=&LOADLIB,DISP=OLD
//*****
//**      INVOKE TSO DSN COMMAND TO DO BIND      **
//*****
//DB2BIND EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB DD DSN=&DB2LOD1,DISP=SHR
// DD DSN=&DB2LOD2,DISP=SHR
//DBRMLIB DD DISP=SHR,DSN=&DBRMLIB
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//REPORT DD SYSOUT=*
//*
//      PEND
//INSTSQL EXEC EZKEYSQL
//BUILDOBJ.SYSIN DD *
./ ADD NAME=.....
ESD
...
/*
//ASMMACR.SYSIN DD *
      PRINT OFF
...
      DQSMCMD MAXSTMT=10,MAXCUR=6,PLAN=DQSMCMD      ***-8-***
      END
/*
//LINKEDIT.SYSLIN DD *
      INCLUDE ...
...
/*

```

continued ...

... continued

```

//DB2BIND.SYSTSIN DD *
  DSN SYSTEM(ssid)                                     ***-9-***
  BIND PLAN(DQEKCMD) MEMBER(DQEKCMD)                  ***-8-***
    ACT(REPLACE) ISOLATION(CS)
  RUN PROGRAM(DSNTIAD) PLAN(DSNTIAD2) -
    LIB('your.IBM.DB2.loadlib2')                     ***-5-***
  END
/*
//DB2BIND.SYSIN DD *
  GRANT EXECUTE ON PLAN DQEKCMD TO PUBLIC;             ***-8-***
/*
//

```

Notes for SQL Interface Installation JCL

At the right of some of the JCL lines shown above is the character string *****-n-*****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB name and accounting information as needed. Change the USER=userid parameter to specify a valid DB2 userid with DBA authority.
2. Change the DBRMLIB= parameter to identify your IBM DB2 DBRM library.
3. Change the DCLGLIB= parameter to identify your IBM DB2 DCLGN library that contains the definition of the SQLCA.
4. Change the DB2LOD1= parameter to identify the IBM DB2 load library where the DB2 module DSNTIAR resides.
5. Change the DB2LOD2= parameter and the LIB(...) parameter to identify the IBM DB2 load library where the DB2 module DSNTIAD resides. Note that at your site this may be the same library referred to in Note #4 above.
6. Change the CICSLIB= parameter to identify the IBM CICS load library where the CICS module DFHEAI resides.
7. Change the LOADLIB= parameter to identify the EZ/KEY load library where the SQL Interface is to be installed.

Optional SQL Installation Procedures

8. Change the MAXSTMT= and MAXCUR= parameters to the same values as specified in your EASYTRIEVE PLUS SQL installation job. Change the MEMNAME=DQSMCMD parameter to MEMNAME=DQEKCMD, or any other name, as long as the name you choose is different from the name specified in your EASYTRIEVE PLUS SQL installation job. The name recommended for EZ/KEY is DQEKCMD.
9. Change the SYSTEM(ssid) parameter to specify the DB2 subsystem that EZ/KEY operates with. This should be the same as the one specified in your EASYTRIEVE PLUS SQL installation job.

Additional CICS Considerations

EZ/KEY must be able to access the following modules under CICS: DQSEKCF, DQSEKCI, DQSEKCR, DQSEKCS, and DQSEKCT. Therefore, if you have not already done so, you must install the PPT entries for these modules. For further information see "CICS/OS/VS Table Changes" earlier in this section.

If you have just completed installing these modules, you may need to tell CICS to load new copies of them. To get new copies, do *one* of the following:

- Bring CICS down and up again.

Or

- Issue the CICS command.

CEMT INQUIRE PROGRAM(DQSEKC*)

Then issue a *Newcopy* for each of these programs, as shown in the following illustrations.

Additional DB2 Considerations

```
INQUIRE PROGRAM(DQS*)
STATUS:  RESULTS - OVERTYPE TO MODIFY
Pro(DQSEKCF ) Len(0000000) Res(000) Use(000000) Ass Ena new
Pro(DQSEKCI ) Len(0000000) Res(000) Use(000000) Ass Ena new
Pro(DQSEKCR ) Len(0000000) Res(000) Use(000000) Ass Ena new
Pro(DQSEKCS ) Len(0000000) Res(000) Use(000000) Ass Ena new
Pro(DQSEKCT ) Len(0000000) Res(000) Use(000000) Ass Ena new

RESPONSE: NORMAL                                APPLID=CICS17
PF: 1 HELP          3 END                        TIME: 14.22.47  DATE: 87.244
                                                    7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Exhibit 4.1: Requesting a Newcopy

To issue a newcopy, tab the cursor to the position next to the word **Ena**, type the word **new**, as shown, and press ENTER.

The result is shown here:

INQUIRE PROGRAM(DQS*)

STATUS: RESULTS - OVERTYPE TO MODIFY

Pro(DQSEKCF)	Len(0035456)	Res(000)	Use(000000)	Ass Ena	NEW COPY
Pro(DQSEKCI)	Len(0059192)	Res(000)	Use(000000)	Ass Ena	NEW COPY
Pro(DQSEKCR)	Len(0026408)	Res(000)	Use(000000)	Ass Ena	NEW COPY
Pro(DQSEKCS)	Len(0065864)	Res(000)	Use(000000)	Ass Ena	NEW COPY
Pro(DQSEKCT)	Len(0052400)	Res(000)	Use(000000)	Ass Ena	NEW COPY

RESPONSE: NORMAL
PF: 1 HELP 3 END

APPLID=CICS17
TIME: 14.23.12 DATE: 87.244
7 SBH 8 SFH 9 MSG 10 SB 11 SF

Exhibit 4.2: Results of Newcopy Request

DB2 USERIDS

The EZ/KEY SQL Interface under CICS invokes the DB2 CICS Attachment facility to perform syntax checking of SQL statements embedded in your EASYTRIEVE PLUS programs.

DB2 uses one of several methods to determine your userid, depending on how DB2 was installed at your site. The most common method is to use the three-character CICS operator-id.

The CICS operator-id may or may not have anything to do with the EZ/KEY userids at your site. EZ/KEY users must be aware of this difference when coding SQL statements in their EASYTRIEVE PLUS programs. All security and access checking performed by DB2 is based on the DB2 userid, not the EZ/KEY userid.

Your DB2 administrator must GRANT authority to access the appropriate tables within DB2 to each of the three-character CICS operator-ids, as required. You could make EZ/KEY userids match the operator-ids at your site, eliminating a possible source of confusion for your EZ/KEY users.

If you have CICS 1.7 and DB2 Release 3 or above, you can specify that DB2 use the new eight-character CICS userid, rather than the old three-character operator-id, as the DB2 userid. Instruct your EZ/KEY users to sign-on to CICS using the new CESN transaction, rather than using the old CSSN transaction. You could also make the new eight-character CICS userids the same as the EZ/KEY userids, eliminating a possible source of confusion for your EZ/KEY users.

If you have TSO DB2 users at your site, you may want to ensure that the EZ/KEY userid and CICS userid match the TSO userid. That way, only one userid per user must be identified to the DB2 administrator for purposes of issuing the appropriate GRANTS.

OPTIONAL IDMS/IDD INSTALLATION

Your site must have IDMS-CV (Central Version) installed and working with the CICS system where EZ/KEY is installed. This requires that you add table entries to your CICS system. For an example of how to code the PLT and PPT entries for IDMSINTC, see the sample table entries provided with EZ/KEY.

For more information on adding table entries to your CICS system and for details on installing IDMS/IDD, refer to the following Cullinet publications:

- *IDMS-CV/DC System Operations*. See the section on TP monitor considerations for CICS.
- *Cullinet System Software, Integrated Installation Guide*. See the section on special considerations for interfaces, particularly the Central Version/CICS interface.
- *Cullinet System Software, System Generation*. See the section on ABEND detection.

IDMS/IDD Installation JCL

In addition to the requirements noted above, you must also run the following job to create the EZKYIDMS module EZ/KEY needs to access the IDMS IDD Interface. This allows you to use the IDD statement from within EZ/KEY.

Optional IDMS/IDD Installation

A copy of the JCL shown below is in **EXAMPLE.PIELIB** in the member named **LKIDDMVS JCL**. The JCL must be modified as described below:

```
//jobname1 JOB (accounting info),REGION=512K          ***-1-***
//LKED      EXEC  PGM=IEWL,REGION=512K,
//          PARM='LIST,LET,MAP,NCAL,RENT'
//SYSUT1    DD   UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSLIB    DD   DISP=SHR,DSN=your.CICS.loadlib        ***-2-***
//IDMSLIB   DD   DISP=SHR,DSN=your.IDMS.R010.OBJLIB     ***-3-***
//SYSLMOD   DD   DISP=SHR,DSN=your.EZKEY.LOAD         ***-4-***
//SYSPRINT  DD   SYSOUT=*
//SYSLIN    DD   *
            INCLUDE IDMSLIB(IDMSCINT)
            INCLUDE SYSLIB(DFHEAI)
            ENTRY IDMSCINT
            NAME EZKYIDMS(R)
/*
//
```

Tailoring IDMS/IDD JCL

At the right of some of the JCL lines shown above is the character string *****-n-*****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB statement as required for your installation.
2. Specify the dataset name where the IBM CICS module DFHEAI resides.
3. Specify the dataset name where the IDMS module IDMSCINT resides.
4. Specify the dataset name of your EZ/KEY load library, where all EZ/KEY modules reside. This job creates a new module, EZKYIDMS, and places it in this library. EZKIDMS is used by EZ/KEY to access the IDMS/IDD interface.

After tailoring, submit the above JCL for execution under MVS. This job should run with no errors and receive a return code of 0.

Additional IDMS/IDD Considerations

To permit EZ/KEY users to access the IDD using Central Mode, add the following DD statement to the CICS start-up JCL:

```
//SYSCTL DD DSN='your.IDMS.control.dataset',DISP=SHR
```

If you have not already done so, you must install the PPT entry for the module EZKYIDMS. See "CICS/OS/VS Table Changes" earlier in this section for further information.

If you have just completed installing this module, you may need to tell CICS to load a new copy. To do this, do *one* of the following:

- Bring CICS down and then up again.

Or

- Issue the CICS command.

CEMT INQUIRE PROGRAM(EZKYIDMS)

Then issue a *Newcopy* for this program, as shown in the following illustrations.

```
INQUIRE PROGRAM(EZKYIDMS)
STATUS:  RESULTS - OVERTYPE TO MODIFY
Pro(EZKYIDMS) Len(0000000) Res(000) Use(000000) Ass Ena new
```

```
RESPONSE: NORMAL
PF: 1 HELP      3 END
```

```
APPLID=CICS17
TIME: 14.22.47  DATE: 87.244
7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Exhibit 4.3: Requesting a Newcopy

To issue a newcopy, tab the cursor to the position next to the word **Ena**, type the word *new*, as shown, and press ENTER. The result is shown in the following illustration.

INQUIRE PROGRAM(EZKYIDMS)

STATUS: RESULTS - OVERTYPE TO MODIFY

Pro(EZKYIDMS) Len(0000328) Res(000) Use(000000) Ass Ena NEW COPY

RESPONSE: NORMAL

PF: 1 HELP 3 END

APPLID=CICS17

TIME: 14.22.47 DATE: 87.244

7 SBH 8 SFH 9 MSG 10 SB 11 SF

Exhibit 4.4: Results of Newcopy Request

IDMS/IDD Interface Storage Usage

With a single IDD statement, an EZ/KEY user can bring into virtual storage hundreds or even thousands of field definitions from the IDMS/IDD Dictionary.

Each field definition in EZ/KEY requires approximately 256 bytes of symbol table space in virtual memory, which is obtained from the CICS dynamic storage area belonging to the user who makes the request. This space is normally released after a short period of time.

If you enable the *Incremental Compile* feature of EZ/KEY, these field definitions remain in virtual storage for the duration of the EZ/KEY user's Edit session. Caution is therefore advised with regard to using this feature of EZ/KEY in conjunction with the IDD interface.

Unless you have installed EZ/KEY in its own large MRO region, you should not set *Incremental Compile* to YES; rather ensure that it is set to NO (the default). For further information, see the discussion of tuning parameters in Section 3 of the *EZ/KEY Administrator's Guide*.

ADDITIONAL IDMS/IDD CICS CONSIDERATIONS

Depending on how your IDMSINTC module for CICS is generated, you may need to add the following DD statement to your CICS start-up JCL to permit EZ/KEY users to access the IDD using Central Mode:

```
//SYSCTL DD DSN='your.IDMS.control.dataset',DISP=SHR
```

The topic "IDMSINTC Generation" discusses how to generate the IDMSINTC module using the IDMSINTC macro.

IDMSINTC Generation

Under CICS, EZ/KEY requires a particular version of the IDMS module IDMSINTC. If you do not already have such a version available, you must generate and install it under CICS.

The following example illustrates how to code the IDMSINTC macro to produce a version of the IDMSINTC module for use with CICS:

```
IDMSINTC IDMSINTC SVC=255,                                X
      CVNUM=01,                                           X
      CWADISP=0,                                          X
      CSASIZE=512,                                        X
      CICSLVL=16,                                         X
      ACTDISP=12,                                         X
      OPSYS=MVS,                                          X
      TPNAME=CICS,                                       X
      SYSCTL=,                                           X
      ESCDLI=NO,                                         X
      NODENAM=,                                          X
      DBNAME=                                           X
      END
```

The SVC, CVNUM, CWADISP, CICSLVL, ACTDISP, NODENAM and DBNAME parameters may be different for your installation. Consult your IDMS Data Base Administrator or a systems programmer who installed IDMS for assistance.

For more information regarding use of the IDMSINTC macro instruction and on installing IDMS in a CICS environment, refer to the Cullinet *IDMS-CV/DC Integrated Installation Guide*. Specifically, see the section covering special considerations for the Central Version/CICS interface.

IDMSCINT Generation

A version of the IDMS module IDMSCINT must be available for link editing with EZ/KEY. This module must have been generated for use with Command Level programs.

The following example illustrates how to code the IDMSCINT macro to produce a Command Level version of the IDMSCINT module.

```
IDMSCINT IDMSCINT CWADISP=0,EXEC=YES  
END
```

The EXEC=YES parameter indicates Command Level programming. The CWADISP parameter value may be different for your installation. Consult your IDMS Data Base Administrator or a systems programmer who installed IDMS for assistance.

Refer to the following Cullinet publication for more information regarding use of the IDMSCINT macro instruction: *IDMS-CV/DC Integrated Installation Guide*. Specifically, see the section covering special considerations for the Central Version/CICS interface.

IDMS Background Execution Considerations**Central Version Mode Background Operation**

If EASYTRIEVE PLUS is to access the IDD through Central Version, your site must have IDMS-CV (Central Version) installed on the MVS system where batch EASYTRIEVE PLUS runs.

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Central Mode, add the DD statement shown below to the EZTPLUS Header JCL for Background Execution in EZ/KEY.

Identify the IDMS control file for use by EASYTRIEVE PLUS programs:

```
//SYSCTL DD DSN=your.idms.control.file,DISP=SHR
```

where *your.idms.control.file* is the name of the control file for IDMS. Consult your IDMS Data Base Administrator for assistance. Refer to the *EASYTRIEVE PLUS Installation Guide* for more information.

For more detailed information on operating IDMS-CV in MVS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

Local Mode Background Operation

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Local Mode, add the DD statements shown below to the EZTPLUS Header JCL for Background Execution in EZ/KEY:

1. Identify the load libraries where the Cullinet IDMS modules reside; this provides access to the modules needed to execute IDMS in Local Mode:

```
//STEPLIB DD DSN=your.EZTP.load.library,DISP=SHR  
//          DD DSN=your.IDMS.load.library,DISP=SHR
```

where *your.IDMS.load.library* is the name of the load module library for IDMS.

Additional IDMS/IDD CICS Considerations

2. Identify the IDMS journal file for use by EASYTRIEVE PLUS programs:

```
//SYSJRNL DD DSN=your.idms.journal.file,DISP=(,KEEP),UNIT=TAPE
```

where *your.idms.journal.file* is the name of a journal file for IDMS. Note that you can either allocate this data set to disk or specify a DUMMY data set.

3. Identify all of the databases accessed by EASYTRIEVE PLUS programs:

