AllFusion™ Endevor® Change Manager

Exits Guide

4.0

Computer Associates™

ENEXI400

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

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation “as is” without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

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

Provided with “Restricted Rights” as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

First Edition, August 2002

© 2002 Computer Associates International, Inc. (CA)
All rights reserved.

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

Contents

Chapter 1. Using Exits with Endevor 1-1
1.1.1 Writing and Storing Exits 1-2
1.1.2 Exit Names 1-2
1.1.3 High-Level Language Support 1-2
1.1.3.1 Defining Multiple COBOL Programs for the Same Exit Point . . . 1-3
1.1.3.2 Sample COBOL Linkage Section 1-3
1.1.4 Storing User Exits 1-3
1.1.5 Debugging Exit Programs 1-3
1.1.6 Sample Exit Trace Log 1-4
1.2 Identifying Exits to Endevor 1-16
1.2.1 Overview 1-16
1.2.2 BC1JXITS Job 1-16
1.2.3 @C1UEXIT Macro Statements 1-17
1.2.4 @C1UEXIT TYPE=START... 1-17
1.2.5 @C1UEXIT
EXIT#=number,NAME=program,ANCHID=id,AUTH=yes|no 1-18
1.2.6 @C1UEXIT TYPE=END 1-19
1.2.7 Using Anchor IDs with Multiple User Exits 1-19
1.2.8 Example 1-19
1.3 Documentation Overview 1-21
1.4 Syntax for Long File and Path Names 1-22
1.4.1 HFSFile Syntax Rules 1-22
1.4.2 Path Name Syntax Rules 1-23
1.4.3 Element Name Syntax Rules 1-23
1.4.4 SCL Continuation Syntax Rules 1-23

Chapter 2. Exit Point Descriptions 2-1
2.1 Overview 2-2
2.1.1 Exit Point Functions 2-2
2.2 Exit Point 1: After Endevor Security Checks 2-3
2.2.1 Overview 2-3
2.2.2 How to Use Exit 1 Programs 2-3
2.2.3 Assembler Exit Control Blocks 2-3
2.2.4 COBOL Exit Control Blocks 2-3
2.2.5 Exit Control Block Return Codes 2-4
2.3 Exit Point 2: Before Actions 2-5
2.3.1 Overview 2-5
2.3.2 How to Use Exit 2 Programs 2-5
2.3.3 Assembler Control Blocks Passed 2-5

2.3.4 ENVFCBA Field in Assembler 2-7
2.3.5 COBOL Control Blocks 2-7
2.3.6 COBOL Source and Target Fields 2-9
2.3.7 Return Codes 2-9
2.4 Exit Point 3: After Actions 2-11
2.4.1 Overview 2-11
2.4.2 How to Use Exit 3 Programs 2-11
2.4.3 Assembler Control Blocks Passed 2-11
2.4.4 ENVFCBA Field 2-12
2.4.5 COBOL Control Blocks Passed 2-12
2.4.6 COBOL Source and Target Fields 2-13
2.4.7 Return Codes 2-14
2.5 Exit Point 4: Data Set Validation 2-15
2.5.1 Overview 2-15
2.5.2 How to Use Exit 4 Programs 2-15
2.5.3 Assembler Control Blocks Passed 2-15
2.5.4 ENVFCBA Assembler Field 2-16
2.5.5 COBOL Control Blocks Passed 2-16
2.5.6 COBOL Source and Target Fields 2-17
2.5.7 Return Codes 2-17
2.6 Exit Point 5: At Session Initialization 2-19
2.6.1 Overview 2-19
2.6.2 How to Use Exit 5 Programs 2-19
2.6.3 Assembler Control Blocks Passed 2-19
2.6.4 COBOL Control Blocks Passed 2-20
2.6.5 Return Codes 2-20
2.7 Exit Point 6: Before Session Termination 2-21
2.7.1 Overview 2-21
2.7.2 How to Use Exit 6 Programs 2-21
2.7.3 Assembler Control Blocks Passed 2-21
2.7.4 COBOL Control Blocks Passed 2-21
2.7.5 Return Codes 2-22
2.8 Example of Exit Use 2-23
2.8.1 When Endevor Calls Exit Programs 2-23
2.9 User Exit Message Handling 2-24
2.9.1 Coding Exit Programs to Fill in Fields 2-24
2.9.1.1 Assembler Coding 2-24
2.9.1.2 COBOL Coding 2-24
2.9.2 Building Messages 2-25
2.9.3 Issuing Messages 2-25