```
//dbfile1 DD DSN=your.idms.database.file1,DISP=SHR
```

```
//dbfileN DD DSN=your.idms.database.fileN,DISP=SHR
```

where *dbfileN* and *your.idms.database.fileN* are the DDname and data set name, respectively, for each database file that may be accessed by your EASYTRIEVE PLUS programs.

There must be a DD statement in the EZTPLUS JCL for each database that is accessed by any EASYTRIEVE PLUS program running in Local Mode. Consult your IDMS Data Base Administrator for assistance. Refer to the *EASYTRIEVE PLUS Installation Guide* for more information.

For more detailed information on operating IDMS-CV in MVS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- MVS Considerations

SECTION 5 CICS/DOS/VSE INSTALLATION

Introduction	5-1
Storage Estimates	5-1
Reinstalling EZ/KEY under CICS/DOS/VSE	5-2
Reconnect Procedures	5-2
Reinstallation Procedures	5-2
Installation Files	5-4
CICS Release Dependencies	5-5
CICS 1.5 Considerations	5-5
POWER Dependencies	5-5
Disk Space Requirements	5-5
Memory Requirements	5-6
Auxilliary Temporary Storage Requirements	5-7
Installation JCL	5-8
Tailoring CICS/DOS/VSE Installation JCL	5-10
CICS/DOS/VSE Installation JCL Notes	5-11
Tailoring CICS/DOS/VSE Start-Up JCL	5-13
Choosing the Size of PIELIBs	5-14
Number of Blocks	5-14
Size of Each Block	5-14
CICS/DOS/VSE Table Changes	5-15
CICS 1.7 Considerations	5-17
Alternate Screen Size	5-17
CICS Dynamic Transaction Backout	5-18
CICS Multi-Region Option (MRO)	5-18
CICS Runaway Task Interval	5-20
CICS/DOS/VSE BDAM PIELIBS	5-21
ICCF Considerations for CICS/DOS/VSE	5-22
Validate CICS Installation	5-23
EASYTRIEVE PLUS VSAM Macro Libraries for CICS/DOS/VSE	5-24
PANVALET Considerations for CICS/DOS/VSE	5-25
PANVALET Security for CICS/DOS/VSE	5-25
PANVALET/CICS Memory Requirements	5-26
CICS/DOS/VSE PANVALET Version 12.0 or Above	

Considerations	5-26
Multiple EZ/KEY Systems Under One CICS	5-28
SQL Interface Option Installation	5-30
SQL Operating Environments	5-30
EZ/KEY SQL Interface Virtual Memory Requirements ..	5-30
SQL Interface Installation	5-31
SQL Interface Installation - VSE/SP 2.1	5-32
VSE/SP 2.1 JCL (File 8)	5-32
VSE/SP 2.1 SQL Installation JCL Notes	5-37
Tailoring VSE/SP 2.1 SQL Installation JCL	5-38
SQL Interface Installation - DOS/VSE (pre-SP 2.1)	5-41
Pre-SP 2.1 JCL (File 9)	5-41
DOS/VSE SQL Installation JCL Notes	5-46
Tailoring DOS/VSE SQL Installation JCL	5-47
Additional SQL CICS Considerations	5-51
Additional SQL/DS Considerations	5-52
SQL/DS Userids	5-53
Optional IDMS/IDD Installation	5-54
IDMS/IDD Installation JCL	5-54
Tailoring IDMS/IDD JCL	5-55
Additional IDMS/IDD CICS Considerations	5-56
IDMS/IDD Interface Storage Usage	5-58
Additional IDMS Considerations	5-59
IDMSCINT Generation	5-59
IDMSINTC Generation	5-60
IDMS Background Execution Considerations	5-60
Central Version Mode Background Operation	5-60
Local Mode Background Operation	5-61

INTRODUCTION

This section contains detailed procedures for installing EZ/KEY under CICS/DOS/VSE. Also review the *Administrator's Guide* to determine how to operate EZ/KEY at your site.

EZ/KEY can execute as either a conversational or as a pseudo-conversational CICS transaction.

When EZ/KEY executes as a conversational transaction, CICS allocates memory for EZ/KEY when a user signs on. EZ/KEY acquires additional memory as needed, and releases memory when no longer needed. The memory acquired for the basic EZ/KEY functions is freed when the user signs off.

When EZ/KEY executes as a pseudo-conversational transaction, CICS allocates memory for EZ/KEY when a user presses ENTER. EZ/KEY frees memory when it displays the next panel for a user. Some memory is shared by all EZ/KEY users. This memory is freed when the last EZ/KEY user signs off.

Storage Estimates

EZ/editor is an in-storage editor, similar to XEDIT or ISPF/PDF. The size of the program or data being edited is limited by the availability of virtual storage.

These storage estimates are affected by user activities and installation options which may require additional virtual storage for EZ/KEY. The following factors can affect virtual storage availability:

- The number of open files
- The number of CICS programs residing in virtual storage
- The number of CICS transactions (users) currently active
- The number of *conversational* programs currently running.

See the section titled "Memory Requirements" for further details on specific memory requirements.

Memory requirements, disk space sizes, JCL, and CICS table entries presented in this section are for planning purposes only. More precise information is provided in the files on the EZ/KEY distribution tape.

REINSTALLING EZ/KEY UNDER CICS/DOS/VSE

If you have already installed EZ/KEY under CICS, almost everything that you need is already in place. For Version 3.2 of EZ/KEY you must:

- Add a new PCT entry for transaction EZKT for program EZKYZ01
- Add a new PPT entry for program EZKYOPT
- Remove the PPT entries for programs EZKYSCOM and EZKYUTIL because they are no longer required.

Also, if you choose a transaction ID other than EZKE for EZ/KEY, the first three characters of the EZ/KEY (EZKE) transaction must match those of the removal transaction (EZKT).

Reconnect Procedures

In CICS environments, EZ/KEY maintains a SYSTEM CONNECT member in the SYSTEM PIELIB. This member contains the names of all LIBSPACES which are connected to the user community sharing a particular SYSTEM PIELIB.

This CONNECT member is lost when a new release of EZ/KEY is installed from the distribution tape. Have the EZ/KEY System Administrator make a backup copy of the SYSTEM CONNECT member, as described under Reconnect Procedures in the *EZ/KEY Administrator's Guide*, before you install a new release of EZ/KEY.

Reinstallation Procedures

Once the EZ/KEY System Administrator has made a backup copy of the SYSTEM CONNECT member, you may proceed with the installation as follows:

Have available for review, a copy of the output produced by the installation jobstream when the last version of EZ/KEY was installed. If you cannot locate this information, review the appropriate material in the previous version of the *EZ/KEY Installation Guide*.

Reinstalling EZ/KEY under CICS/DOS/VSE

Review the new sample table entries for EZ/KEY and compare them with your current table entries. Note any differences. Make the necessary changes to your CICS tables. If you have specified parameters other than those shown in the sample entries, ensure that they are the same as the IBM supplied defaults for those values.

Review the sections that pertain to the new features described above.

Review the new installation JCL. Make any changes needed for your installation's environment.

Once you have made the necessary changes, run the new installation procedure, and then verify the results as described in this section under "Validate CICS Installation."

INSTALLATION FILES

The EZ/KEY CICS/DOS/VSE system is distributed on a standard labeled tape that contains ten files:

- File 1 (DSN=FILE01) - Contains the Job Control Language (JCL) to install EZ/KEY. This file is blocked 80/3120.
- File 2 (DSN=FILE02) - Contains the phases for the EZ/KEY system, in BACKUP/RESTORE format.
- File 3 (DSN=FILE03) - Contains the SYSTEM PIELIB in IDCAMS REPRO format.
- File 4 (DSN=FILE04) - Contains the EXAMPLE PIELIB in IDCAMS REPRO format.
- File 5 (DSN=FILE05) - Contains the USER PIELIB in IDCAMS REPRO format.
- File 6 (DSN=FILE06) - Contains the CICS/DOS/VSE table changes (FCT, PCT, PPT). This file is blocked 80/3120.
- File 7 (DSN=FILE07) - Contains the EZ/REPORT User's Guide. This file is blocked 80/3120.
- File 8 (DSN=FILE08) - Contains the JCL and object code to install the SQL Interface for VSE/SP 2.1 and above systems. It is blocked at 80/3120.
- File 9 (DSN=FILE09) - Contains the JCL and object code to install the SQL Interface for DOS/VSE Pre-SP 2.1 systems. It is blocked at 80/3120.
- File 10 (DSN=FILE10) - Contains the object modules needed to link-edit the EZ/KEY phases. These object modules are in a relocatable library in DOS/VSE BACKUP/RESTORE format.

CICS Release Dependencies

EZ/KEY operates under CICS 1.5 and later releases.

EZ/KEY is written as a Command Level Assembler language program. To run EZ/KEY under CICS, you must specify **EXEC=YES** in the SIT (or allow it to default) or specify **EXEC=YES** in the SIT overrides in the CICS start-up deck.

See the following IBM publications for further information:

- *CICS/DOS/VS Resource Definition (Macro)* (SC33-0149)
SIT - System Initialization Table (DFHSIT)
- *CICS/DOS/VS Installation and Operations Guide* (SC33-0070)
System Initialization Override Parameters

CICS 1.5 Considerations

In order to run EZ/KEY under CICS 1.5 you must call PANSOPHIC Customer Service to obtain a PTF to apply to EZ/KEY.

If you are running CICS 1.5, you must re-link EZKYZ00 and EZKYZ01 with a copy of DFHEAI. DFHEAI is located in your CICS/DOS/VSE Release 1.5 system. See the section titled "Link-Editing CICS/DOS/VSE Phases" in "Appendix A" of this manual for sample JCL and a description of how to re-link EZKYZ00 and EZKYZ01.

POWER Dependencies

EZ/KEY uses PUTSPOOL to submit jobs for background execution.

You must specify **POWER SPOOL=YES** in your VSE system generation.

Disk Space Requirements

Name	Disk Blocks	Blocksize or Control Interval	
		Size	3350 Tracks
EZ/KEY load library	1350	1024	120
SYSTEM PIELIB	1500	4089	375
USER PIELIB	1000	2041	250
RESTORE PIELIB	1500	4089	375
EXAMPLE PIELIB	400	2041	50

Memory Requirements

Increase the CICS Partition Size by 768K for the EZ/KEY main module EZKYZ00 (256K), the program cache (448K), plus the panel cache (48K), and by an additional amount that depends on the number of users who will be using EZ/KEY at the same time and on whether or not EZ/KEY operates in Conversational or Pseudo-conversational mode (the default for CICS).

The program cache limit defaults to 448K and the panel cache limit defaults to 48K for CICS. See the section "Site Options" in the *EZ/KEY Administrator's Guide* for a description of how to change the size of the Program and Panel Caches. If you change the size of either of the cache limits, you may need to adjust your CICS Partition Size accordingly.

When EZ/KEY is executing in Conversational mode, the virtual storage required for each user is needed for the duration of the EZ/KEY session. See the section "Tuning Parameters" in the *EZ/KEY Administrator's Guide* for a description of how to change the Execute Conversational parameter.

When EZ/KEY is executing in Pseudo-conversational mode, the virtual storage required for each user is needed only for a short duration. Before the next screen is displayed, the per-user data is swapped out to Auxiliary Temporary Storage. CICS dynamic area virtual storage is acquired when the user presses ENTER and is released just before the next screen is displayed.

Use the following formula to compute your user memory requirements:

48K per user (initial requirements to sign on)
plus 200 bytes times the number of lines being edited

EZ/KEY is an in-storage editor, similar to XEDIT or ISPF/PDF. The size of a program or of data being edited is limited by the availability of virtual storage. One hundred bytes of storage are required for each line a user is editing, plus approximately 100 bytes per line when EZ/Checker is being used.

For example, suppose you anticipate that the largest member to be edited by any user is 500 lines in length and that there will be a maximum of 10 simultaneous users. To determine the total memory required, perform the following calculations:

$$\begin{array}{r} 500 \times 200 = 200 \text{ K} \\ + \quad 48 \text{ K} \\ \hline 148 \text{ K per user} \\ \times \quad 10 \text{ users} \\ \hline 1480 \text{ K total additional storage required} \end{array}$$

When EZ/KEY is executing in Pseudo-conversational mode, the actual amount of storage needed at any one moment is usually much less than the total shown above. This depends, however, on the actual activities being performed by users who are signed on and active within EZ/KEY at that time.

Auxilliary Temporary Storage Requirements

You may need to increase the size of Auxilliary Temporary Storage (disk space), depending on the maximum number of users using EZ/KEY at the same time.

Use the formula shown above to determine the additional Auxilliary Temporary Storage needed for EZ/KEY to operate pseudo-conversationally under CICS. For planning purposes, assume that all users may be swapped out to Auxilliary Temporary Storage at the same time.

INSTALLATION JCL

The following JCL is on the first file of the EZ/KEY distribution tape. You must load this file onto your system and edit it. Items that you should change are identified by the character string ***-n-*** to the right:

```

* $$ JOB UNM=jobname,CLASS=A,DISP=D                ***-1-***
* $$ LST CLASS=A,DISP=D                            ***-1-***
// JOB EZKEY1                CREATE LIBRARY FOR EZ/KEY ***-1-***
// ASSGN SYS003,xxx          ***-2-***
// DLBL EZKEY,'your.EZKEY.phase.1library',99/365,SD ***-3-***
// EXTENT SYS003,volser,1,0,nnnn,120                ***-3-***
// LIBDEF CL,NEW=EZKEY,TEMP ***-4-***
// EXEC CORGZ,SIZE=1COK ***-4-***
NEWVOL CL=4(2) ***-4-***
/*
/&
// JOB EZKEY2                RESTORE EZ/KEY PHASE LIBRARY ***-1-***
// ASSGN SYS006,yyy          INPUT TAPE ***-5-***
// MTC REW,SYS006
// MTC FSF,SYS006,3          POSITION AT 2ND FILE (CORE IMAGE BACKUP)
// DLBL EZKEY,'your.EZKEY.phase.1library',,SD ***-3-***
// EXTENT SYS003,volser,1,0,nnnn,120                ***-3-***
// EXEC RESTORE ***-6-***
RESTORE CL,LIB=EZKEY(2) ***-6-***
/*
/&
// JOB EZKEY3                DEFINE VSAM DATA SETS ***-1-***
// DLBL IJVSUC,'your.VSAM.CATALOG',,VSAM ***-7-***
// EXTENT ,volser ***-7-***
// EXEC IDCAMS,SIZE=AUTO
DELETE your.SYSTEM.PIELIB ***-8-***
DEFINE CLUSTER -
    (NAME(your.SYSTEM.PIELIB) - ***-8-***
    VOLUMES(volser) RECORDS(1500 150) - ***-8-***
    CISZ(4096) SHR(2 3) -
    RECORDSIZE(4089 4089) NUMBERED)
DELETE your.EXAMPLE.PIELIB ***-9-***
DEFINE CLUSTER -
    (NAME(your.EXAMPLE.PIELIB) - ***-9-***
    VOLUMES(volser) RECORDS(400 100) - ***-9-***
    CISZ(2048) SHR(2 3) -
    RECORDSIZE(2041 2041) NUMBERED)

```

continued ...

... continued

```

DELETE your.USER.PIELIB                ***-10-***
DEFINE CLUSTER -
    (NAME(your.USER.PIELIB) -           ***-10-***
     VOLUMES(volser) RECORDS(nnn nnn) - ***-11-***
     CISZ(2048) SHR(2 3) -
     RECORDSIZE(2041 2041) NUMBERED)
DELETE your.RESTORE.PIELIB             ***-12-***
DEFINE CLUSTER -
    (NAME(your.RESTORE.PIELIB) -        ***-12-***
     VOLUMES(volser) RECORDS(nnn nnn) - ***-11-***
     CISZ(4096) SHR(2 3) -
     RECORDSIZE(4089 4089) NUMBERED)

/*
/&
// JOB EZKEY4                REPRO THE SYSTEM PIELIB        ***-1-***
// ASSGN SYS004,yyy          ASSIGN PRODUCT TAPE            ***-5-***
// MTC REW,SYS004            REWIND TAPE
// MTC FSF,SYS004,6          POSITION AT 3RD FILE ((3 - 1) * 3)
// TLBL INPUT,'FILE03'
// DLBL IJSYSUC,'your.VSAM.CATALOG',,VSAM                  ***-7-***
// EXTENT ,volser                                              ***-7-***
// DLBL EZKSYS,'your.SYSTEM.PIELIB',,VSAM                   ***-8-***
// EXTENT ,volser                                              ***-8-***
// EXEC IDCAMS,SIZE=AUTO
// REPRO INFILE(INPUT ENV(NOREWIND RECFM(F) BLKSZ(4089) PDEV(2400))) -
//   OUTFILE (EZKSYS)

/*
/&
// JOB EZKEY5                REPRO THE EXAMPLE PIELIB       ***-1-***
// ASSGN SYS004,yyy          ASSIGN PRODUCT TAPE            ***-5-***
// MTC REW,SYS004            REWIND TAPE
// MTC FSF,SYS004,9          POSITION AT 4TH FILE ((4 - 1) * 3)
// TLBL INPUT,'FILE04'
// DLBL IJSYSUC,'your.VSAM.CATALOG',,VSAM                  ***-7-***
// EXTENT ,volser                                              ***-7-***
// DLBL EZKXEMPL,'your.EXAMPLE.PIELIB',,VSAM               ***-9-***
// EXTENT ,volser                                              ***-9-***
// EXEC IDCAMS,SIZE=AUTO
// REPRO INFILE(INPUT ENV(NOREWIND RECFM(F) BLKSZ(2041) PDEV(2400))) -
//   OUTFILE (EZKXEMPL)

/*
/&

```

continued ...

```

... continued
// JOB EZKEY6          REPRO THE USER PIELIB          ***-1-***
// ASSGN SYS004.yyy    ASSIGN PRODUCT TAPE            ***-5-***
// MTC REW,SYS004      REWIND TAPE
// MTC FSF,SYS004,12   POSITION AT 5TH FILE ((5 - 1) * 3)
// TLBL INPUT,'FILE05'
// DLBL IJSYSUC,'your.VSAM.CATALOG',,VSAM             ***-7-***
// EXTENT ,volser     ***-7-***
// DLBL EZKUSR,'your.USER.PIELIB',,VSAM               ***-10-***
// EXTENT ,volser     ***-10-***
// EXEC IDCAMS,SIZE=AUTO
// REPRO INFILE(INPUT ENV(NOREWIND RECFM(F) BLKSZ(2041) PDEV(2400))) -
//   OUTFILE (EZKUSR)
/*
/&
* $$ EOJ

```

Tailoring CICS/DOS/VSE Installation JCL

At the right of some of the JCL lines shown above is the character string *****-n-*****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB name, CLASS and accounting information as needed.
2. Specify the address for SYS003, the EZ/KEY phase library.
3. Specify the name and location of the phase library for EZ/KEY.
4. For VSE/SP 2.1 and above only:
 - Delete the LIBDEF statement
 - Change CORGZ,SIZE=100K to LIBR
 - Replace NEWVOL CL=4(1) with:


```

          DEFINE LIB=EZ
          DEFINE SUBLIB=EZ.KEY
          
```
5. Specify the device address for SYS004, the EZ/KEY distribution tape.

6. For VSE/SP 2.1 and above only:
 - Change RESTORE to LIBR
 - Replace RESTORE CL,LIB=EZKEY(1) with:
RESTORE TAPE=SYS004,REPLACE=YES,OLDLIB=EZKEY:EZ.KEY
7. Specify the name and location of the VSAM user catalog to use for the EZ/KEY VSAM datasets.
8. Specify the name and location of the EZ/KEY SYSTEM PIELIB.
9. Specify the name and location of the EZ/KEY EXAMPLE PIELIB.
10. Specify the name and location of the EZ/KEY USER PIELIB.
11. Specify the number of primary and secondary records for the USER PIELIB and RESTORE PIELIB. A recommended minimum is 200 primary records for the USER PIELIB. Allow at least 100 secondary records per EZ/KEY user.
12. Specify the name and location of the EZ/KEY RESTORE PIELIB.

CICS/DOS/VSE Installation JCL Notes

<u>Job Name</u>	<u>Processing</u>
EZKEY1	Create a private core image library for EZ/KEY. This step is not required for installing EZ/KEY but is supplied in case you want to install EZ/KEY in a private library. If you install EZ/KEY this way, you must add this library to the search list in your CICS start-up deck.
EZKEY2	Restore file 2 of the distribution tape (phases for EZKYZ00, EZKYZ01, and EZKYLAM).

EZKEY3 Invoke IDCAMS to define the VSAM data spaces for the EXAMPLE, RESTORE, SYSTEM and USER PIELIBS.

Change the names on the DEFINE CLUSTER statements to names that comply with your site's conventions. Change the VOLUMES parameter to identify the correct volumes for your site. Do not modify any other parameters of the DEFINE CLUSTER statements for the EXAMPLE and SYSTEM PIELIBs.

On the DEFINE for the RESTORE and USER PIELIBS, you must specify the number of records that these files are to contain. You must specify at least 200 primary records for your USER PIELIB. See "Choosing the Size of PIELIBs" below. The number of RECORDS and the RECORDSIZE parameters for the RESTORE PIELIB should be at least as large as SYSTEM or USER PIELIB, whichever is larger.

NOTE: Do not DELETE and DEFINE the USER.PIELIB if you are re-installing EZ/KEY.

EZKEY4 Invoke IDCAMS to REPRO the SYSTEM.PIELIB from the distribution tape to your system.

EZKEY5 Invoke IDCAMS to REPRO the EXAMPLE.PIELIB from the distribution tape to your system.

EZKEY6 Invoke IDCAMS to REPRO the USER.PIELIB from the distribution tape to your system.

NOTE: Do not perform this step if you are re-installing EZ/KEY.

Tailoring CICS/DOS/VSE Start-Up JCL

Add the following JCL to your CICS start-up deck to identify the SYSTEM, USER, EXAMPLE, and RESTORE PIELIBs, and to identify the EZ/KEY phase library to CICS:

```
// DLBL EZKSYS,'prefix.SYSTEM.PIELIB',,VSAM
// EXTENT ,volser
// DLBL EZKUSR,'prefix.USER.PIELIB',,VSAM
// EXTENT ,volser
// DLBL EZXMPL,'prefix.EXAMPLE.PIELIB',,VSAM
// EXTENT ,volser
// DLBL EZKRSTR,'prefix.RESTORE.PIELIB',,VSAM
// EXTENT ,volser
```

Add the following JCL statements to your CICS start-up deck to identify the EZ/KEY phase library:

```
// DLBL EZKEY,'your.EZKEY.phase.library',,SD
// EXTENT ,volser
```

Add JCL statements to your CICS start-up deck to identify any additional datasets that you want EZ/KEY to use, such as PANVALET or VSAM macro libraries. For example:

```
// DLBL PANDD,'your.EZTPLUS.macrolib',,VSAM
// EXTENT ,volser
```

CHOOSING THE SIZE OF PIELIBS***Number of Blocks***

Select the number of blocks in the LIBSPACE (minimum of 200). This should be large enough for future expansion. A LIBSPACE cannot be expanded easily.

Size of Each Block

Enter the number of bytes for each block. EZ/KEY uses 10 bytes of each block for internal pointers. The blocksize should approximate the average size of members in the LIBSPACE. Large blocksize waste disk space if most members are small; small blocksize incur unnecessary I/O if most members are larger than one block.

For PIEBDAM, the blocksize must be the same value specified in the CICS FCT (File Control Table).

For PIEVSAM, the blocksize must be the CISIZE minus 7 bytes (for VSAM overhead). 505, 1017, 2041, or 4089 are good choices.

PIEBDAM PIELIBs on FBA devices should have blocksize equal to or an exact multiple of 512. PIEBDAM PIELIBs on non-FBA devices should have blocksize equal to or slightly less than one fourth of the DASD track length. For help in choosing an optimal blocksize, consult the IBM *Direct Access Storage* (GA26-1638) and *Reference Summary* (GX20-1983) publications.

Each record must be wholly contained in one block. The maximum recordsize is 10 characters less than the PIELIB blocksize (no spanned records).

Each block contains records from only one member. The LIBSPACE blocksize should be large enough to accommodate the largest record but small enough to avoid wasted space at the end of the block (505 - 4096 is the allowed range).

Each member must be wholly contained in one LIBSPACE. The size of the LIBSPACE (blocksize, number of blocks) must be large enough for the largest member.

One LIBSPACE must contain all members within a library.

With a blocksize of 1024, a 100-block LIBSPACE can hold 10 members with 200 80-byte records each (approximately 2,000 80-byte records or about 160,000 characters).

CICS/DOS/VSE TABLE CHANGES

You can use any system utility to copy the sample CICS table changes from file six of the distribution tape. You must incorporate the following changes into your CICS tables and reassemble your FCT, PCT, and PPT:

```

*
* PCT ENTRIES FOR EZ/KEY
*
      DFHPCT TYPE=ENTRY,PROGRAM=EZKYZOO,TRANSID=EZKE,           C
      SPURGE=NO,TPURGE=NO,SCRNSZ=ALTERNATE,                   C
      TWASIZE=42,CLASS=LONG,DTB=NO
      DFHPCT TYPE=ENTRY,PROGRAM=EZKYZO1,TRANSID=EZKT,          C
      SPURGE=NO,TPURGE=NO,TWASIZE=42
*
* ADDITIONAL PCT ENTRIES FOR EZ/KEY IF PANVALET IS USED
*
      DFHPCT TYPE=ENTRY,PROGRAM=PANENQ,TRANSID=PVEQ,           C
      SPURGE=YES,TPURGE=YES,SCRNSZ=ALTERNATE,                   C
      TWASIZE=256,CLASS=LONG
      DFHPCT TYPE=ENTRY,PROGRAM=PANDEQ,TRANSID=PVDQ,           C
      SPURGE=YES,TPURGE=YES,SCRNSZ=ALTERNATE,                   C
      TWASIZE=250,CLASS=LONG
*
* PPT ENTRIES FOR EZKEY
*
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYZOO,RES=YES
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYZO1
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYOPTN
*
* ADDITIONAL PPT ENTRIES FOR EZ/KEY IF PANVALET IS USED
*
      DFHPPT TYPE=ENTRY,PROGRAM=EZKYLAM
      DFHPPT TYPE=ENTRY,PROGRAM=FGPAN23
      DFHPPT TYPE=ENTRY,PROGRAM=PANDEQ
      DFHPPT TYPE=ENTRY,PROGRAM=PANENQ
      DFHPPT TYPE=ENTRY,PROGRAM=PANENQPM,USAGE=MAP
      DFHPPT TYPE=ENTRY,PROGRAM=PANMODK
      DFHPPT TYPE=ENTRY,PROGRAM=PANMPP
      DFHPPT TYPE=ENTRY,PROGRAM=PVCOMENQ

```

continued ...

... continued

```

*
* ADDITIONAL PPT ENTRIES FOR EZ/KEY IF IDMS/IDD INTERFACE IS USED
*
    DFHPPT TYPE=ENTRY,PROGRAM=EZKYIDMS
    DFHPPT TYPE=ENTRY,PROGRAM=IDMSINTC
*
* PLTPI ENTRY FOR EZ/KEY IF IDMS/IDD INTERFACE IS USED
*
    DFHPLT TYPE=ENTRY,PROGRAM=IDMSINTC
*
* ADDITIONAL PPT ENTRIES FOR EZ/KEY IF SQL/DS INTERFACE IS USED
*
    DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCF
    DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCI
    DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCR
    DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCS
    DFHPPT TYPE=ENTRY,PROGRAM=DQSEKCT
*
* FCT ENTRIES FOR EZ/KEY
*
    DFHFCT TYPE=DATASET,DATASET=EZKSYS,                                C
        ACCMETH=(VSAM,RRDS),                                          C
        SERVREQ=(GET,PUT,UPDATE,BROWSE,NEWREC),                      C
        STRNO=1,BUFND=2,LOG=NO
    DFHFCT TYPE=DATASET,DATASET=EZXEMPL,                                C
        ACCMETH=(VSAM,RRDS),                                          C
        SERVREQ=(GET,PUT,UPDATE,BROWSE,NEWREC),                      C
        STRNO=1,BUFND=2,LOG=NO
    DFHFCT TYPE=DATASET,DATASET=EZKUSR,                                C
        ACCMETH=(VSAM,RRDS),                                          C
        SERVREQ=(GET,PUT,UPDATE,BROWSE,NEWREC),                      C
        STRNO=1,BUFND=2,LOG=NO
    DFHFCT TYPE=DATASET,DATASET=EZKRSTR,                                C
        ACCMETH=(VSAM,RRDS),                                          C
        SERVREQ=(GET,PUT,UPDATE,BROWSE,NEWREC),                      C
        STRNO=1,BUFND=2,LOG=NO

```

If you specify any parameters not shown in the sample table entries, they should be the same as the IBM supplied default values for those parameters. Otherwise, the results may be unpredictable.

If storage is often overutilized in your CICS system, you may need to specify SPURGE=YES on the PCT entry for EZKE. However, if CICS purges EZ/KEY users, any members being edited by those users are not saved. If EZ/KEY users receive the message:

CICS/DOS/VSE Table Changes

DFH2004 TRANSACTION EZKE PURGED - SYSTEM UNDER STRESS
PLEASE RESUBMIT

the System Administrator must issue the EZKT transaction, and reply **YES** to the message:

EZKEYO25 - Type YES to remove EZ/KEY

before any users sign on to EZ/KEY again. Otherwise, users will receive the message:

EZKEYO07 - User already signed on to EZ/KEY

CICS 1.7 Considerations

If you are installing EZ/KEY in CICS 1.7, you may make the following changes to the FCT table entries for EZKSYS, EZKXMPL, EZKUSR, and EZKRSTR to avoid receiving any warning messages:

- Change:

ACCMETH=(VSAM,RRDS)

to:

ACCMETH=VSAM

- Change:

SERVREQ=(BROWSE,GET,NEWREC,PUT,UPDATE)

to:

SERVREQ=(BROWSE,READ,ADD,UPDATE)

Alternate Screen Size

EZ/KEY uses the alternate screen size defined in the TCT for 3270 terminals, if one is specified. You must ensure that the ALTSCRN= parameter of the DFHTCT macro (or ALTSCREEN for RDO) is correct for all of the terminals that you use with EZ/KEY. If this value is not specified correctly, only part of the screen is displayed, and it does not appear aligned properly. To correct this situation, change the ALTSCRN= parameter for the terminal to the correct size for that particular 3270 model. Alternatively, you can change SCRNSIZE=ALTERNATE to SCRNSIZE=DEFAULT on the PCT

entry for EZ/KEY, which forces EZ/KEY to use the default screen (DEFSCRN=) for all terminals using EZ/KEY, regardless of whether ALTSCRN= specified for those terminals.

Refer to the following IBM manuals for further information:

- *CICS/VS Resource Definition Guide* (SC33-0149)
- *CICS/VS Resource Definition Online* (SC33-0186)
- *CICS/VS Resource Definition (Macro)* (SC33-0237)

CICS Dynamic Transaction Backout

EZ/KEY supports the use of the CICS Dynamic Transaction Backout (DTB) Facility. To use this facility, change DTB=NO to DTB=YES and specify DTIMOUT=0030 (or larger) on the PCT entry for EZ/KEY.

EZ/KEY performs its own commit processing, and so there is *no* benefit if you specify LOG=YES. If you do specify LOG=YES with EZ/KEY, spurious ASCG abends may occur. These abends are due to deadlocks caused by enques issued automatically by the CICS File Control Program on behalf of each task that updates any of these files.

You must, therefore, specify LOG=NO for all FCT entries used with EZ/KEY.

If you use Resource Definition Online (RDO) to define the transactions and programs for EZ/KEY with the CEDA transaction, there is no way to specify DTB=NO. When RDO is installed, all transactions run with DTB=YES. Therefore, you must specify LOG=NO for all FCT entries used with EZ/KEY.

CICS Multi-Region Option (MRO)

EZ/KEY supports the use of MRO to provide virtual storage constraint relief.

To use MRO with EZ/KEY, change DFHPCT TYPE=ENTRY to TYPE=REMOTE for the primary CICS region (the one that has most of the terminals) and add SYSIDNT=xxxx to identify the remote CICS region in which EZ/KEY executes. Install only the DFHPCT entry in the primary CICS region. Do *not* install the remaining table entries for EZ/KEY in the primary CICS region. Install all table entries (including DFHPCT) in the remote region. This allows EZ/KEY to use the Transaction Routing feature of MRO.

Refer to the IBM *CICS/DOS/VS Intercommunication Facilities Guide* |
(SC33-0133) for further information.

CICS RUNAWAY TASK INTERVAL

EZ/KEY runs as either a *conversational* or *pseudo-conversational* transaction under CICS. Some activities, such as error-checking a large EASYTRIEVE PLUS program, can require more CPU cycles than other CICS applications.

To prevent spurious AICA abends, you may need to increase your Runaway Task Interval. To find out what the current setting is, enter the transaction:

CEMT INQUIRE RUNAWAY

The following screen is displayed:

INQUIRE RUNAWAY	
STATUS: RESULTS - OVERTYPE TO MODIFY	
MAxtasks(020)	
AMxtasks(010)	
BMAxtasks()	NOT IN SYSTEM
ATp()	NOT IN SYSTEM
AKp()	NOT IN SYSTEM
CUshion(016384)	
IOcp()	NOT ON DOS
Time(005000)	
RUnaway(0005000)	
STall(020000)	
SWt(0040)	
APPLID=CICS	
RESPONSE: NORMAL	TIME: 16.40.35 DATE: 85.037
PF: 1 HELP 3 END	7 SBH 8 SFH 9 MSG 10 SB 11 SF

The default value for CICS (and for this example) is 5000 milliseconds or five seconds. If your current setting is less than 5000 and you are experiencing spurious AICA abends while running EZ/KEY, you should increase your setting's value to at least 5000. To change the value for the duration of this CICS job, overtype the RUnaway(value) with a new value, such as 0005000, for 5 seconds.

CICS Runaway Task Interval

For more information on using the CEMT command, refer to the IBM *CICS/VS Operator's Guide* (SC33-0080).

The value recommended for EZ/KEY is 5 seconds (the IBM-supplied default).

Once you have decided on a Runaway Task Interval suitable for your environment, you may specify it in the CICS System Initialization Table (SIT) using the DFHSIT macro parameter ICVR=nnnnnn.

For more information on specifying the Runaway Task Interval in the SIT, refer to the IBM *CICS/DOS/VS System Programmers Reference Manual* (SC33-0070).

CICS/DOS/VSE BDAM PIELIBS

If you want to use BDAM as the access method for one or more PIELIBs, you must specify RELTYPE=HEX in the FCT entry for each BDAM PIELIB, as shown below:

```
DFHFCT TYPE=DATASET,DATASET=fctname,ACCMETH=BDAM,RELTYPE=HEX,  
      RECFORM=(FIXED,UNBLOCKED),LRECL=nnnn,BLKSIZE=nnnn,  
      SERVREQ=(BROWSE,GET,NEWREC,PUT,UPDATE),LOG=YES
```

ICCF CONSIDERATIONS FOR CICS/DOS/VSE

EZ/KEY can operate as a transaction under CICS in an ICCF partition. Check the ICCF DTSOPTNS macro for the specification of the PARTN= parameter. This defines the size of each ICCF partition. Partition number 0 is the CICS partition. This must be increased as explained above under Memory Requirements.

You should check the following:

- ICCF partition size (VSE ALLOC value)
- CICS partition size within ICCF partition (PARTN=)
- Size of the GETVIS area
- Size of the dynamic storage area
- EXECute statement size in the ICCF/CICS startup JCL.

Adjust the size of the VSE partition, the CICS partition within ICCF, or the SIZE= parameter on the EXEC statement in the ICCF/CICS start-up JCL to obtain the amount of CICS Dynamic Storage Area needed.

Refer to the following IBM manuals for more information:

- *VSE/ICCF Installation and Operations Reference* (SC33-6062). See the installation step which describes defining ICCF options.
- *CICS/DOS/VS System Programmer's Guide* (SC33-0070). See the overview of CICS operations.

VALIDATE CICS INSTALLATION

To verify the correct installation of EZ/KEY, type in the following CICS transaction:

ezkey system

and press ENTER.

A panel should be displayed, with the name PROSTART in the upper-left corner, as illustrated below:

```
PROSTART ----- Primary Selection Menu ----- EZ/KEY
COMMAND ==>                                     USERID: userid
                                                TIME  : 10:31:51
                                                DATE   : 05/12/88

Select Option ==>

      0  Program Function Key Settings
      1  Edit an EASYTRIEVE PLUS Program
      2  Edit any Member
      3  Library and System Utilities
      4  Run a Program On-Line
      5  Submit a Program for Batch Processing
      6  Report Processing Facility

      T  Tutorial for EZ/KEY

      X  Exit the EZ/KEY System

EZ/KEY, Version X.X
Copyright (c) 1983, 1988 Pansophic Systems, Inc.
```

Immediately after validating the EZ/KEY installation, perform the procedures described in Section 4 of the *EZ/KEY Administrator's Guide*.

*EASYTRIEVE PLUS VSAM Macro Libraries for CICS/DOS/VSE***EASYTRIEVE PLUS VSAM MACRO LIBRARIES FOR CICS/DOS/VSE**

EZ/KEY can read members from one or more EASYTRIEVE PLUS VSAM Macro libraries. For each library, a DLBL statement similar to:

```
// DLBL PANDD, 'your.EZTMACRO.library', ,VSAM
```

must be in the CICS startup JCL. Each macro library must be CONNECTed and ENABLEd to EZ/KEY.

Additional FCT entries are required since CICS File Control OPENs, READs, and CLOSEs are used to access EASYTRIEVE PLUS Macro libraries. Sample FCT entries are shown below:

DFHFCT	TYPE=DATASET,DATASET=PANDD1,SERVREQ=(BROWSE,GET),	C
	BUFND=2,BUFNI=1,STRNO=1,ACCMETH=(VSAM,KSDS)	
DFHFCT	TYPE=DATASET,DATASET=PANDD2,SERVREQ=(BROWSE,GET),	C
	BUFND=2,BUFNI=1,STRNO=1,ACCMETH=(VSAM,KSDS)	
DFHFCT	TYPE=DATASET,DATASET=PANDD3,SERVREQ=(BROWSE,GET),	C
	BUFND=2,BUFNI=1,STRNO=1,ACCMETH=(VSAM,KSDS)	

As explained in the *EASYTRIEVE PLUS Installation Guide*, to use VSAM as the EASYTRIEVE PLUS macro library, specify MACRO=VSAM as an operand of the EZTPOPT options macro.

The System Administrator must use the External LIBSPACE Utility (option 3.a.4) to CONNECT each EASYTRIEVE PLUS Macro library to EZ/KEY.

*PANVALET Considerations for CICS/DOS/VSE***PANVALET CONSIDERATIONS FOR CICS/DOS/VSE**

EZ/KEY can read members from one or more PANVALET Release 11.0 or above libraries. For each library, a DLBL statement similar to:

```
// DLBL PANDD1,'your.PANVALET.library',,DA
```

must be in the CICS startup JCL, and each library must be CONNECTed and ENABLEd to EZ/KEY.

FCT entries are not required since DOS OPENs, READs, and CLOSEs (with CICS WAITs) are used to access PANVALET libraries.

The additional PPT entries shown under the topic "CICS/DOS/VSE Table Changes" must be made to your CICS tables.

As explained in the *EASYTRIEVE PLUS Installation Guide*, to use PANVALET as the EASYTRIEVE PLUS macro library, specify MACRO=PAN as an operand of the EZTPOPT options macro. PANDD, the default value of the MACDDN option of EZTPOPT, is the prefix of DLBL names for EASYTRIEVE PLUS' macro library. Multiple PANVALET libraries can be used in EASYTRIEVE PLUS; the DLBL names for the PANVALET libraries must be PANDD1, PANDD2, ..., PANDDn, where *n* is the value of MAC#LIB on the EZTPOPT macro. PANDD1 should be used as the DDname and External LIBSPACE Name for the PANVALET library to comply with the naming conventions of EASYTRIEVE PLUS.

The System Administrator must use the External LIBSPACE Utility (option 3.a.4) to CONNECT each PANVALET library to EZ/KEY.

When the first PANVALET library is CONNECTed to EZ/KEY, a LOAD is issued for the module EZKYLAM, the access method for PANVALET (this module is distributed and installed as part of the EZ/KEY system for CICS). EZKYLAM issues a CICS LOAD for FGPAN23, the options module for PANVALET. FGPAN23 is a component of your PANVALET system; it is produced by assembling and link-editing the PVOPT macro. FGPAN23 must be located in the DFHRPL concatenation of core image libraries.

See the "Reconnect Procedures" and "Reinstallation Procedures" topics earlier in this section for further instructions.

PANVALET Security for CICS/DOS/VSE

There are no special EZ/KEY considerations when you do not use any of the PANVALET security features.

PANVALET Considerations for CICS/DOS/VSE

If the PANVALET library is protected with a library control code and the ++PRINT or ++WRITE command is suppressed for this library, you must specify this control code as the External Password. The control code should be the same as the code specified on the ++CONTROL statement in batch PANVALET.

If a PANVALET member that has an access code is invoked as a macro in an EASYTRIEVE PLUS program, you must precede the %macro-name statement by an ACCESS 'access-code' statement in the program.

To edit an access-protected member, use the Copy/Move Utility (option 3.3) to copy the member from the PANVALET library to your PIELIB. Specify the member access code as the Password of the FROM: member.

During an edit session, you can retrieve an access-protected PANVALET member by entering **COPY** (with no operands) on the COMMAND line. This invokes the COPY/MOVE Utility panel on which you can specify the member to be retrieved and its access code.

PANVALET/CICS Memory Requirements

If PANVALET is used with EZ/KEY, allow an additional 85K for the PANVALET module, 10K for each connected PANVALET library, and 10K for each enabled PANVALET library.

EZKYLAM module	85K
FGPAN23 module	1K
Additional PANVALET modules	200K
Each CONNECTed Library	10K
Each ENABLEed Library	10K per user

CICS/DOS/VSE PANVALET Version 12.0 or Above Considerations

If you are using PANVALET Version 12.0 or above, use the System Tuning Parameter panel (PROULTPS) to inform EZ/KEY to use the PANVALET Interface module, PANMODK, to access PANVALET libraries instead of EZKYLAM.

Type **3.a.7** on the PROSTART panel to obtain the System Tuning Parameter panel (PROULTPS). On PROULTPS, type **PANMODK** in the PANVALET Module Name field.

PANVALET Considerations for CICS/DOS/VSE

So that EZ/KEY can locate the PANMODK module, the PANVALET load library must be added to the DFHRPL concatenation of load libraries.

The following PPT entry must be added:

```
DFHPPT TYPE=ENTRY,PROGRAM=PANMODK
```

EZ/KEY must execute in Conversational Mode to use PANMODK with EZ/KEY. Before attempting to CONNECT to any PANVALET libraries, and before executing the CICS Reconnect Procedures, sign on as **SYSTEM** and change the **Execute Conversational** System Tuning parameter to **YES**. This parameter indicates that all users execute in Conversational Mode.

MULTIPLE EZ/KEY SYSTEMS UNDER ONE CICS

Different departments have differing security needs. Some departments need complete isolation from users in other departments.

To provide increased security in a CICS environment, several distinct transaction IDs for EZ/KEY can be run under the same CICS system at the same time. These EZ/KEY systems are completely isolated from each other. Each has its own copy of the SYSTEM and EXAMPLE PIELIBs, and one or more USER PIELIBs.

EZ/KEY is normally installed in CICS with a transaction ID of EZKE. To allow multiple EZ/KEY systems to run under a single CICS, EZ/KEY:

- Prefixes all CICS ENQ/DEQ Resource names with the 4-character CICS Transaction Identifier (TRANSID)
- Prefixes its CICS Temporary Storage name with the TRANSID
- Uses the first 3 characters of the CICS TRANSID as the first 3 characters of the FCTnames for SYSTEM (EZKSYS) and EXAMPLE (EZKXMPL) PIELIBs.
- Uses the first 3 characters of the CICS TRANSID as the first 3 characters of the EZ/KEY Options Module (EZKYOPTN) in the PPT.
- The first 3 characters of the CICS TRANSID for program EZKYZ01 must be the same as the first 3 characters of the CICS TRANSID for EZKYZ00.

Suppose you have three departments: Accounting, Distribution and Manufacturing. You can set up three different TRANSIDs. For example: EZAC for Accounting, EZDI for Distribution, and EZMA for Manufacturing. The removal transactions may be called EZAT, EZDT, and EZMT, respectively. The TRANSIDs must be unique within the first three characters. All three transactions can use the same EZKYZ00, EZKYZ01, and EZKYLAM, since the EZ/KEY code is fully reentrant.

Separate FCT entries must be made for EZASYS, EZDSYS, EZMSYS, EZAXMPL, EZDXMPL, and EZMXMPL. Separate VSAM files must be allocated for each of these. Initially, the three sets of SYSTEM and EXAMPLE PIELIBs contain identical data from the Installation Tape.

Multiple EZ/KEY Systems Under One CICS

Separate FCT entries must be made for EZAUSR, EZDUSR, and EZMUSR for the respective USER PIELIBs for each department (or you may use names such as ACCTNG, DISTRIB, and MNFCTNG). There must be a unique FCT entry and a corresponding VSAM dataset to contain each department's USER PIELIB.

You should assign one person as the EZ/KEY System Administrator for each of the departments. You may use ordinary CICS operator sign-on security (or some other CICS security package) to control access to the various EZ/KEY transactions.

The EZ/KEY System Administrator for each department should sign on with the userid SYSTEM to initialize, format, and connect the USER PIELIBs. For example, one would sign on as EZKA SYSTEM, while another would use EZKD SYSTEM. Refer to the *EZ/KEY Administrator's Guide* for further information.

Once the separate EZ/KEY systems are established, each department is off and running. Each System Administrator can create new users, add new userids and libraries, and perform other maintenance functions, independent of any other departments. For example, two different departments isolated as described above might each have a userid called SMITH.

The only time that you need to be involved is when one of the System Administrators wants to add a new PIELIB or macro library. A new PIELIB requires an additional FCT entry. A new external library (such as PANVALET) requires additional JCL to be added to the CICS start-up deck. An EASYTRIEVE PLUS VSAM macro library requires both an additional FCT entry and JCL in the CICS start-up deck.

One note of caution is in order. If more than one of these EZ/KEY systems connect up to a single physical PIELIB, then only one of them may specify WRITE or CREATE access. All others must specify READ access. Otherwise, the integrity of the PIELIB is NOT guaranteed, and most likely will be corrupted.

SQL INTERFACE OPTION INSTALLATION

The PANSOPHIC SQL Interface to SQL/DS allows EZ/KEY users to syntax-check SQL commands embedded in their EASYTRIEVE PLUS programs. The installation of the EZ/KEY SQL Interface transfers the interface from the installation tape to your operating system.

This section describes the installation procedure and the step-by-step installation sequence.

SQL Operating Environments

The EZ/KEY SQL Interface operates on all systems that support IBM'S SQL/DS or DB2 systems.

The EZ/KEY SQL Interface requires the following disk storage:

VSE/SP 2.1

500K byte library with:

300K bytes used for PHASE executable code and
200K bytes used for OBJ relocatable code.

DOS/VSE Pre 2.1

300K byte Core image library
200K byte relocatable library.

EZ/KEY SQL Interface Virtual Memory Requirements

EZ/KEY uses approximately 100K of additional memory whenever it encounters an SQL statement. It uses the memory to bring the appropriate SQL Interface module into virtual storage and LINK to it. Depending on how many EZ/KEY users are active concurrently, the amount of memory used may actually be more than 100K at any one instant.

Each SQL statement requires approximately 100 bytes of additional dynamic storage for work areas.

Due to the additional storage requirements of the SQL Interface, you may need to increase the size of your CICS partition by at least 512K in addition to the increases already required for normal EZ/KEY installation.

SQL Interface Installation

The EZ/KEY SQL Interface should be installed in the same phase or core image library as EZ/KEY. The size of this library must be increased by 500K to accommodate the SQL Interface modules.

The following messages are normal and expected when link-editing the SQL Interface modules (DQSEKCF, DQSEKCI, DQSEKCR, DQSEKCS, and DQSEKCT) for CICS/DOS/VSE:

UNRESOLVED EXTERNAL REFERENCES

WXTRN

DFHAICBA

SQL INTERFACE INSTALLATION – VSE/SP 2.1

NOTE: The next several pages describe the SQL Interface installation procedures for VSE/SP 2.1 systems. If you are using a DOS/VSE (pre-SP 2.1) system, do not use these instructions. Instead, see "SQL Interface Installation – DOS/VSE (pre-SP 2.1) Systems" elsewhere in this section.

File 8 on the installation tape contains the JCL you need to install the SQL Interface on VSE/SP 2.1 systems. A copy of file 8 is shown below.

There are two sets of notes following the JCL. The first, "VSE/SP 2.1 SQL Installation JCL Notes," gives brief descriptions of the jobs and steps in the JCL. The second set of notes, "Tailoring VSE/SP 2.1 Installation JCL," describes the changes you must make to the JCL before using it.

VSE/SP 2.1 JCL (File 8)

```

* $$ JOB JNM=SQL,CLASS=0,DISP=D
// JOB PRECOMP1
// OPTION NODUMP,LOG
// DLBL IJSPH, 'sql processed output',0
// EXTENT SYSPCH,volser,1,0,start,lgth
ASSGN SYSPCH,DISK,VOL=volser,SHR
// EXEC PGM=ARIPRPA,SIZE=AUTO,
      PARM='USERID=userid/password,
      PREPNAME=DQSMMTB,KEEP,NOPRINT,ISOLATION(CS)'
*      DATA SET Q010BMMTB1 AT LEVEL 001 AS OF 09/17/87
*****
*
* THIS PROGRAM READS THE IBM SQL/DS HELP TEXT TABLES, EXTRACTS
...
MESSAGE DS OH
      EXEC SQL
            DECLARE C1 CURSOR FOR
            SELECT ITEM, "SQL/DS HELP"
            FROM SQLDBA.SYSTXT2
            WHERE ITEM >= :WKITEM
...
WKITEM DC H'19099'
USERID DC CL8'SQLDBA '
PASSWORD DC CL8'SQLDBA '
      DC C'CMDMSG'

```

continued ...

... continued

```

/*
/&
// JOB ASSEMBL1
CLOSE SYSPCH,cuu
// DLBL IJSYSIN,'sql processed output',0
// EXTENT SYSIPT,volser,1,0,start,lgth
ASSGN SYSIPT,DISK,VOL=volser,SHR

// OPTION CATAL,SYSPARM='VSE21'
  PHASE DQSMMTB,S
// EXEC ASSEMBLY,SIZE=256K
/*
// DLBL  SQLLIB,'your.ezkey.library'
// EXTENT ,volser
// LIBDEF PHASE,CATALOG=SQLLIB:sublib
// DLBL  IBMLIB,'your.ibm.sql.object.library'
// EXTENT ,volser
// LIBDEF *,SEARCH=(SQLLIB.sublib,IBMLIB.sublib)
  INCLUDE ARIPRDID
// EXEC LNKEDT,SIZE=256K
/*
/&
// JOB CLOSE1
CLOSE SYSIPT,cuu
/*
/&
// JOB DQSMMTB
// DLBL  SQLLIB,'your.ezkey.library'
// EXTENT ,volser
// DLBL IJSYSPH,'ibm help text2 output',0
// EXTENT SYSPCH,volser,1,0,start,lgth
ASSGN SYSPCH,DISK,VOL=volser,SHR
// LIBDEF *,SEARCH=SQLLIB.sublib
// EXEC PGM=DQSMMTB,SIZE=256K
/*
/&
// JOB CLOSE2
CLOSE SYSPCH,cuu
/*
/&

```

```

***-1-***
***-4-***
***-5-***
***-5-***
***-5-***

```

```

***-6-***
***-6-***
***-6-***
***-7-***
***-7-***
***-8-***

```

```

***-1-***
***-9-***

```

```

***-1-***
***-10-***
***-10-***
***-11-***
***-11-***
***-11-***
***-12-***

```

```

***-1-***
***-13-***

```

continued ...

... continued

```

// JOB ASSEMBL2                                     ***-1-***
// DLBL IUSYSIN,'ibm help text2 output',0           ***-14-***
// EXTENT SYSIPT,volser,1,0,start,lgth              ***-14-***
ASSGN SYSIPT,DISK,VOL=volser,SHR                     ***-14-***
// DLBL IUSYSPH,'text2 object file',0               ***-15-***
// EXTENT SYSPCH,volser,1,0,start,lgth              ***-15-***
ASSGN SYSPCH,DISK,VOL=volser,SHR                     ***-15-***
// OPTION DECK
// EXEC ASSEMBLY,SIZE=256K
/*
/&
// JOB CATAL2                                     ***-1-***
CLOSE SYSIPT,cuu                                     ***-16-***
CLOSE SYSPCH,cuu                                     ***-16-***
// DLBL SQLLIB,'your.ezkey.library'                 ***-17-***
// EXTENT ,volser                                    ***-17-***
// DLBL IUSYSIN,'text2 object file',0               ***-18-***
// EXTENT SYSIPT,volser,1,0,start,lgth              ***-18-***
ASSGN SYSIPT,DISK,VOL=volser,SHR                     ***-18-***
// EXEC LIBR,SIZE=256K,PARM='ACCESS SUBLIB=SQLLIB.sublib' ***-19-***
/*
/&
// JOB CLOSE3                                     ***-1-***
CLOSE SYSIPT,cuu                                     ***-20-***
/*
/&
// JOB LIBR                                     ***-1-***
// DLBL SQLLIB,'your.ezkey.library'                 ***-21-***
// EXTENT ,volser                                    ***-21-***
// EXEC LIBR,SIZE=256K,PARM='ACCESS SUBLIB=SQLLIB.sublib' ***-21-***
    CATALOG OM#EKCIS.OBJ      REPLACE=YES
ESD          DQSEKCI          -DFHEAIO          DQ#EKCI
ESD          DQ#EKCI          -  QXGSTKFST        XFSTKFST
...
...
/*
/&

```

continued ...

SQL Interface Installation VSE/SP 2.1

... continued

```

// JOB PRECOMP2                                     ***-1-***
// OPTION NODUMP,LOG
// DLBL IJSPH, 'sql processed output',0             ***-22-***
// EXTENT SYSPCH,volser,1,0,start,lgth              ***-22-***
ASSGN SYSPCH,DISK,VOL=volser,SHR                    ***-22-***
// EXEC PGM=ARIPRPA,SIZE=AUTO,                      X
                PARM='USERID=userid/password,       ***-23-*** X
                PREPNAME=DQEKCMD,KEEP,NOPRINT,ISOLATION(CS)'
PUNCH ' CATALOG OMSMCMDC1.OBJ REPLACE=YES'
*      DATA SET QO1OBMCMDC1 AT LEVEL 001 AS OF 09/17/87
*****
*
* THIS MODULE PROCESSES ALL SQL COMMANDS THAT ARE VALID FOR BOTH
* THE COMPILE AND EXECUTION PHASE. THE CALLING MODULE MUST CREATE
...
MESSAGE DS      OH
        EXEC SQL
                DECLARE C1 CURSOR FOR
                SELECT "SQL/DS HELP"
                FROM SQLDBA.SYSTEXT1      TEXT1,
                SQLDBA.SYSTEXT2      TEXT2
                WHERE TOPIC = :CMDMSTPC AND
                TEXT1.ITEM = TEXT2.ITEM
...
/*
/&
// JOB ASSEMBL3                                     ***-1-***
CLOSE SYSPCH,uuu                                   ***-24-***
// DLBL IJSPH, 'sql processed output',0             ***-25-***
// EXTENT SYSPCH,volser,1,0,start,lgth              ***-25-***
ASSGN SYSPCH,DISK,VOL=volser,SHR                    ***-25-***
// DLBL IJSPH, 'assembler object file',0            ***-26-***
// EXTENT SYSPCH,volser,1,0,start,lgth              ***-26-***
ASSGN SYSPCH,DISK,VOL=volser,SHR                    ***-26-***
// OPTION DECK
// EXEC ASSEMBLY,SIZE=256K
/*
/&

```

continued ...

... continued

```

// JOB CATAL3                                     ***-1***
CLOSE SYSIPT,cuu                                  ***-27-***
CLOSE SYSPCH,cuu                                  ***-27-***
// DLBL   SQLLIB,'your.ezkey.library'             ***-28-***
// EXTENT ,volser                                 ***-28-***
// DLBL IJSYSIN,'assembler object file',0         ***-29-***
// EXTENT SYSIPT,volser,1,0,start,lgth            ***-29-***
ASSGN SYSIPT,DISK,VOL=volser,SHR                  ***-29-***
// EXEC   LIBR,SIZE=256K,PARM='ACCESS SUBLIB=SQLLIB.sublib' ***-30-***
/*
/&
// JOB CLOSE4                                     ***-1***
CLOSE SYSIPT,cuu                                  ***-31-***
// EXEC PROC=ARISLIBS
// EXEC PROC=ARISLIBP
// EXEC PGM=ARIDBS,SIZE=AUTO
CONNECT userid IDENTIFIED BY password;           ***-32-***
GRANT RUN ON userid.DQEKCMD TO PUBLIC;            ***-32-***
/*
/&
// JOB SQLLNK                                     ***-1-***
// DLBL   SQLLIB,'your.ezkey.library'             ***-33-***
// EXTENT ,volser                                 ***-33-***
// LIBDEF PHASE,CATALOG=SQLLIB.sublib             ***-33-***
// DLBL   IBMLIB,'your.ibm.sql.object.library'    ***-34-***
// EXTENT ,volser                                 ***-34-***
// LIBDEF OBJ,SEARCH=(SQLLIB.sublib,IBMLIB.sublib) ***-35-***
// OPTION LOG,CATAL
* ACTION AUTO
  PHASE DQSEKCI,*
  INCLUDE DFHEAI
...
// EXEC LNKEDT
/*
/&
* $$ EDJ

```

VSE/SP 2.1 SQL Installation JCL Notes

1. Job PRECOMP1 preprocesses the SQL/DS interface module, DQSMMTB.
2. Job ASSEMBL1 assembles the output of the preprocessor.
3. Job CLOSE1 reassigns SYSIPT back to its original address.
4. Job DQSMMTB executes DQSMMTB.
5. Job CLOSE2 reassigns SYSPCH back to its original address.
6. Job ASSEMBL2 assembles the IBM message text data.
7. Job CATAL2 places the assembler object into an object library.
8. Job CLOSE3 reassigns SYSIPT back to its original address.
9. Job LIBR places the SQL Interface modules into an object library.
10. Job PRECOMP2 preprocesses the SQL command processor.
11. Job ASSEMBL3 assembles the SQL command processor.
12. Job CATAL3 places the assembler object file into an object library.
13. Job CLOSE4 reassigns SYSIPT back to its original address.
14. Job SQLLNK links the Pansophic SQL Interface.

Tailoring VSE/SP 2.1 SQL Installation JCL

At the right of some of the JCL lines shown above is the character string `***-n-***`, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Change the JOB statement to conform to your installation's standards.
2. Change the dataset name, volser, and starting and ending track addresses on the IJSYSPH, SYSPCH, and ASSGN statements to identify an output disk dataset. A minimum of 40 3350 tracks is recommended for this dataset.
3. Change the userid and password for the SQLDBA UserID.
 - a. As distributed, the DQSMMTB program assumes the name of the SQL/DS Help Table is SQLDBA.SYSTEXT2. If your site changed the name of this table during installation of SQL/DS, you must supply the name you use at the location in the job stream marked by `***-a-***`. Note that `***-a-***` appears only in this documentation, not in the file you are working with.
 - b. Change the userid and password for the SQLDBA UserID as you did for step 3. Note that `***-b-***` appears only in this documentation, not the file you are working with.
4. Change cuu on the CLOSE statement to point back to the original address for SYSPCH.
5. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 2.
6. Change the dataset name, volser, and sublib to identify your EZ/KEY library and sublib.
7. Change the dataset name and volser on the IBMLIB DLBL statement to identify the object library that contains the IBM object modules for SQL/DS.
8. Change sublib to identify the EZ/KEY sublib and the SQL/DS sublib.
9. Change cuu on the CLOSE statement to point back to the original address for SYSIPT.

10. Change the dataset name and volser to identify your EZ/KEY library.
11. Change the dataset name, volser, and starting and ending track addresses on the IJSYSPH, SYSPCH, and ASSGN statements to identify an output disk dataset. A minimum of 37 3350 tracks is recommended for this dataset.
12. Change sublib to identify the EZ/KEY sublib.
13. Change cuu on the CLOSE statement to point back to the original address for SYSPCH.
14. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 11.
15. Change the dataset name, volser, and starting and ending track addresses on the IJSYSPH, SYSPCH, and ASSGN statements to identify an output disk dataset. A minimum of 37 3350 tracks is recommended for this dataset.
16. Change cuu on the CLOSE statements to point back to the original addresses for SYSIPT and SYSPCH.
17. Change the dataset name and volser to identify your EZ/KEY library and sublib.
18. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 15.
19. Change sublib to identify the EZ/KEY sublib.
20. Change cuu on the CLOSE statement to point back to the original address for SYSIPT.
21. Change the dataset name, volser, and sublib to identify your EZ/KEY library and sublib.
22. Change the dataset name, volser, and starting and ending track addresses on the IJSYSPH, SYSPCH, and ASSGN statements to identify the disk dataset from step 2.
23. Change the userid and password for the SQLDBA UserID as you did in step 3.

SQL Interface Installation VSE/SP 2.1

- c. The command processor assumes that the names of the SQL/DS Help Tables are SQLDBA.SYSTEXT1 and SQLDBA.SYSTEXT2. If your site changed the names of these tables during installation of SQL/DS, you must supply the names you use at the location in the job stream marked by `***-c-***`. Note that `***-c-***` appears only in this documentation, not in the file you are working with.
24. Change cuu on the CLOSE statement to point back to the original address for SYSPCH.
25. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 22.
26. Change the volser, and starting and ending track addresses on the IJSYSPH and SYSPCH statements to identify a disk dataset to contain the output object from the assembler. A minimum of 27 3350 tracks is recommended for this dataset.
27. Change cuu on the CLOSE statements to point back to the original addresses for SYSIPT and SYSPCH.
28. Change the dataset name and volser to identify your EZ/KEY library.
29. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 26.
30. Change sublib to identify the EZ/KEY sublib.
31. Change cuu on the CLOSE statement to point back to the original address for SYSIPT.
32. Change the userid and password for the SQLDBA UserID as you did in step 23.
33. Change the dataset name, volser, and sublib to identify your EZ/KEY library and sublib.
34. Change the dataset name on the IBMLIB DLBL statement to identify the object library that contains the IBM object modules for SQL/DS. Change the volser on the EXTENT statement to identify the correct volume.
35. Change sublib to identify the EZ/KEY sublib and the SQL/DS sublib.

*SQL Interface Installation - DOS/VSE (pre-SP 2.1)***SQL INTERFACE INSTALLATION - DOS/VSE (PRE-SP 2.1)**

NOTE: The next several pages describe the SQL Interface installation procedures for DOS/VSE (pre-SP 2.1) systems. If you are using a VSE/SP 2.1 system, do not use these instructions. Instead, see "SQL Interface Installation - VSE/SP 2.1 Systems" elsewhere in this section.

File 9 on the installation tape contains the JCL you need to install the SQL Interface on DOS/VSE (pre-SP2.1) systems. A copy of file 9 is shown below.

There are two sets of notes following the JCL. The first, "DOS/VSE (pre-SP 2.1) SQL Installation JCL Notes," gives brief descriptions of the jobs and steps in the JCL. The second set of notes, "Tailoring DOS/VSE (pre-SP 2.1) Installation JCL," describes the changes you must make to the JCL before using it.

Pre-SP 2.1 JCL (File 9)

```

* $$ JOB JNM=SQL,CLASS=0,DISP=D          ***-1-***
// JOB      CREATE                      ***-1-***
// OPTION NODUMP
// ASSGN  SYS004,DISK,VOL=volser,SHR      ***-2-***
// DLBL   SQLRLIB,'your.sql.relo.library',0,SD ***-2-***
// EXTENT SYS004,volser,,start,lgth      ***-2-***
// LIBDEF RL,NEW=SQLRLIB,TEMP
// EXEC   CORGZ,SIZE=100K
        NEWVOL RL=cylinder(tracks)      ***-3-***
/*
/&
// JOB PRECOMP1                          ***-1-***
// OPTION NODUMP,LOG
// DLBL  IBMLIB,'your.ibm.sql.core.library' ***-4-***
// EXTENT ,volser                        ***-4-***
// LIBDEF CL,SEARCH=IBMLIB
// DLBL  IUSYSPH,'sql processed output',0 ***-5-***
// EXTENT SYSPCH,volser,1,0,start,lgth   ***-5-***
ASSGN SYSPCH,DISK,VOL=volser,SHR        ***-5-***
// EXEC  PGM=ARIPRPA,SIZE=AUTO,
        PARM='USERID=userld/password,    ***-6-*** X
        PREPNAME=DQSMMTB,KEEP,NOPRINT,ISOLATION(CS) ' X

```

continued ...

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

... continued

```

*****
*
*   THIS PROGRAM READS THE IBM SQL/DS HELP TEXT TABLES, EXTRACTS
*   ...
MESSAGE  DS      OH
        EXEC SQL
                DECLARE C1 CURSOR FOR
                SELECT ITEM, "SQL/DS HELP"
                FROM   SQLDBA.SYSTEXT2
                WHERE  ITEM >= :WKITEM
                *****-a-*****
                X
                X
                X
                X
        ...
WKITEM   DC      H'19099'
USERID   DC      CL8'SQLDBA '
PASSWORD DC      CL8'SQLDBAPW'
        DC      C'CMDMSG'
                *****-b-*****
                *****-b-*****
        ...
/*
/&
// JOB ASSEMBL1
CLOSE SYSPCH, cuu
// DLBL IJSYSIN, 'sql processed output', 0
// EXTENT SYSIPT, volser, 1, 0, start, lgth
ASSGN SYSIPT, DISK, VOL=volser, SHR
// OPTION CATAL, SYSPARM='VSE13'
        PHASE DQSMMTB
// EXEC ASSEMBLY, SIZE=256K
/*
// DLBL   SQLRLIB, 'your.sql.relo.library'
// EXTENT ,volser
// DLBL   SQLCLIB, 'your.ezkey.core.library'
// EXTENT ,volser
// DLBL   IBMLIB, 'your.ibm.sql.relo.library'
// EXTENT ,volser
// LIBDEF RL, SEARCH=(SQLRLIB, IBMLIB)
// LIBDEF CL, TO=SQLCLIB
        INCLUDE ARIPRDID
// EXEC LNKEDT, SIZE=256K
/*
/&
                *****-1-*****
                *****-7-*****
                *****-8-*****
                *****-8-*****
                *****-8-*****
                *****-9-*****
                *****-9-*****
                *****-10-*****
                *****-10-*****
                *****-11-*****
                *****-11-*****

```

continued ...

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

... continued

```

// JOB CLOSE1
CLOSE SYSIPT,uuu
/*
/&
// JOB DQSMMTB
// DLBL IBMLIB,'your.ibm.sql.core.library'
// EXTENT ,volser
// DLBL SQLCLIB,'your.ezkey.core.library'
// EXTENT ,volser
// LIBDEF CL,SEARCH=(IBMLIB,SQLCLIB)
// DLBL IJSPCH,'ibm help text2 output',0
// EXTENT SYSPCH,volser,1,0,start,lgth
ASSGN SYSPCH,DISK,VOL=volser,SHR
// EXEC PGM=DQSMMTB,SIZE=256K
/*
/&
// JOB CLOSE2
CLOSE SYSPCH,uuu
/*
/&
// JOB ASSEMBL2
// DLBL IJSYSIN,'ibm help text2 output',0
// EXTENT SYSIPT,volser,1,0,start,lgth
ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL IJSPCH,'text2 object file',0
// EXTENT SYSPCH,volser,1,0,start,lgth
ASSGN SYSPCH,DISK,VOL=volser,SHR
// OPTION CATAL,DECK,LOG
// EXEC ASSEMBLY,SIZE=256K
/*
/&
// JOB CATAL2
CLOSE SYSIPT,uuu
CLOSE SYSPCH,uuu
// DLBL IJSYSIN,'text2 object file',0
// EXTENT SYSIPT,volser,1,0,start,lgth
ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL SQLRLIB,'your.sql.relo.library'
// EXTENT ,volser
// LIBDEF RL,TO=SQLRLIB
// EXEC MAINT,SIZE=AUTO
/*
/&

```

```

***-1-***
***-12-***

***-1-***
***-13-***
***-13-***
***-14-***
***-14-***

***-15-***
***-15-***
***-15-***

***-1-***
***-16-***

***-1-***
***-17-***
***-17-***
***-17-***
***-18-***
***-18-***
***-18-***

***-1-***
***-19-***
***-19-***
***-20-***
***-20-***
***-20-***
***-21-***
***-21-***

```

continued ...

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

... continued

```

// JOB CLOSE3                                     ***-1-***
CLOSE SYSIPT, cuu                                ***-22-***
/*
/&
// JOB MAINT                                     ***-1-***
// DLBL   SQLRLIB, 'your.sql.relo.library'       ***-23-***
// EXTENT ,volser                               ***-23-***
// LIBDEF RL,TO=SQLRLIB
// EXEC   MAINT,SIZE=AUTO
        CATALR OM#EKCIS
        ESD          DQSEKCI          -DFHEAIO          DQ#EKCI
        ESD          DQ#EKCI          -  QXGSTKFST        XFSTKFST
        ESD          DQSCINT          XHEAPRUN
...
/*
/&
// JOB PRECOMP2                                 ***-1-***
// OPTION NODUMP,LOG
// DLBL IBMLIB, 'your.ibm.sql.core.library'       ***-24-***
// EXTENT ,volser                               ***-24-***
// LIBDEF CL,SEARCH=IBMLIB
// DLBL IJSPCH, 'sql processed output',0          ***-25-***
// EXTENT SYSPCH,volser,1,0,start,lgth           ***-25-***
ASSGN SYSPCH,DISK,VOL=volser,SHR                 ***-25-***
// EXEC PGM=ARIPRPA,SIZE=AUTO,
        PARM='USERID=userid/password,
        PREPNAME=DQEKCMD,KEEP,NOPRINT,ISOLATION(CS)',
PUNCH / CATALR OM#MCMD1 /
*****
*
* THIS MODULE PROCESSES ALL SQL COMMANDS THAT ARE VALID FOR BOTH
...
MESSAGE DS      OH
        EXEC SQL
                DECLARE C1 CURSOR FOR
                SELECT "SQL/DS HELP"
                FROM SQLDBA.SYSTEXT1      TEXT1,
                SQLDBA.SYSTEXT2      TEXT2
                WHERE TOPIC = :CMDMSTPC AND
                TEXT1.ITEM = TEXT2.ITEM
...
/*
/&

```

continued ...

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

... continued

// JOB ASSEMBL3	***-1-***
CLOSE SYSPCH, cuu	***-27-***
// DLBL IJVSYSIN, 'sql processed output', 0	***-28-***
// EXTENT SYSIPT, volser, 1, 0, start, lgth	***-28-***
ASSGN SYSIPT, DISK, VOL=volser, SHR	***-28-***
// DLBL IJVSYPH, 'assembler object file', 0	***-29-***
// EXTENT SYSPCH, volser, 1, 0, start, lgth	***-29-***
ASSGN SYSPCH, DISK, VOL=volser, SHR	***-29-***
// OPTION CATAL, DECK, LOG	
// EXEC ASSEMBLY, SIZE=256K	
/*	
/ &	
// JOB CATAL3	***-1-***
CLOSE SYSIPT, cuu	***-30-***
CLOSE SYSPCH, cuu	***-30-***
// DLBL IJVSYSIN, 'assembler object file', 0	***-31-***
// EXTENT SYSIPT, volser, 1, 0, start, lgth	***-31-***
ASSGN SYSIPT, DISK, VOL=volser, SHR	***-31-***
// DLBL SQLRLIB, 'your.sql.relo.library'	***-32-***
// EXTENT , volser	***-32-***
// LIBDEF RL, TO=SQLRLIB	
// EXEC MAINT, SIZE=AUTO	
/*	
/ &	
// JOB CLOSE4	***-1-***
CLOSE SYSIPT, cuu	***-33-***
// EXEC PROC=ARISLIBS	
// EXEC PROC=ARISLIBP	
// EXEC PGM=ARIDBS, SIZE=AUTO	
CONNECT userid IDENTIFIED BY password;	***-34-***
GRANT RUN ON userid.DQEKCMD TO PUBLIC;	***-34-***
/*	
/ &	

continued ...

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

... continued

```

// JOB SQLLNK
// DLBL   SQLRLIB,'your.sql.relo.library'
// EXTENT ,volser
// DLBL   SQLCLIB,'your.ezkey.core.library'
// EXTENT ,volser
// DLBL   IBMLIB,'your.ibm.sql.relo.library'
// EXTENT ,volser
// LIBDEF RL,SEARCH=(SQLRLIB,IBMLIB)
// LIBDEF CL,TO=SQLCLIB
// OPTION LOG,CATAL
// PHASE DQSEKCI,*
// INCLUDE DFHEAI
// INCLUDE OM#EKCIS
...
// EXEC LNKEDT
/*
/&
* $$ EOJ

```

DOS/VSE SQL Installation JCL Notes

1. Job CREATE creates the SQL relocatable library.
2. Job PRECOMP1 preprocesses the SQL/DS interface module, DQSMMTB.
3. Job ASSEMBL1 assembles the output of the preprocessor.
4. Job CLOSE1 reassigns SYSIPT back to its original address.
5. Job DQSMMTB executes DQSMMTB.
6. Job CLOSE2 reassigns SYSPCH back to its original address.
7. Job ASSEMBL2 assembles the IBM message text data.
8. Job CATAL2 places the assembler object into an object library.
9. Job CLOSE3 reassigns SYSIPT back to its original address.
10. Job MAINT places the SQL Interface modules into the relocatable library.
11. Job PRECOMP2 preprocesses the SQL command processor.

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

12. Job ASSEMBL3 assembles the SQL command processor.
13. Job CATAL3 places the assembler object file into an object library.
14. Job CLOSE4 reassigns SYSIPT back to its original address.
15. Job SQLLNK links the Pansophic SQL Interface.

Tailoring DOS/VSE SQL Installation JCL

At the right of some of the JCL lines shown above is the character string `***-n-***`, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Change the JOB statement to conform to your installation's standards.
2. Change the dataset name, volser, and starting and ending track addresses, on the ASSGN, DLBL, and EXTENT statements to identify the disk dataset to contain the SQL relocatable modules. A minimum of 30 3350 tracks or 1300 blocks is recommended for this dataset.
3. Indicate the number of cylinders and tracks for the relocatable library. Refer to your IBM manuals for more information on CORGZ.
4. Change the dataset name on the IBMLIB DLBL statement to identify the object library that contains the IBM core image library for SQL/DS. Change the volser on the EXTENT statement to identify the correct volume.
5. Change the dataset name, volser, and starting and ending track addresses, on the IJSYSPH, SYSPCH, and ASSGN statements to identify an output disk dataset. A minimum of 40 3350 tracks is recommended for this dataset.
6. Change the userid and password for the SQLDBA UserID.
 - a. As distributed, the DQSMMTB program assumes the name of the SQL/DS Help Table is SQLDBA.SYSTEXT2. If your site changed the name of this table during installation of SQL/DS, you must supply the name you use at the location in the job stream marked by `***-a-***`. Note that `***-a-***` appears only in this documentation, not the file you are working with.

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

- b. Change the userid and password for the SQLDBA UserID as you did for Step 6. Note that ****-b-**** appears only in this documentation, not in the file you are working with.
7. Change cuu on the CLOSE statement to point back to the original address for SYSPCH.
8. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 5.
9. Change the dataset name and volser on the SQLRLIB DLBL statement to identify the disk dataset from step 2.
10. Change the dataset name and volser to identify your EZ/KEY core image library.
11. Change the dataset name on the IBMLIB DLBL statement to identify the object library that contains the IBM SQL/DS relocatable modules. Change the volser on the EXTENT statement to identify the correct volume.
12. Change cuu on the CLOSE statement to point back to the original address for SYSIPT.
13. Change the dataset name on the IBMLIB DLBL statement to identify the object library that contains the IBM core image library for SQL/DS. Change the volser on the EXTENT statement to identify the correct volume.
14. Change the dataset name and volser to identify your EZ/KEY core image library.
15. Change the dataset name, volser, and starting and ending track addresses on the IJSYSPH, SYSPCH, and ASSGN statements to identify an output disk dataset. A minimum of 37 3350 tracks is recommended for this dataset.
16. Change cuu on the CLOSE statement to point back to the original address for SYSPCH.
17. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 15.

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

18. Change the dataset name, volser, and starting and ending track addresses on the IJSYSPH, SYSPCH, and ASSGN statements to identify an output disk dataset. A minimum of 37 3350 tracks is recommended for this dataset.
19. Change cuu on the CLOSE statements to point back to the original addresses for SYSIPT and SYSPCH.
20. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 18.
21. Change the dataset name and volser on the SQLRLIB DLBL and EXTENT statements to identify the disk dataset from step 2.
22. Change cuu on the CLOSE statement to point back to the original address for SYSIPT.
23. Change the dataset name and volser on the SQLRLIB DLBL and EXTENT statements to identify the disk dataset from step 2.
24. Change the dataset name on the IBMLIB DLBL statement to identify the object library that contains the IBM core image library for SQL/DS. Change the volser on the EXTENT statement to identify the correct volume.
25. Change the dataset name, volser, and starting and ending track addresses on the IJSYSPH, SYSPCH, and ASSGN statements to identify the disk dataset from step 8.
26. Change the userid and password for the SQLDBA Userid as you did in step 6.

The command processor assumes that the names of the SQL/DS Help Tables are SQLDBA.SYSTEXT1 and SQLDBA.SYSTEXT2. If your site changed the names of these tables during installation of SQL/DS, you must supply the names you use at the location in the job stream marked by ***-c-***. Note that ***-c-*** appears only in this documentation, not in the file you are working with.
27. Change cuu on the CLOSE statement to point back to the original address for SYSPCH.
28. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 25.

SQL Interface Installation - DOS/VSE (pre-SP 2.1)

29. Change the dataset name, volser, and starting and ending track addresses on the IJSYSPH, SYSPCH, and ASSGN statements to identify an output disk dataset. A minimum of 27 3350 tracks is recommended for this dataset.
30. Change cuu on the CLOSE statements to point back to the original addresses for SYSIPT and SYSPCH.
31. Change the dataset name, volser, and starting and ending track addresses on the IJSYSIN, SYSIPT, and ASSGN statements to identify the disk dataset from step 29.
32. Change the dataset name and volser on the SQLRLIB DLBL and EXTENT statements to identify the disk dataset from step 2.
33. Change cuu on the CLOSE statement to point back to the original address for SYSIPT.
34. Change the userid and password for the SQLDBA UserID as you did in step 26.
35. Change the dataset name and volser on the SQLRLIB DLBL and EXTENT statements to identify the disk dataset from step 2.
36. Change the dataset name and volser to identify your EZ/KEY core image library.
37. Change the dataset name on the IBMLIB DLBL statement to identify the object library that contains the IBM core image library for SQL/DS. Change the volser on the EXTENT statement to identify the correct volume.

ADDITIONAL SQL CICS CONSIDERATIONS

EZ/KEY must be able to access the following modules under CICS: DQSEKCF, DQSEKCI, DQSEKCR, DQSEKCS, and DQSEKCT. Therefore, if you have not already done so, you must install the PPT entries for these modules. For further information see "CICS/DOS/VSE Table Changes" earlier in this section.

If you have just completed installing these modules, you may need to tell CICS to load new copies of them. To get new copies, do *one* of the following:

- Bring CICS down and up again.

or

- Issue the CICS command.

CEMT INQUIRE PROGRAM(DQS*)

Then issue a *Newcopy* for each of these programs, as shown in the following illustrations.

Additional SQL/DS Considerations

```
INQUIRE PROGRAM(DQS*)
STATUS: RESULTS - OVERTYPE TO MODIFY
Pro(DQSEKCF ) Len(0000000) Res(000) Use(0000000) Ass Ena new
Pro(DQSEKCI ) Len(0000000) Res(000) Use(0000000) Ass Ena new
Pro(DQSEKCR ) Len(0000000) Res(000) Use(0000000) Ass Ena new
Pro(DQSEKCS ) Len(0000000) Res(000) Use(0000000) Ass Ena new
Pro(DQSEKCT ) Len(0000000) Res(000) Use(0000000) Ass Ena new
```

```
RESPONSE: NORMAL
PF: 1 HELP      3 END
```

```
TIME: 14.07.53  DATE: 87.244
7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Exhibit 5.1: Requesting a Newcopy

To issue a newcopy, tab the cursor to the position next to the word **Ena**, type the word **new**, as shown, and press ENTER. The result is shown in the following illustration.

```

INQUIRE PROGRAM(DQS*)
STATUS: RESULTS - OVERTYPE TO MODIFY
Pro(DQSEKCF ) Len(0060432) Res(000) Use(000000) Ass Ena      NEW COPY
Pro(DQSEKCI ) Len(0058248) Res(000) Use(000000) Ass Ena      NEW COPY
Pro(DQSEKCR ) Len(0012304) Res(000) Use(000000) Ass Ena      NEW COPY
Pro(DQSEKCS ) Len(0065392) Res(000) Use(000000) Ass Ena      NEW COPY
Pro(DQSEKCT ) Len(0057304) Res(000) Use(000000) Ass Ena      NEW COPY

```

RESPONSE: NORMAL
PF: 1 HELP 3 END

TIME: 14.08.30 DATE: 87.244
7 SBH 8 SFH 9 MSG 10 SB 11 SF

Exhibit 5.2: Results of Newcopy Request

SQL/DS Userids

The EZ/KEY SQL Interface under CICS invokes SQL/DS to perform syntax checking of the SQL statements embedded in your EASYTRIEVE PLUS programs.

EZ/KEY passes a userid and password to the SQL Interface, using what is coded on the PARM statement USERID parameter in the EASYTRIEVE PLUS program. The PARM USERID parameter is required since EASYTRIEVE PLUS uses it to identify the appropriate userid to SQL/DS when the batch job runs. EZ/KEY users must be aware of this when coding SQL statements in their EASYTRIEVE PLUS programs. All security and access checking performed by SQL/DS is based on this userid, not the EZ/KEY userid.

Your SQL/DS administrator must GRANT authority to access appropriate tables within SQL/DS to each userid, as required. You could make the EZ/KEY userids match the SQL/DS userids at your site, eliminating a possible source of confusion for your EZ/KEY users.

OPTIONAL IDMS/IDD INSTALLATION

Your site must have IDMS-CV (Central Version) installed and working with the CICS system where EZ/KEY is installed. This requires that you add table entries to your CICS system. For an example of how to code the PLT and PPT entries for IDMSINTC, see the sample table entries provided with EZ/KEY.

For more information regarding adding table entries to your CICS system and for details on installing IDMS/IDD into CICS, refer to the following Cullinet publications:

- *Cullinet System Software, Integrated Installation Guide.* See the section on special considerations for interfaces, particularly the Central Version/CICS interface.
- *Cullinet System Software, System Generation.* See the section on ABEND detection.
- *IDMS-CV/DC System Operations.* See the section on TP monitor considerations for CICS.

IDMS/IDD Installation JCL

In addition to the requirements noted above, you must also run the following job to create the EZKYIDMS module EZ/KEY needs to access the IDMS IDD Interface. This allows you to use the IDD statement from within EZ/KEY.

A copy of the JCL shown on the next page is in EXAMPLE.PIELIB in the member named LKIDDVSE JCL. The JCL must be modified as described in "Tailoring IDMS/IDD JCL."

* \$\$ JOB UNM=jobname,CLASS=A,DISP=D	***-1-***
// JOB CATAL EZKYIDMS module	***-1-***
// ASSGN SYS003,xxx	***-2-***
// DLBL EZKEY,'your.EZKEY.core.image.library',99/365,SD	***-3-***
// EXTENT SYS003,volser,1,0,nnnn,80	***-3-***
// LIBDEF CL,TO=EZKEY	***-4-***
// DLBL IDMS,'your.idms.reloc.lib'	***-5-***
// EXTENT ,volser	***-5-***
// LIBDEF RL,SEARCH=IDMS	***-4-***
// OPTION LOG,CATAL	
ACTION NOAUTO	
PHASE EZKYIDMS,ROOT	
INCLUDE IDMSCINT	
/*	
// EXEC LNKTEDT	
/&	
* \$\$ EOJ	

Tailoring IDMS/IDD JCL

At the right of some of the JCL lines above is the character string ***-*n****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB statement as required for your installation.
2. Specify the device information for the disk volume where your EZ/KEY core image library resides.
3. Specify the dataset name, volume serial, and extent information for the EZ/KEY core image library where all EZ/KEY phases reside. This job creates the new phase EZKYIDMS and places it in this library.
4. For VSE/SP 2.1 and above only, change RL to OBJ, change CL to PHASE on the indicated statements, and change TO=EZKEY to CATALOG=EZ.KEY.
5. Specify the dataset name of the relocatable library where the IDMS object module IDMSCINT resides.

After modifying the JCL, submit it for execution under DOS/VSE. The job should run with no errors.

Additional IDMS/IDD CICS Considerations

To permit EZ/KEY users to access the IDD using Central Mode, add the following statement to your CICS start-up JCL:

```
// UPSI b
```

where *b* is the appropriate UPSI switch, if one was specified in the IDMSOPTI module linked with EZTPA00 when EASYTRIEVE PLUS was installed. Refer to the *EASYTRIEVE PLUS Installation Guide* for more information. Consult your IDMS Data Base Administrator for assistance.

If you have not already done so, you must install the PPT entry for the module EZKYIDMS. See "CICS/DOS/VSE Table Changes" earlier in this section for further information.

If you have just completed installing this program, you may need to tell CICS to load a new copy of this phase. To do this, do *one* of the following:

- Bring CICS down and then up again.

Or

- Issue the CICS command.

CEMT INQUIRE PROGRAM(EZKYIDMS)

Then issue a *Newcopy* for this program, as shown in the following illustrations.

```
INQUIRE PROGRAM(EZKYIDMS)
STATUS: RESULTS - OVERTYPE TO MODIFY
Pro(EZKYIDMS) Len(0000000) Res(000) Use(000000) Ass Ena new
```

```
RESPONSE: NORMAL
PF: 1 HELP      3 END
```

```
TIME: 14.22.47  DATE: 87.244
7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Exhibit 5.3: Requesting a Newcopy

To issue a newcopy, tab the cursor to the position next to the word **Ena**, type the word *new*, as shown, and press ENTER. The result is shown in the following illustration.