Chapter 3. Exit Control Blocks 3-1
3.1 Overview 3-2
3.2 Primary Information Block 3-4
3.2.1 $ECBDS DSECT Sample 3-4
3.2.2 Assembler Field Descriptions 3-5
3.2.3 EXIT-CONTROL-BLOCK control block layout sample 3-11
3.2.4 COBOL Field Descriptions 3-13
3.3 Request Information Block 3-20
3.3.1 $REQPDS DSECT sample 3-20
3.3.2 $REQPDS DSECT Field Descriptions 3-20

3.3.3 REQUEST-INFO-BLOCK control block layout sample 3-23
3.3.4 COBOL REQUEST-INFO-BLOCK Field Descriptions 3-24
3.4 Assembler Environment Block 3-31
3.4.1 $ENVDS DSECT sample 3-31
3.4.2 $ENVDS DSECT Field Descriptions 3-32
3.5 COBOL Environment Block 3-35
3.5.1 XXX-ENVIRONMENT-BLOCK control block layout sample 3-35
3.5.2 SRC-ENVIRONMENT-BLOCK Field Descriptions 3-36
3.5.3 TGT-ENVIRONMENT-BLOCK Field Descriptions 3-39
3.6 Assembler Element Master Information Block 3-42
3.6.1 $ELMDS DSECT sample 3-42
3.6.2 $ELMDS DSECT Field Descriptions 3-45
3.7 COBOL Element Master Information Block 3-53
3.7.1 ELEMENT-MASTER-INFO-BLOCK control block layout sample . . 3-53
3.7.2 SRC-ELEMENT-MASTER-INFO-BLOCK Field Descriptions 3-57
3.7.3 TGT-ELEMENT-MASTER-INFO-BLOCK Field Descriptions 3-64
3.8 Assembler File Block 3-72
3.8.1 $FILDS DSECT sample 3-72
3.8.2 $FILDS DSECT Field Descriptions 3-72
3.9 COBOL File Block 3-74
3.9.1 XXX-FILE-CONTROL-BLOCK Control block layout sample 3-74
3.9.2 SRC-FILE-CONTROL-BLOCK Field Descriptions 3-74
3.9.3 TGT-FILE-CONTROL-BLOCK Field Descriptions 3-76
3.10 SMF Action Record 3-77
3.10.1 Overview 3-77

Chapter 4. Package Exits 4-1
4.1 Overview 4-2
4.1.1 Package Processing 4-2
4.2 Writing Exit Programs for Package Actions 4-3
4.3 Package Exit Points 4-4
4.3.1 With Package Actions and Functions 4-4
4.4 Enabling Exit Points 4-5
4.4.1 Assembler Exit Points 4-5
4.4.1.1 $PECBDS Fields 4-5
4.4.1.2 Sample SETUP Routine 4-8
4.4.2 COBOL Exit Points 4-8
4.4.2.1 PACKAGE-EXIT-BLOCK Fields 4-9
4.4.2.2 Sample SETUP Routine 4-11
4.5 Requests for Data 4-12
4.5.1 Record-Level Interface 4-12
4.5.2 How Requests for Data Work 4-12
4.5.3 Assembler Request Flags and Return Codes 4-13
4.5.4 COBOL Request Flags and Return Codes 4-14
4.5.5 Availability of Data Records 4-15
4.5.5.1 Assembler - Availability of Data Records 4-16
4.5.5.2 COBOL - Availability of Data Records 4-16
4.6 Description of Package Exit Actions 4-18
4.6.1 Package Exit Return Codes 4-18
4.6.1.1 Archive 4-19