```
INQUIRE PROGRAM(EZKYIDMS)
STATUS: RESULTS - OVERTYPE TO MODIFY
Pro(EZKYIDMS) Len(0000328) Res(000) Use(000000) Ass Ena      NEW COPY

RESPONSE: NORMAL
PF: 1 HELP          3 END

TIME: 14.22.47  DATE: 87.244
7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Exhibit 5.4: Result of Newcopy Request

IDMS/IDD Interface Storage Usage

With a single IDD statement, an EZ/KEY user can bring into virtual storage hundreds or even thousands of field definitions from the IDMS/IDD Dictionary. Each field definition in EZ/KEY requires approximately 256 bytes of symbol table space in virtual memory, which is obtained from the CICS dynamic storage area belonging to the user who makes the request. This space is released after a short period of time.

If you enable the *Incremental Compile* feature of EZ/KEY, these field definitions remain in virtual storage for the duration of the EZ/KEY user's Edit session. Caution is therefore advised with regard to using this feature of EZ/KEY in conjunction with the IDD Interface.

In a CICS/DOS/VSE environment, you should not set Incremental Compile to YES; rather ensure that it is set to NO (the default). See Section 3 of the *EZ/KEY Administrator's Guide* for information regarding the Incremental Compile feature of EZ/KEY.

ADDITIONAL IDMS CONSIDERATIONS

Your site must have IDMS-CV (Central Version) installed and working with the CICS system where EZ/KEY is installed. This requires you to add table entries to your CICS system. For an example of how to code the PLT and PPT entries for IDMSINTC, see the sample table entries provided with EZ/KEY.

For more information on adding table entries to your CICS system and for details on installing IDMS/IDD into CICS, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the section that covers TP monitor considerations for CICS.

IDMSCINT Generation

You must have a version of the IDMS module IDMSCINT available for link editing with EZ/KEY. This module must be generated for use with Command Level programs.

The following example illustrates how to code the IDMSCINT macro to produce a Command Level version of the IDMSCINT program:

```
IDMSCINT IDMSCINT CWADISP=O,EXEC=YES  
END
```

The EXEC=YES parameter indicates Command Level programming. The CWADISP parameter value may be different for your installation. Consult your IDMS Data Base Administrator or a systems programmer who installed IDMS for assistance.

For more information regarding use of the IDMSCINT macro instruction, refer to the Cullinet *IDMS-CV/DC Integrated Installation Guide*. Specifically, see the section that covers special considerations for the Central Version/CICS Interface.

IDMSINTC Generation

You must have a specific version of the IDMS module IDMSINTC installed under CICS for use by EZ/KEY. The following example illustrates how to code the IDMSINTC macro to produce a version of this program for use with CICS:

```
IDMSINTC IDMSINTC SVC=255,
                                CVNUM=01,
                                CWADISP=0,
                                CSASIZE=512,
                                CICSLVL=16,
                                ACTDISP=12,
                                OPSYS=DOSVS,
                                TPNAME=CICS,
                                SYSCTL=,
                                ESCDLI=NO,
                                NODENAM=,
                                DBNAME=
                                END
```

The SVC, CVNUM, CWADISP, CICSLVL, ACTDISP, NODENAM and DBNAME parameters may be different for your installation. Consult your IDMS Data Base Administrator or a systems programmer who installed IDMS for assistance.

For more information on using the IDMSINTC macro instruction and for information on installing IDMS in a CICS environment, refer to the Cullinet *IDMS-CV/DC Integrated Installation Guide*. See the section that covers special considerations for the Central Version/CICS interface.

IDMS Background Execution Considerations**Central Version Mode Background Operation**

If EASYTRIEVE PLUS is to access the IDD through Central Version, your site must have IDMS-CV (Central Version) installed on the DOS/VSE system where EASYTRIEVE PLUS runs.

Additional IDMS Considerations

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Central Mode, add the following statement to the EZTPLUS Header JCL for Background Execution in EZ/KEY:

```
// UPSI b
```

where *b* is the appropriate UPSI switch, if one was specified in the IDMSOPTI module linked with EZTPA00 when EASYTRIEVE PLUS was installed. Refer to the *EASYTRIEVE PLUS Installation Guide* for more information. Consult your IDMS Data Base Administrator for assistance.

For more information regarding operating IDMS/IDD in VSE, refer to the Cullinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections that cover the following topics:

- Program and IDMS-CV/DC Communication
- DOS/VS(E) Considerations

Local Mode Background Operation

To permit EZ/KEY users to run batch EASYTRIEVE PLUS programs that access IDMS and IDD using Local Mode, add the statements shown below to the EZTPLUS Header JCL for Background Execution in EZ/KEY:

1. Identify the phase libraries where the Cullinet IDMS phases reside. This provides access to the phases needed to execute IDMS in Local Mode:

```
// DLBL IDMS,'your.IDMS.phase.library'  
// EXTENT ,volser  
// LIBDEF CL,SEARCH=(EZTP,IDMS)
```

where *your.IDMS.phase.library* is the name of the phase or core image library for IDMS, and *volser* is the label of the disk volume where the phase library is located.

NOTE: For VSE/SP 2.1, change **CL** to **PHASE** on the LIBDEF statement in the above illustration.

Additional IDMS Considerations

2. Identify the IDMS journal file for use by EASYTRIEVE PLUS programs. Use either:

```
// DLBL SYSJRNL,'your.idms.journal.file'  
// EXTENT ,volser
```

or

```
// TLBL SYSJRNL,'your.idms.journal.file',,volser,,f
```

where *your.idms.journal.file* is the name of the journal file for IDMS, *volser* is the label of the disk volume where this file is located, and *f* is the file number if the file is on tape.

3. Identify all of the databases accessed by EASYTRIEVE PLUS programs:

```
// DLBL dbase1,'your.idms.database.file1',O,SD  
// EXTENT ,volser
```

```
// DLBL dbaseN,'your.idms.database.fileN',O,SD  
// EXTENT ,volser
```

where *dbaseN* and *your.idms.database.fileN* are a DLBL name and dataset name, respectively, and *volser* is a label of a disk volume where a database file is located. There must be DLBL and EXTENT statements in the EZTPLUS JCL for each database that is accessed by any EASYTRIEVE PLUS program running in Local Mode. Consult your IDMS Data Base Administrator for assistance. Refer to the *EASYTRIEVE PLUS Installation Guide* for more information.

For more information on operating IDMS/IDD in VSE, refer to the Culinet publication *IDMS-CV/DC System Operations*. Specifically, see the sections covering the following topics:

- Program and IDMS-CV/DC Communication
- DOS/VS(E) Considerations

SECTION 6 MAINTENANCE

Introduction	6-1
Support	6-1
Technology	6-1
Product Awareness	6-1
Customer Service Interface	6-2
Know Software Version	6-2
Review Documentation	6-2
Maintain Paper Trail	6-2
Describe Problem	6-2
Maintenance Procedures	6-3
DOS Maintenance Procedures	6-3
OS Maintenance Procedures	6-5

(This page intentionally left blank.)

INTRODUCTION

Maintenance is an important and effective service which ensures that your company and software product receive proper attention when needed. Maintenance with PANSOPHIC means:

Support

Members of the PANSOPHIC Customer Service Staff respond promptly and accurately to your inquiries and problems. They are prepared to handle any situation relating to PANSOPHIC products, operating environments, or general data processing. Standard support hours are from 7:30 a.m. to 7:00 p.m. Central Time. The support hotline telephone number is (312) 357-5920.

Technology

PANSOPHIC constantly enhances its products to interface with modern technology as well as to provide new features and capabilities.

Product Awareness

Customer Service provides information in three major areas:

- Problems – Product early warning reports are distributed to users quarterly. Priority or unique situations are handled by special mailings or telephone calls.
- New Ideas – Customer involvement in new ideas is essential to product success. Users get involved with product verification and acceptance of new design and code prior to release.
- Releases – Customers receive advance notification of new releases and new features of PANSOPHIC products. These releases are distributed to customers at no extra cost.

CUSTOMER SERVICE INTERFACE

Your knowledge of EZ/KEY and your operating system environment is vital. You can do several things to help obtain a timely response from the PANSOPHIC Customer Service staff.

Know Software Version

Determine the version of EZ/KEY you are running. The version number is displayed on PROSTART -- the EZ/KEY Primary Selection Menu.

Review Documentation

If you have a problem, review the *EZ/KEY Reference Manual*, the *Administrator's Guide*, and the *Installation Guide*. Try to find examples of the function you are trying to execute. Determine how the example in the Guide is different from your attempt to perform a similar function.

In addition, review the quarterly microfiche to see if the malfunction has already been reported and resolved. The microfiche contains an index of the problem descriptions for each PANSOPHIC product. The entry for the problem usually contains a fix or a circumvention.

Maintain Paper Trail

Maintain a paper trail. If you have a problem with EZ/KEY write down what you were doing. If you get an abnormal termination (ABEND), get a printout of the panel and any output generated before the ABEND occurred.

Describe Problem

When you call Customer Service, tell the receptionist you are calling about EZ/KEY. The receptionist will connect you with someone who can answer technical questions. Describe a problem beginning with symptoms. For example: "I have EZ/KEY and I'm getting an error on the PROPED PANEL."

This helps Customer Service determine if your problem has already been reported and perhaps fixed. The problem lists are arranged by symptom to facilitate quick identification.

MAINTENANCE PROCEDURES

Malfunctions in EZ/KEY are resolved by distributing system refresh releases or PTFs (Program Temporary Fixes).

An EZ/KEY refresh release contains the entire EZ/KEY system. You must reinstall the EZ/KEY module(s) from file two of the distribution tape and the SYSTEM PIELIB from file three.

For VM/CMS systems, transfer the files on the distribution tape to a temporary mini-disk and copy/replace the EZ/KEY module and System PIELIB. EZ/KEY users must reaccess the mini-disk after completing the copy/replace process.

PTFs in MVS AMASPZAP or VSE PDZAP format are sent to you to resolve some malfunctions. The following describes how to use PDZAP and AMASPZAP.

DOS Maintenance Procedures

PDZAP is a program that is supplied and supported by IBM. PDZAP allows you to modify programs cataloged in a system or private core image library. It prints the changes on SYSLST. File this printout with the installation link edit listing to keep track of the changes made.

PDZAP can be executed in any partition. SYSLOG must be assigned to the operator console. When card input is used, SYSIPT must be assigned to a card reader. SYSLST should be assigned to a printer.

Maintenance Procedures

The following example illustrates the use of the control cards necessary to execute PDZAP:

* \$\$ JOB JNM=jobname	
// JOB PDZAP	
// ASSGN SYS003,nnn	Specifies the library and
// DLBL EZKEY,'your.EZ/KEY.sysc1b',0,SD	and location where the
// EXTENT SYS003, ...	EZ/KEY executable
	modules reside.
// LIBDEF CL,TO=EZKEY,TEMP	Specifies the name of the
	library to be used.
// UPSI 1	Specifies card input.
// EXEC PDZAP,SIZE=100K	Executes the program.
NAME=xxxxxxx	Specifies a USER NAME.
EZKYxxxx	Identifies the module
	to be updated.
+6D4	Specifies the offset of
	the data to be displayed.
	(loaded address of phase
	minus loaded address of
	csect plus the offset of
	data to be displayed)
VER=0580,1211	Specifies the data to be
or	verified; if the data is
VER=05801211	not found, no updates occur.
REP=0700,0700	Specifies the data which
or	is to replace the data
REP=07000700	just verified.
END	Terminates the run.
/*	
// EXEC LISTLOG,SIZE=96K	
/&	
* \$\$ EOJ	

Additional information about PDZAP can be found in the IBM publication *DOS/VSE Advanced Functions - Serviceability Aids and Debugging Procedures* (SC33-6099).

OS Maintenance Procedures

OS AMASPZAP is a program that is supplied and supported by IBM. It modifies modules that reside in an executable load library.

Input is read from SYSIN; output is written to SYSPRINT. The modules to be modified are pointed to by SYSLIB.

The following example illustrates how to use the control cards necessary to execute AMASPZAP:

```
//jobname JOB accounting.info
//STEP EXEC PGM=AMASPZAP
//SYSPRINT DD SYSOUT=A
//SYSLIB DD DSN=your.EZ/KEY.loadlib,DISP=OLD
                                           Dsname is the name of the load
                                           library where the executable
                                           EZ/KEY modules reside.

//SYSIN DD *
NAME module-name csect-name             Identifies the module and
                                           csect to be updated

      (offset) (content)
VER 06D4      05810203                   Requests comparison between
                                           the data specified at the
                                           offset address and the data
                                           supplied in the content
                                           parameter.

      (offset) (content)
REP 06D4      07000700                   Requests replacement at
                                           the offset address by the
                                           data supplied in the content
                                           parameter.

/*
//
```

(This page intentionally left blank.)

APPENDIX A INSTALLATION SUBMIT USER EXIT

Introduction	A-1
Summary	A-3
Parameter Lists	A-6
Calling Sequence	A-8
Installation of SUBMIT User Exit under CMS	A-9
Installation of SUBMIT User Exit under TSO	A-11
Installation of SUBMIT User Exit under CICS/OS/VS	A-12
Installation of SUBMIT User Exit under CICS/DOS/VSE	A-13
Link-Editing CICS/DOS/VSE Phases	A-14
Restore Relocatable Library from Installation Tape	A-14
Tailoring Relocatable Library Restore JCL	A-14
Link-Edit Phase EZKYZ00	A-15
Tailoring CICS/DOS/VSE Link-Edit JCL	A-16
Link-Edit Phase EZKYZ01	A-17
Tailoring VSELKED1 Link-Edit JCL	A-17

(This page intentionally left blank.)

INTRODUCTION

JOB submission may be modified under installation exit control in the following ways:

- The submitted JCL card images may be changed during the submit process.
- Card images may be inserted or deleted during the submit process.
- The job being submitted may be cancelled by the installation exit at any time during the submit process.

If present, the submit user exit is given control once for each line of source that has a **slash (/)** in column one of the card image (JCL), or under CMS or VSE, * **\$\$** in columns 1-4 of the card image (JECL). The exit is not given control for any other source card images, unless specifically requested by the user exit. A parameter list is passed to the exit on each call.

When the submit user exit is running under CICS, it does not have addressability to any CICS control blocks or areas. No CICS or OS macros should be issued in the exit routine. Standard IBM linkage conventions must be followed.

Working storage for the exit routine is retained for the duration of the job submission. It must be initialized on the first call of each job submission by the exit routine itself.

In CICS, job submissions are single-threaded; the submit user exit routine need only be concerned with one job at a time. JCL for multiple jobs is never intermixed.

For this release of EZ/KEY, a submit user exit, if installed, must be written in Assembler Language. Someone knowledgeable in assembler language must write and install this exit in EZ/KEY, since an incorrect exit routine can have adverse effects on the performance and correct operation and functioning of EZ/KEY. For example, an error in the user exit could cause EZ/KEY toabend each time a user attempts to submit a job to batch. For this reason, you should plan to test your submit user exit carefully before installing it permanently.

Sample exits for OS and DOS environments are provided in the EXAMPLE PIELIB distributed with EZ/KEY. The members are named PIEEXIT01 ASMOS and PIEEXIT01 ASMDOS. Print out the appropriate version for your environment, so that you can refer to it when studying this Appendix. For CMS and TSO users, you can copy the sample exit directly to your CMS or TSO library. You can then modify it using XEDIT or ISPF.

You can also create up to twelve unique messages for use with your submit user exit routine. A sample MESSAGES member is provided in the EXAMPLE PIELIB distributed with EZ/KEY. This member is named SUBMSG MESSAGES. This member illustrates the format for creating your own customized messages for use with your submit exit. You can use the EZ/Editor (Option 2) to create or edit this member. Be sure to set CAPS OFF if you want to have lower case letters in your messages.

To test your messages with your submit exit routine, copy SUBMSG MESSAGES into the SYSTEM PIELIB to make the messages available to all users. You can do this by connecting SYSTEM with CREATE access using utility 3.a.3, and then using utility 3.3 to copy the member to SYSTEM.

Standard IBM linkage conventions are used to call the submit exit:

- R15 - Entry point address of the user reader exit routine.
- R14 - Return address.
- R13 - Address of an 18-fullword save area.
- R1 - Address of the parameter list described below.
- all other registers are unpredictable.

SUMMARY

The exit routine must follow standard IBM linkage conventions for saving and restoring registers.

The exit routine must not attempt to modify any area other than the current 80-byte card image buffer, or the dynamic work area provided.

Control is given to the submit exit only if a card image has a **slash (/)** in column one (JCL) or *** \$\$** in columns 1-4 for DOS/VSE JECL.

A submit user exit is not required to use the EZ/KEY batch job submission capability. If not present, no calls to an exit are attempted. Unless you desire to edit or validate JCL prior to processing by the system, a submit exit should not be implemented.

During the course of SUBMIT processing, each JCL statement to be submitted is passed to the exit routine.

The return codes (to be set in register 15 before the exit routine returns control to the SUBMIT processor) are shown on the following pages.

Undefined return codes cause the SUBMIT processor to issue an error message and terminate submit processing for this job.

Value	Symbolic	Meaning
0	OK	- process the current statement.
15	ALL	- examine all source statements, not just JCL.
16	CANCEL	- cancel the JOB being processed.
17	INSERT	- process the current statement and invoke the exit routine again so that it can insert another statement.
18	DELETE	- ignore the current statement.
19	SUBMSG01	- a message is displayed at the terminal in the message area of the screen. The message text must be stored in member SUBMSG MESSAGES in SYSTEM PIELIB. Up to 12 unique installation messages are provided for.
20	SUBMSG02	- message SUBMSG02 is displayed.
21	SUBMSG03	- message SUBMSG03 is displayed.
22	SUBMSG04	- message SUBMSG04 is displayed.
23	SUBMSG05	- message SUBMSG05 is displayed.
24	SUBMSG06	- message SUBMSG06 is displayed.
25	SUBMSG07	- message SUBMSG07 is displayed.
26	SUBMSG08	- message SUBMSG08 is displayed.
27	SUBMSG09	- message SUBMSG09 is displayed.
28	SUBMSG10	- message SUBMSG10 is displayed.
29	SUBMSG11	- message SUBMSG11 is displayed.
30	SUBMSG12	- message SUBMSG12 is displayed.

For assembler language, these may be defined as follows:

OK	EQU	0
ALL	EQU	15
CANCEL	EQU	16
INSERT	EQU	17
DELETE	EQU	18
SUBMSG01	EQU	19
SUBMSG02	EQU	20
SUBMSG03	EQU	21
SUBMSG04	EQU	22
SUBMSG05	EQU	23
SUBMSG06	EQU	24
SUBMSG07	EQU	25
SUBMSG08	EQU	26
SUBMSG09	EQU	27
SUBMSG10	EQU	28
SUBMSG11	EQU	29
SUBMSG12	EQU	30

As an example, to return the CANCEL return code in R15, code:

```
LA    R15,CANCEL
```

PARAMETER LISTS

On entry, register 1 contains all zeros the first time the exit routine is called for a job to be submitted. This is a SIZE call. The exit routine may request a work area to be obtained on its behalf. The size of the requested work area, in bytes, should be returned in R15, or a zero should be returned in R15 if no dynamic working storage is required.

Otherwise, register 1 contains a standard IBM parameter list pointer. That is, register 1 contains the address of a four-word parameter list. The meaning of each word in the parameter list is defined below:

- WORD 1 - Contains a 4-character field that indicates what kind of call this is, either OPEN, PUT, INS, or CLOS. The exit routine should examine this field to determine the meaning of WORD 3 in the parameter list, as well as how to process the data and what actions are permitted.
- WORD 2 - Contains the address of the EZ/KEY userid. The userid is eight characters, left-justified, padded with blanks.
- WORD 3 - Depends on the type in WORD 1 of the parameter list:
- for OPEN - Contains the address of a System ID field, which is eight characters, left-justified, padded with blanks.
 - for PUT - Contains the address of the card image, which is 80 characters, left-justified, padded with blanks.
 - for INS - Contains the address of the card area, which is 80 characters, left-justified, padded with blanks.
 - for CLOS - Contains the last return code in the low-order byte of WORD 3 (R15 the last time the exit was called).

WORD 4 – Contains the address of the exit routine's work area, or all zeros if the exit did not request any working storage. The exit routine must initialize the area when called for OPEN and should be careful not to overlay storage beyond the amount requested.

The following return codes are valid for each type of call:

For OPEN – Only OK, ALL, CANCEL, or SUBMSG01 through SUBMSG12 are valid. You cannot INSERT or DELETE.

For PUT – All of the return codes listed above are valid, except ALL.

For INS – All of the return codes listed above are valid, except ALL.

For CLOS – Only OK, CANCEL or SUBMSG01 through SUBMSG12 are valid. You cannot INSERT or DELETE.

The results of issuing a return code of CANCEL or SUBMSG01 through SUBMSG12 are described below:

- CANCEL stops all further processing of the member being submitted and issues a standard message.
- SUBMSG01 through SUBMSG12 have the same effect as a CANCEL return code, but the exit also specifies a customized message that you provide.

CALLING SEQUENCE

The first call is always a SIZE request. The second call is always an OPEN request. Depending on the return code given by your exit, the next call may be a PUT, an INS, or a CLOS.

The normal sequence is SIZE, OPEN, PUT,...PUT, CLOS.

If your exit returns a CANCEL or SUBMSG01 through SUBMSG12 on the OPEN call, the sequence is SIZE, OPEN, CLOS.

If your exit returns a CANCEL or SUBMSG01 through SUBMSG12 on the first PUT call, the sequence is SIZE, OPEN, PUT, CLOS.

INSTALLATION OF SUBMIT USER EXIT UNDER CMS

EZ/KEY user exits are referenced through weak external references (WXTRN). Therefore it is necessary to insert into the linkedit stream an INCLUDE statement for each desired exit.

Standard IBM OS linkage conventions are required. It is recommended that the exit be coded in assembler language. The exit should be linked into the EZ/KEY load module with a CSECT name of PIEXIT01.

To include your exit in EZ/KEY in a DCSS:

1. Copy your PIEXIT01 TEXT file to the minidisk where you keep all of the EZ/KEY DCSS materials and object decks.
2. Edit the EZKEYSEG EXEC and find the following:

```
*      INCLUDE PIEXIT01 ( SAME )
```

You can find this alphabetically in the list of include commands in the EXEC. Remove the asterisk. Save the updated EXEC.

3. Run EZKEYSEG EXEC to re-install EZ/KEY in the DCSS.
4. Run the EZKEYMAP EXEC. This edits the new EZKEYSEG MAP created by step 3 above.

To include your PIEXIT01 routine in the EZKEYMOD MODULE:

1. If you have not installed EZ/KEY in a DCSS, or you did not install the DCSS materials on a permanent minidisk, load all of the TEXT decks and associated materials from File 2 of the tape, as described on page 2-17.
2. Copy your PIEXIT01 TEXT deck to the minidisk where the above materials are stored.
3. Modify the EZKEYLNK EXEC to tailor it to your installation. Find the following:

```
*      INCLUDE PIEXIT01 ( SAME )
```

You can find this alphabetically in the list of include commands in the EXEC. Change the asterisk to a blank. Save the updated EXEC.

Installation of SUBMIT User Exit under CMS

4. Run the EZKEYLNK EXEC. This creates a new EZKEYMOD MODULE on the minidisk where you have installed EZ/KEY.
5. Run the EZKEYMAP EXEC. This edits the new EZKEYMOD MAP created by step 4.

INSTALLATION OF SUBMIT USER EXIT UNDER TSO

EZ/KEY user exits are referenced through weak external references (WXTRN). Therefore it is necessary to insert into the linkedit stream an INCLUDE statement for each desired exit.

Standard IBM OS linkage conventions are required. It is recommended that the exit be coded in assembler language. The exit should be linked into the EZ/KEY load module with a CSECT name of PIEXIT01.

Here is an example LINKEDIT JCL for TSO EZ/KEY:

```
//LINKEXIT EXEC PGM=IEWL,PARM=LKED='LET,LIST,MAP,NCAL,RENT,XREF'  
//SYSPRINT DD SYSOUT=*  
//SYSLMOD DD DSN=your.EZKEY.LOAD,DISP=SHR  
//SYSLIB DD DSN=your.EZKEY.LOAD,DISP=SHR  
//EXITLIB DD DSN=your.EXIT.OBJLIB,DISP=SHR  
//SYSLIN DD *  
    INCLUDE SYSLIB(EZKEYTSO)  
    INCLUDE EXITLIB(PIEXIT01)  
    ENTRY EZKEYTCP  
    NAME EZKEYTSO(R)  
/*  
//
```

INSTALLATION OF SUBMIT USER EXIT UNDER CICS/OS/VS

EZ/KEY user exits are referenced through weak external references (WXTRN). Therefore it is necessary to insert into the linkedit stream an INCLUDE statement for each desired exit.

Standard IBM OS linkage conventions are required. It is recommended that the exit be coded in assembler language. The exit should be linked into the EZ/KEY load module with a CSECT name of PIEXIT01.

Here is an example LINKEDIT JCL for CICS/OS/VS EZ/KEY:

```
//LINKEXIT EXEC PGM=IEWL,PARM,LKED='LET,LIST,MAP,NCAL,RENT,XREF'  
//SYSPRINT DD SYSOUT=*  
//SYSLMOD DD DSN=your.EZKEY.LOAD,DISP=SHR  
//SYSLIB DD DSN=your.EZKEY.LOAD,DISP=SHR  
//EXITLIB DD DSN=your.EXIT.OBJLIB,DISP=SHR  
//SYSLIN DD *  
    ORDER DFHEAI  
    ORDER PIESSCOO  
    INCLUDE SYSLIB(EZKYZOO)  
    INCLUDE EXITLIB(PIEXIT01)  
    ENTRY DFHEAI  
    NAME EZKYZOO(R)  
/*  
//
```

*Installation of SUBMIT User Exit under CICS/DOS/VSE***INSTALLATION OF SUBMIT USER EXIT UNDER CICS/DOS/VSE**

EZ/KEY user exits are referenced through weak external references (WXTRN). Therefore it is necessary to include in the linkedit stream the object deck for your PLEXIT01 exit routine.

You accomplish this by using a text editor (such as XEDIT under VM/CMS or ICCF) to locate the first card that reads " ENTRY DFHEAI". Insert the new object deck for your user exit immediately in front of this statement.

Standard IBM OS linkage conventions are required. It is recommended that the exit be coded in assembler language. The exit must be linked into EZ/KEY with a CSECT name of PLEXIT01.

LINK-EDITING CICS/DOS/VSE PHASES

Restore Relocatable Library from Installation Tape

Use the following JCL to restore the relocatable library from the EZ/KEY distribution tape. This JCL is contained in EXAMPLE PIELIB as member VSERELOC JCL:

```

* $$ JOB UNM=jobname,CLASS=A,DISP=D                ***-1-***
* $$ LST CLASS=A,DISP=D                            ***-1-***
// JOB EZKEY7                                CREATE RELO. LIBRARY FOR EZ/KEY ***-1-***
// ASSGN SYS003,xxx                            ***-2-***
// DLBL EZKEY,'your.EZKEY.relo.library',99/365,SD ***-2-***
// EXTENT SYS003,volser,1,0,nnnn,120           ***-2-***
// LIBDEF RL,NEW=EZKEY,TEMP                    ***-3-***
// EXEC CORGZ,SIZE=100K                        ***-3-***
NEWVOL RL=4(1)                                ***-3-***
/*
/&
// JOB EZKEY8                                RESTORE EZ/KEY RELO. LIBRARY ***-1-***
// DLBL EZKEY,'your.EZKEY.relo.library',.,SD   ***-2-***
// EXTENT ,volser                             ***-2-***
// ASSGN SYS004,yyy                            INPUT TAPE ***-4-***
// MTC FSF,SYS004,21                          8TH FILE (RELO. LIB BACKUP)
// EXEC RESTORE                                ***-5-***
RESTORE RL,LIB=EZKEY(1)                      ***-5-***
/*
// MTC RUN,SYS004
/&
* $$ EOU

```

Tailoring Relocatable Library Restore JCL

At the right of some of the JCL lines shown above is the character string *****-n-*****, where *n* is a number corresponding to one of the notes listed below. Modify the JCL as described in the notes:

1. Modify the JOB name, CLASS, and accounting information as needed.
2. Specify the name and location of the relocatable library for EZ/KEY.
3. For VSE/SP 2.1 and above only:
 - Delete the LIBDEF statement

Link-Editing CICS/DOS/VSE Phases

- Change CORGZ,SIZE=100K to LIBR
- Replace NEWVOL RL=4(1) with:


```
DEFINE LIB=EZKEY
DEFINE SUBLIB=EZKEY.RELO
```
- 4. Specify the device address for SYS004, the EZ/KEY distribution tape.
- 5. For VSE/SP 2.1 and above only:
 - Change RESTORE to LIBR
 - Replace RESTORE RL,LIB=EZOBJ(1) with:


```
RESTORE TAPE=SYS004,REPLACE=YES,OLDLIB=EZOBJ:EZKEY.RELO
```

Link-Edit Phase EZKYZ00

Use the following JCL to re-link the EZKYZ00 phase from the object contained in the EZ/KEY relocatable library. This JCL is contained in EXAMPLE PIELIB as member VSELKED0 JCL. If you want to run EZ/KEY under CICS/DOS/VSE 1.5, you must relink EZKYZ00 to obtain a copy of the 1.5 version of DFHEAI.

If you are installing a Submit User Exit into EZ/KEY, insert the following statement just before the "ENTRY DFHEAI" statement:

```
INCLUDE PIEXIT01
```

Before you run the link-edit job, you must have assembled your PIEXIT01 assembler language program, and placed its object into the relocatable library above, using LIBR or MAINT.

The JCL to re-link the EZKYZ00 phase is:

* \$\$ JOB UNM=jobname,CLASS=A,DISP=D	***-1-***
* \$\$ LST CLASS=A,DISP=D	***-1-***
// JOB EZKEYOO LINKEDIT EZKYZ00	***-1-***
// DLBL EZKEY,'your.EZKEY.phase.library',,SD	***-2-***
// EXTENT ,volser	***-2-***
// LIBDEF CL,TO=EZKEY	***-3-***
// DLBL EZKOBJ,'your.EZKEY.relo.library',,SD	***-4-***
// EXTENT ,volser	***-4-***
// DLBL C17OPRL,'your.IBM.CICS170.relo.library',,SD	***-5-***
// EXTENT ,volser	***-5-***
// LIBDEF RL,SEARCH=(C17OPRL,EZKOBJ)	***-6-***
// OPTION CATAL	

continued ...

... continued

```
ACTION CLEAR,MAP
PHASE EZKYZOO,*
INCLUDE DFHEAI
INCLUDE PIESSCOO
INCLUDE PIESSFIX
INCLUDE PIEDMAIN
.
.
.
INCLUDE AMPZQSAM
INCLUDE AMPZREAD
INCLUDE AMPZWRT
ENTRY DFHEAI
// EXEC LNKEDT
/&
* $$ EOJ
```

Tailoring CICS/DOS/VSE Link-Edit JCL

At the right of some of the JCL lines shown above is the character string `***-n-***`, where *n* is a number corresponding to one of the notes below. Modify the JCL as described in the notes:

1. Modify the JOB name, CLASS, and accounting information as needed.
2. Specify the name and location of the phase library for EZ/KEY.
3. For VSE/SP 2.1 and above only:
 - Change CL to PHASE
 - Change TO= to CATALOG=
4. Specify the name and location of the relocatable library for EZ/KEY.
5. If C170PRL, C160PRL, or C150PRL are defined in standard labels, you can delete these DLBL and EXTENT statements. If these statements are not defined in standard labels, specify the name and location of the relocatable library for CICS. Change C170PRL to C160PRL or C150PRL, as needed, for your release of CICS.
6. For VSE/SP 2.1 and above only, change RL to OBJ. Change C170PRL to C160PRL or to C150PRL, as needed, for your release of CICS.

Link-Edit Phase EZKYZ01

Use the following JCL to re-link the EZKYZ01 phase from the object contained in the EZ/KEY relocatable library. This JCL is contained in library EXAMPLE PIELIB as member VSELKED1 JCL. If you want to run EZ/KEY under CICS/DOS/VSE 1.5, you must relink EZKYZ01 to obtain a copy of the 1.5 version of DFHEAI.

It is not necessary to run this job if you are only installing a Submit User Exit.

The JCL to link-edit the EZKY01 phase is:

* \$\$ JOB JNM=jobname,CLASS=A,DISP=D	***-1-***
* \$\$ LST CLASS=A,DISP=D	***-1-***
// JOB EZKEY01 LINKEDIT EZKYZ01	***-1-***
// DLBL EZKEY,'your.EZKEY.phase.library',,SD	***-2-***
// EXTENT ,volser	***-2-***
// LIBDEF CL,TO=EZKEY	***-3-***
// DLBL EZKOBJ,'your.EZKEY.relo.library',,SD	***-4-***
// EXTENT ,volser	***-4-***
// DLBL C17OPRL,'your.IBM.CICS170.relo.library',,SD	***-5-***
// EXTENT ,volser	***-5-***
// LIBDEF RL,SEARCH=(C17OPRL,EZKOBJ)	***-6-***
// OPTION CATAL	
ACTION CLEAR,MAP	
PHASE EZKYZ01,*	
INCLUDE DFHEAI	
INCLUDE PIESCO1	
ENTRY DFHEAI	
// EXEC LNKEDT	
/&	

Tailoring VSELKED1 Link-Edit JCL

The procedure for tailoring VSELKED1 Link-edit JCL is the same as that for tailoring CICS/DOS/VSE link-edit JCL. That procedure is described in "Tailoring CICS/DOS/VSE Link-Edit JCL" on the previous page.

(This page intentionally left blank.)

APPENDIX B PANAUDIT PLUS CONSIDERATIONS

Introduction	B-1
Step 1: Change Members for Foreground and Back- ground Execution	B-1
Step 2: Place the members in EXAMPLE PIELIB	B-1
Step 3: Allocate the PANAUDIT PLUS Macro Library	B-1
Step 4: Instruct Users to CONNECT and ENABLE the Macro Library	B-1
Step 1: Change Members for Foreground and Background Execution	B-2
Changes to Make to PAPLUSO JCL	B-3
Changes to make to PAPLUSD JCL	B-4
Changes to Make to PAPLUS EXEC	B-5
Changes to Make to PAPLUS CLIST	B-7
Step 2: Place the Members in EXAMPLE PIELIB	B-10
Step 3: Allocate the PANAUDIT PLUS Macro Library	B-11
Changes to Make to EZKEY EXEC for CMS	B-11
Changes to Make to EZKEY CLIST for TSO	B-11
Changes to Make for CICS/OS/VS or CICS/DOS/VSE	B-12
Step 4: Instruct Users to CONNECT and ENABLE the Macro Library	B-13

(This page intentionally left blank.)

INTRODUCTION

You can use PANAUDIT PLUS with EZ/KEY. The following installation section describes the changes you must make to the PANAUDIT PLUS EXECs, CLISTs, and JCL that are supplied in the EZ/KEY EXAMPLE library in order to use PANAUDIT PLUS with EZ/KEY.

There are four steps you must perform to make PANAUDIT PLUS available to EZ/KEY. These steps are briefly described here and are explained in detail on the following pages.

Step 1: Change Members for Foreground and Background Execution

Modify PAPLUS CLIST, PAPLUS EXEC, PAPLUSD JCL, and PAPLUSO JCL for foreground and background execution.

Step 2: Place the members in EXAMPLE PIELIB

Place the modified CLIST, EXEC, and JCL members into EXAMPLE PIELIB to make them available to all EZ/KEY and PANAUDIT PLUS users.

Step 3: Allocate the PANAUDIT PLUS Macro Library

Depending on your operating system, change one of the following to allocate the PANAUDIT PLUS macro library:

- EZKEY EXEC for CMS
- EZKEY CLIST for TSO
- Your startup JCL for CICS

Step 4: Instruct Users to CONNECT and ENABLE the Macro Library

Instruct your users to CONNECT the PANAUDIT PLUS macro library to EZ/KEY and to ENABLE the PANAUDIT PLUS macro library.

*Step 1: Change Members for Foreground and Background Execution***STEP 1: CHANGE MEMBERS FOR FOREGROUND AND BACKGROUND EXECUTION**

The following discussion pertains to all of the EXECs, CLISTs, and JCL that you may need to tailor for use with EZ/KEY Foreground or Background execution.

Change the following EXAMPLE PIELIB members, depending on where EZ/KEY and PANAUDIT are installed:

- PAPLUS EXEC - If you are going to execute PANAUDIT PLUS programs online in CMS.
- PAPLUS CLIST - If you are going to execute PANAUDIT PLUS programs online in TSO.
- PAPLUSO JCL - If you are going to submit batch PANAUDIT PLUS programs to an MVS system.
- PAPLUSD JCL - If you are going to submit batch PANAUDIT PLUS programs to a VSE system.

When changing any member for use with EZ/KEY, place the tailored members in a *common* library, such as EXAMPLE PIELIB. When a user specifies the name of the member to use for Foreground or Background execution, EZ/KEY searches all enabled PIELIBs for a member with that name.

Follow these steps to tailor the EXECs, CLISTs, and JCL members:

1. Disable EXAMPLE PIELIB using option 3.2.
2. On the **PROULSM** panel, select all EXAMPLE members by typing an **S** next to **EXAMPLE**, and typing the member name under the **Member Name** column.
3. Edit a member by typing **ED** next to the member name.
4. After making your changes, press the END (PF3) key. This saves the member in your library.

*Step 1: Change Members for Foreground and Background Execution***Changes to Make to PAPLUSO JCL**

Use PAPLUSO JCL to submit batch PANAUDIT PLUS programs to MVS. Make the changes described below if you are installing EZ/KEY and PANAUDIT PLUS in TSO, CICS/OS, or CMS and submitting PANAUDIT PLUS programs to an MVS system. The PAPLUSO member is in EXAMPLE PIELIB and is shown here:

```
//%EZKEY1| JOB (%EZKEY2|,%EZKEY3|),%ZUSERID|,MSGCLASS=%EZKEY4|.
//          CLASS=A,REGION=512K
//* %CPGMNAM| SUBMITTED ON %ZDATE| AT %ZTIME|
//EZTPLUSO EXEC PGM=PAPLAOO
//STEPLIB DD DSN=your.PANAUDIT.PLUS.loadlib,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSSNAP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//PANDD DD DSN=your.PAPL.macro.library,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,1)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,1)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,1)
//PAPVFM DD UNIT=SYSDA,SPACE=(4096,(100,100),,ROUND)
//%DDNAM] DD DSN=%DSNAM],DISP=SHR
//%DDNAM2] DD DSN=%DSNAM2],DISP=SHR
//SYSIN DD *
```

Exhibit B.1: PAPLUSO JCL

To tailor PAPLUSO JCL, use the EZ/KEY editor (=2) and make the following changes:

1. Type the name of the PANAUDIT PLUS load library on the STEPLIB DD statement.
2. Type the name of the PANAUDIT PLUS macro library on the PANDD DD statement.
3. Review the JCL and make any additional changes to make the JCL conform to your installation's requirements.

*Step 1: Change Members for Foreground and Background Execution****Changes to make to PAPLUSD JCL***

Use PAPLUSD JCL to submit batch PANAUDIT PLUS programs to VSE. Make the changes described below if you are installing EZ/KEY and PANAUDIT PLUS in CICS/VSE—or CMS, and submitting PANAUDIT PLUS programs to a VSE system. The PAPLUSD member is in EXAMPLE PIELIB and is shown here:

```
* $$ JOB JNM=%EZKEY1|,CLASS=A,USER='%ZUSERID|',NTFY=(RSCS,%ZUSERID))
* $$ PRT CLASS=%EZKEY4|,DEST=(RSCS,%ZUSERID|)
* $$ LST CLASS=%EZKEY4|,DEST=(RSCS,%ZUSERID|)
// JOB %EZKEY1| %EZKEY2|
*
* %CPGMNAM| SUBMITTED ON %ZDATE| AT %ZTIME|
*
// ASSGN SYS003,...
// DLBL PAPL,'your.PAPL.sysc1b',O,SD
// EXTENT SYS003,volser,1,O,start,lgth
// LIBDEF CL,SEARCH=PAPL,TEMP-
// ASSGN SYS010,...
// DLBL PANDD,'your.PAPL.macro.library',O,SD
// EXTENT SYS010,volser,,,start,lgth
// ASSGN SYS001,...
// DLBL SORTWK1,,O,DA
// EXTENT SYS001,volser,,,start,lgth
// DLBL SORTWK2,,O,DA
// EXTENT SYS001,volser,,,start,lgth
// DLBL SORTWK3,,O,DA
// EXTENT SYS001,volser,,,start,lgth
// ASSGN SYS011,...
// DLBL %DDNAM|,'%DSNAM|',O,SD
// EXTENT SYS011,volser,1,O,start,lgth
// ASSGN SYS012,...
// DLBL %DDNAM2|,'%DSNAM|',O,SD
// EXTENT SYS012,volser,1,O,start,lgth
// ASSGN SYS008,...
// DLBL PAPVFM,,O,SD
// EXTENT SYS008,volser,,,start,lgth
// EXEC PAPLA00,SIZE=384K
```

Exhibit B.2: PAPLUSD JCL

Step 1: Change Members for Foreground and Background Execution

To tailor PAPLUSD JCL, use the EZ/KEY editor (=2) and make the following changes:

1. Type the name of the PANAUDIT PLUS core library on the PAPL DLBL statement.
2. Type the name of the PANAUDIT PLUS macro library on the PANDD DLBL statement.
3. Type the correct volser, start, and lengths on each of the EXTENT cards.
4. Review the JCL and make any additional changes to make the JCL conform to your installation's requirements.

Changes to Make to PAPLUS EXEC

Use PAPLUS EXEC to execute PANAUDIT PLUS programs online in CMS. Make the changes described below if you are installing EZ/KEY and PANAUDIT PLUS in CMS. The PAPLUS member is in EXAMPLE PIELIB and is shown here:

```

&TRACE
  &PAPLPGM = PAPLAOO
  &PAPLLIB = PAPLTX
  &PAPLSRT = SORTLIB
*
* LINK and ACCESS any sequential, VSAM, SQL/DS or other data base disk(s)
*
* CP LINK userid mmm nnn RR
* ACCESS nnn m
* ...add more LINKs and ACCESSs here as needed...
* Set up FILEDEFS for all files needed by PANAUDIT PLUS program
*
FILEDEF SYSPRINT CLEAR
&IF .&5 NE .P &SKIP 1
  FILEDEF SYSPRINT PRINTER
&IF .&5 NE .T &SKIP 1
  FILEDEF SYSPRINT TERMINAL
&IF .&5 EQ .B &SKIP 1
&IF .&5 NE .D &SKIP 1
  FILEDEF SYSPRINT DISK &1 LISTING A
&IF .&2 = .2 &SKIP 1
  FILEDEF &2 DISK &3 &4 *
```

continued ...

Step 1: Change Members for Foreground and Background Execution

```
... continued
&IF .&6 = .6 &SKIP 1
    FILEDEF &6 DISK &7 &8 *
FILEDEF PANDD DISK PAPLMAC MACLIB
FILEDEF SYSOUT CLEAR
FILEDEF SYSOUT DISK SYSOUT DATA A
FILEDEF SYSIN CLEAR
FILEDEF SYSIN DISK &1 &9 *
*   ... add any additional FILEDEFs here as needed...
*
* Set up TXTLIBs, clear the screen and invoke PANAUDIT PLUS
*
    GLOBAL TXTLIB & PAPLLIB &PAPLSRT
    VMFCLEAR
    SET CMSTYPE RT
    EXECOS &PAPLPGM
    SET CMSTYPE HT
    CP SET IMSG OFF
*
* Erase the PANAUDIT PLUS program source
*
    ERASE &1 &9
*
* Release any sequential, VSAM, or other data base disk(s)
*
    RELEASE nnn ( DET
*   ... add more RELEASEs here as needed...
*
* Release any minidisks, reset TXTLIBs, and return
*
-EXIT
    GLOBAL TXTLIB
    &EXIT
```

Exhibit B.3: PAPLUS EXEC

Step 1: Change Members for Foreground and Background Execution

Tailor the PAPLUS CLIST to your installation, using the symbolic substitution parameter as follows:

- &PAPLLIB - The name of the PANAUDIT PLUS TXTLIB to use with EZ/KEY for foreground execution. The default name is PAPLTX. Change this to the appropriate name for your installation.
- &PAPLSRT - The name of the SORT TXTLIB, if SORTs are used in PANAUDIT PLUS programs using foreground execution with EZ/KEY. Otherwise, leave blank (null). The default name is SORTLIB. Change this to the appropriate name for your installation.

Then review the EXEC and make any additional changes to conform the EXEC to your installation's requirements.

Changes to Make to PAPLUS CLIST

Use PAPLUS CLIST to execute PANAUDIT PLUS programs on-line in TSO. Make the changes described below if you are installing EZ/KEY and PANAUDIT PLUS in TSO. The PAPLUS member is in EXAMPLE PIELIB and is shown here:

```

PROC 7 PGM DDNAM DSNAM TYP DDNAM2 DSNAM2 PGMTYP
CONTROL NOFLUSH NOMSG
/*-----*/
/* Invoke PANAUDIT PLUS as a CALLED program in the foreground. */
/*
/*   This CLIST is invoked via '&EZKREF..PIETEMP.CLIST',
/*   which is invoked by the EZKEY CLIST.
/*
/* Invoking Parameters:
/*
/*   PGM      = the name of the PANAUDIT PLUS program to execute
/*
/*   DDNAM    = the DDNAME of the primary input file for the above
/*
/*   DSNAM    = the DSNAM of the primary input file for the above
/*
/*   TYP      = the type of output: 'B' for browse,
/*                                     'D' for disk dataset,
/*                                     'P' for printer, or
/*                                     'T' for terminal.
/*
/*

```

continued ...

Step 1: Change Members for Foreground and Background Execution

... continued

```

/*      DDNAM2 = the DDNAME of the second input file for the above      */
/*      DSNAM2 = the DSNAM of the second input file for the above      */
/*      PGM TYP = the type of the program input file, usually 'EZT'    */
/* Installation Tailorable symbolic definitions:                        */
/*      &EZKPREF  = PREFIX for EZ/KEY TSO sequential datasets          */
/*                  (must be the same as &EZKPREF in EZKEY.CLIST)      */
/*      &EZTPDSN  = DSNAM of installation's PANAUDIT PLUS library      */
/*      &EZTCLASS = sysout class for PANAUDIT PLUS printed output      */
/* modify the following 3 lines based on the above definitions:        */
/*-----*/
SET &EZKPREF  = &SYSPREF
SET &PAPLDSN  = your.PAPLUS.LOADLIB
SET &EZTCLASS = A
FREE F(SYSIN, SYSPRINT, SYSOUT, PAPVFM)
IF      &TYP = T THEN ALLOC F(SYSPRINT) DA(*)
ELSE IF &TYP = P THEN ALLOC F (SYSPRINT) SYSOUT(&EZTCLASS)
ELSE IF &TYP = B ] &TYP = D THEN DO
    DELETE '&EZKPREF..&PGM..LIST'
    FREE ATTRLIST(EZKLATTR)
    ATTR EZKLATTR RECFM(F B A) LRECL(133) BLKSIZE(3990)
    ALLOC F(SYSPRINT) DA('&EZKPREF..&PGM..LIST') NEW +
        SPACE(20 100) BLOCK(3990) USING(EZKLATTR)
    END
ELSE ALLOC F(SYSPRINT) DA(*)
IF &DDNAM 1 = 2 THEN DO
    IF &DSNAM = 3 THEN SET &DSNAM = NULLFILE
    FREE F(&DDNAM)
    ALLOC F(&DDNAM) DA(&DSNAM) SHR
    END
IF &DDNAM2 1 = 6 THEN DO
    IF &DSNAM2 = 7 THEN SET &DSNAM2 = NULLFILE
    FREE F(&DDNAM2)
    ALLOC F(&DDNAM2) DA(&DSNAM2) SHR
    END
ALLOC F(SYSOUT) DA(*)
ALLOC F(PAPVFM) TR SP(5 5)

```

continued ...

Step 1: Change Members for Foreground and Background Execution

... continued

```

IF &PGMTYP = 9 THEN SET &PGMTYP = EZT
IF &PGM = 1 THEN ALLOC F(SYSIN) DUMMY
      ELSE ALLOC F(SYSIN) DA('&EZKPREF..&PGM..&PGMTYP') SHR
CALL '&PAPLDSN.(PAPLAOO) ' ' '
FREE F(SYSIN, SYSPRINT, SYSOUT, PAPVFM)
IF &DDNAM  = 2 THEN FREE F(&DDNAM)
IF &DDNAM2 = 6 THEN FREE F(&DDNAM2)
IF &PGM    = 1 THEN DELETE '&EZKPREF..&PGM..&PGMTYP'
EXIT

```

Exhibit B.4: PAPLUS CLIST

Tailor the PAPLUS CLIST to your installation, using the symbolic substitution parameters as follows:

- &EZKPREF - The high-level qualifier used to prefix all sequential datasets created dynamically on behalf of the user by EZ/KEY, and used to copy sequential datasets from TSO into EZ/KEY. This must be the same qualifier as that specified in the EZKEY CLIST.
- &PAPLDSN - The PANAUDIT PLUS load library where the PANAUDIT PLUS program modules reside.
- &EZTCLASS - The SYSOUT class for EZ/KEY printed output. The default is A. Specify a sysout class that is valid for your installation.

Review the CLIST and make any additional changes to make the CLIST conform to your installation's requirements.

*Step 2: Place the Members in EXAMPLE PIELIB***STEP 2: PLACE THE MEMBERS IN EXAMPLE PIELIB**

To make the tailored members always available to your EZ/KEY users for Foreground or Background execution, **EXAMPLE** must be enabled. You enable **EXAMPLE PIELIB** in *one* of two ways:

- Instruct your users to enable **EXAMPLE** using utility option 3.2.

Or

- Use the Report Administrator utility to have EZ/KEY automatically enable **EXAMPLE** for all or selected users. For detailed instructions on how to do this, see Section 7 of the *EZ/KEY Administrator's Guide*. Read the subsection that explains how to establish default **CONNECT** and **LIBCHAIN** members.

1. Use utility option 3.a.3 to **CONNECT** **EXAMPLE PIELIB** with **CREATE** access rights. Type **EXAMPLE** for the **Libspace Name**, **CREATE** for **Access Rights**, and one of the following for **External Name**:

Leave blank - for CMS
EXAMPLE - for TSO
EZKXMPL - for CICS/OS or CICS/VSE

Press **ENTER** after filling in all the information.

2. Use utility option 3.3 to **COPY** the modified member(s) from your library to the **EXAMPLE** library. Specify **YES** for the **Replace** option.

When you are copying either **PAPLUSO JCL** or **PAPLUSD JCL**, rename it to **PAPLUS JCL**. This allows users to use **PAPLUS** as the JCL header name.

*Step 3: Allocate the PANAUDIT PLUS Macro Library***STEP 3: ALLOCATE THE PANAUDIT PLUS MACRO LIBRARY*****Changes to Make to EZKEY EXEC for CMS***

To make the PANAUDIT PLUS macros available to EZ/KEY you must do one of two things:

- Install PANAUDIT PLUS on the same CMS minidisk as EZ/KEY. If you do this, there are no additional changes you need to make to EZKEY EXEC.

Or

- If PANAUDIT PLUS and EZ/KEY are installed on different CMS minidisks, add

```
LINK paplus 123 456 rr  
ACCESS 456 f
```

after the **ACCESS &EZKEYVUU &EZKEYMOD** statement.

The parameters used in the LINK and ACCESS commands are:

```
paplus - the userid that owns the minidisk that contains  
        the PANAUDIT PLUS macros  
123    - the device address of the minidisk  
456    - virtual device address  
f      - filemode
```

In either case, review the EXEC and make any additional changes to conform the EXEC to your installation's requirements.

Changes to Make to EZKEY CLIST for TSO

To make the PANAUDIT PLUS macros available to EZ/KEY, after the **FREE ATTRLIST (EZKEYPRT)** statement, add

```
FREE F(PANDD)  
ALLOC F(PANDD) DA('your.PAPL.macro.library') SHR
```

The parameter **your.PAPL.macro.library** used in the ALLOC statement is the PANAUDIT PLUS macro library name.

Then review the CLIST and make any additional changes to make the CLIST conform to your installation's requirements.

If the macro library is a Panvalet library, change PANDD to PANDD1.

*Step 3: Allocate the PANAUDIT PLUS Macro Library****Changes to Make for CICS/OS/VS or CICS/DOS/VSE***

To make the PANAUDIT PLUS macro library available to EZ/KEY, you must add certain statements, depending on the library's type:

- If your macro library is a VSAM library:
 - Add the following DD statement to your CICS/OS/VS startup JCL:

```
//PANDD DD DISP=SHR,DSN=your.PAPL.macro.library
```

- Add the following DLBL statement to your CICS/DOS/VSE startup JCL:

```
// DLBL PANDD,'your.PAPL.macro.library',,VSAM
```

- Add the following FCT entry:

```
DFHFCT TYPE=DATASET,DATASET=PANDD,          C
      SERVREQ=(BROWSE,GET),BUFFND=2,         C
      BUFFNI=1,STRNO=1,ACCMETH=(VSAM,KSDS)
```

This FCT entry is needed because CICS File Control OPENs, READs, and CLOSEs are used to access EZTMACRO libraries.

- If your macro library is a PANVALET library, add the following DLBL statement to your CICS/DOS/VSE startup JCL:

```
// DLBL PANDD1,'your.PAPL.macro.library',,DA
```

or add the following DD statement to your CICS/OS/VS startup JCL:

```
//PANDD1 DD DISP=SHR,DSN=your.PAPL.macro.library
```

- If the macro library is a PDS library, add the following DD statement to your CICS/OS/VSE startup JCL:

```
//PANDD DD DISP=SHR,DSN=your.PAPL.macro.library
```

FCT entries are required only for VSAM libraries. They are not required for any other library type.

Step 4: Instruct Users to CONNECT and ENABLE the Macro Library

STEP 4: INSTRUCT USERS TO CONNECT AND ENABLE THE MACRO LIBRARY

After the PANAUDIT PLUS macro library has been made available to EZ/KEY and the necessary EXECs, CLISTs, and JCL members have been modified for Foreground and Background execution, instruct your users to perform the following steps. These steps CONNECT and ENABLE the macro library.

Note that for CICS, the CONNECT step is performed *once* by the System Administrator for the entire EZ/KEY system:

1. CONNECT PANDD or PANDD1, using options 3.a.4 on the External LIBSPACE Utility panel, shown below:

PROULVC

External LIBSPACE Utility

EZ/KEY

COMMAND ==>

Describe the External LIBSPACE to Connect

LIBSPACE Name ==> pandd

LIBSPACE Type ==> xxx (MINIDISK, CMSMAC, PDS, EZTMAC) (CMS)

(Access Rights) ==> read (READ, WRITE, CREATE)

(External Name) ==> yyy

(External Password) ==>

Provide the required information and press ENTER.
Press END (PF3) to cancel and return to the previous panel.

Exhibit B.5: The External LIBSPACE Utility Panel

Step 4: Instruct Users to CONNECT and ENABLE the Macro Library

Depending on your system, use the following values:

For **xxx** Type of Library

CMSMAC	- CMS Macro Library
PDS	- Partitioned Dataset
EZTMAC	- EASYTRIEVE PLUS VSAM Macro Library
PANVALET	- Panvalet Macro Library

For **yyy**, type **PANDD1** if the library is a Panvalet library. Otherwise, use **PANDD** for CMS, PDS, or VSAM macro libraries.

2. **ENable PANDD** using option 3.2.

In place of these two steps, you may use **EZ/REPORT** to automatically **CONNECT** and **ENable** the **PANAUDIT PLUS** macro library for your users. You enable **EXAMPLE PIELIB** in *one* of two ways:

- Instruct your users to enable **EXAMPLE** using utility option 3.2.

Or

- Use the Report Administrator utility to have **EZ/KEY** automatically enable **EXAMPLE** for all or selected users. For detailed instructions on how to do this, see Section 7 of the *EZ/KEY Administrator's Guide*. Read the subsection that explains how to establish default **CONNECT** and **LIBCHAIN** members.

A

Additional IDMS Considerations
2-74 to 2-86, 3-45 to 3-49,
5-59 to 5-62

Additional IDMS/IDD CICS Considerations 4-49 to 4-52

Additional SQL CICS Considerations
5-51 to 5-53

B

blocks 2-6, 2-29, 3-5
blocksize 2-6, 3-1, 4-14, 4-25, 5-14

C

Calling Sequence A-8
Choosing the Size of PIELIBs 3-21,
4-14, 5-14
CICS 4-14, 5-14
CICS Runaway Task Interval
4-19 to 4-20, 5-20 to 5-21
CICS sign-on security 4-32, 5-29
CICS/DOS/VSE INSTALLATION
5-1 to 5-62
CICS/DOS/VSE Table Changes
5-15 to 5-19
CICS/OS/VS INSTALLATION
4-1 to 4-52
CICS/OS/VS Submit to Background
4-25 to 4-27
CICS/OS/VS Table Changes
4-15 to 4-18
CISIZE 4-14, 5-14
CMS 4-14, 5-14
CMS INSTALLATION 2-1 to 2-88
CMS User Area 2-28, 2-87
CONNECT
member 4-2, 5-2
Customer Service Interface 6-2

D

DB2 Userids 4-42
DCSS 2-27, 2-87

DMKSNT 2-31

DMKSYS ASSEMBLE 2-37

E

EASYTRIEVE PLUS Under EZ/KEY
2-52
EASYTRIEVE PLUS VSAM Macro
Libraries 3-29, 2-53 to 2-55
EASYTRIEVE PLUS VSAM Macro
Libraries for CICS/DOS/VSE 5-24
EASYTRIEVE PLUS VSAM Macro
Libraries for CICS/OS/VS 4-23
EZ/KEY Foreground Execution CLIST
Considerations 3-18 to 3-20
EZ/KEY SQL Support 2-57 to 2-61
EZKEY CLIST 3-10 to 3-14
EZKSEgn Messages 2-87 to 2-88
EZTMAC 3-29, 4-23, 5-24
EZTPLUS CLIST 3-15 to 3-17

G

GENERAL INFORMATION 1-1 to 1-4

I

ICCF Considerations for CICS/DOS/VSE
5-22
Installation Files 2-5, 3-4 to 3-6,
4-4 to 4-8, 5-4 to 5-7
Installation in a DCSS 2-27 to 2-51
Installation in an MVS/XA Environment
4-21
Installation JCL 3-7 to 3-9, 4-9 to 4-13,
5-8 to 5-13
Installation of SUBMIT User Exit under
CICS/DOS/VSE A-13
Installation of SUBMIT User Exit under
CICS/OS/VS A-12
Installation of SUBMIT User Exit under
CMS A-9 to A-10
Installation of SUBMIT User Exit under
TSO A-11
INSTALLATION SUBMIT USER EXIT
A-1 to A-17

Installation Without DCSS 2-6 to 2-26
 Installing CMS SQL Support
 2-62 to 2-67
 Installing EZ/KEY Under ISPF/PDF
 2-56, 3-37

L

Link Pack Area 3-1, 3-26
 Link-Editing CICS/DOS/VSE Phases
 A-14 to A-17

M

MAINTENANCE 6-1 to 6-5
 Maintenance Procedures 6-3 to 6-5
 minidisk 2-38, 2-47, 2-88
 Multiple EZ/KEY Systems Under One
 CICS 4-31 to 4-32, 5-28 to 5-29

N

NAMESYS 2-31

O

Optional IDMS/IDD Installation 3-44,
 2-70 to 2-73, 4-43 to 4-48,
 5-54 to 5-58
 Optional SQL Installation Procedures
 3-38 to 3-43, 4-33 to 4-41
 Optional TSO Installation Procedures
 3-22 to 3-26

P

PANAUDIT PLUS CONSIDERATIONS
 B-1 to B-14
 PANVALET Considerations for
 CICS/DOS/VSE 5-25 to 5-27
 PANVALET Considerations for
 CICS/OS/VS 4-28 to 4-30
 PANVALET Considerations for TSO
 3-30 to 3-33
 Parameter Lists A-6 to A-7

PDS Macro Libraries for CICS/OS/VS
 4-24
 PIEBDAM and PIEVSAM 4-14, 5-14
 PIEVSAM 4-14, 5-14
 PROSTART 2-26, 2-50, 3-27, 4-22,
 5-23, 6-2

R

reconnect
 procedures 4-2, 5-2
 Reinstalling EZ/KEY under
 CICS/DOS/VSE 5-2 to 5-3
 Reinstalling EZ/KEY under CICS/OS/VS
 4-2 to 4-3
 Reinstalling EZ/KEY under CMS
 2-3 to 2-4
 Reinstalling EZ/KEY under TSO 3-3

S

SQL Interface Installation - DOS/VSE
 (pre-SP 2.1) 5-41 to 5-50
 SQL Interface Installation - VSE/SP 2.1
 5-32 to 5-40
 SQL Interface Option Installation
 5-30 to 5-31
 SQL/DS Userids 2-69
 Step 1: Change Members for Foreground
 and Background Execution B-2 to B-9
 Step 2: Place the Members in
 EXAMPLE PIELIB B-10
 Step 3: Allocate the PANAUDIT PLUS
 Macro Library B-11 to B-12
 Step 4: Instruct Users to CONNECT
 and ENABLE the Macro Library
 B-13 to B-14

T

TSO INSTALLATION 3-1 to 3-49
 TSO/TCAM Considerations 3-34 to 3-35
 TSO/VTAM Considerations 3-36

U

Using SQL with EZ/KEY 2-68
Using VSAM PIELIBs with EZ/KEY
3-28

V

Validate CICS Installation 4-22, 5-23
Validate TSO Installation 3-27

(This page intentionally left blank.)