4.6.1.2 Possible Uses 4-19
4.6.1.3 Assembler Control Blocks Passed 4-19
4.6.1.4 COBOL Control Blocks Passed 4-19
4.6.1.5 Data Resources 4-20
4.6.2 Backin 4-20
4.6.2.1 Possible Uses 4-20
4.6.2.2 Assembler Control Blocks Passed 4-21
4.6.2.3 COBOL Control Blocks Passed 4-21
4.6.2.4 Data Resources 4-21
4.6.3 Backout 4-22
4.6.3.1 Assembler Control Blocks Passed 4-22
4.6.3.2 COBOL Control Blocks Passed 4-22
4.6.3.3 Data Resources 4-23
4.6.4 Cast 4-23
4.6.4.1 Possible Uses 4-23
4.6.4.2 Assembler Control Blocks Passed 4-24
4.6.4.3 COBOL Control Blocks Passed 4-24
4.6.4.4 Data Resources 4-25
4.6.5 Commit 4-26
4.6.5.1 Possible Uses in Assembler 4-26
4.6.5.2 Possible Uses in COBOL 4-26
4.6.5.3 Data Resources 4-27
4.6.6 Create 4-27
4.6.6.1 Possible Uses 4-27
4.6.6.2 Assembler Control Blocks Passed 4-28
4.6.6.3 COBOL Control Blocks Passed 4-28
4.6.6.4 Data Resources 4-29
4.6.7 Delete 4-29
4.6.7.1 Possible Uses 4-29
4.6.7.2 Assembler Control Blocks Passed 4-30
4.6.7.3 COBOL Control Blocks Passed 4-30
4.6.7.4 Data Resources 4-30
4.6.8 Display 4-31
4.6.8.1 Possible Uses 4-31
4.6.8.2 Assembler Control Blocks Passed 4-31
4.6.8.3 COBOL Control Blocks Passed 4-31
4.6.8.4 Data Resources 4-32
4.6.9 Execute 4-32
4.6.9.1 Possible Uses 4-32
4.6.9.2 Assembler Control Blocks Passed 4-32
4.6.9.3 COBOL Control Blocks Passed 4-33
4.6.9.4 Data Resources 4-33
4.6.10 Export 4-34
4.6.10.1 Possible Uses 4-34
4.6.10.2 Assembler Control Blocks Passed 4-34
4.6.10.3 COBOL Control Blocks Passed 4-34
4.6.10.4 Data Resources 4-35
4.6.11 GENPKGID 4-35
4.6.11.1 Possible Uses 4-36
4.6.11.2 Assembler Control Blocks Passed 4-36
4.6.11.3 COBOL Control Blocks Passed 4-36

4.6.11.4 Data Resources 4-37
4.6.12 List 4-37
4.6.12.1 Possible Uses 4-37
4.6.12.2 Assembler Control Blocks Passed 4-38
4.6.12.3 COBOL Control Blocks Passed 4-38
4.6.12.4 Data Resources 4-38
4.6.13 Modify 4-39
4.6.13.1 Possible Uses 4-39
4.6.13.2 Assembler Control Blocks Passed 4-39
4.6.13.3 COBOL Control Blocks Passed 4-40
4.6.13.4 Data Resources 4-41
4.6.14 Reset 4-41
4.6.14.1 Possible Uses 4-41
4.6.14.2 Assembler Control Blocks Passed 4-42
4.6.14.3 COBOL Control Blocks Passed 4-42
4.6.14.4 Data Resources 4-42
4.6.15 Review 4-43
4.6.15.1 Possible Uses 4-43
4.6.15.2 Assembler Control Blocks Passed 4-43
4.6.15.3 COBOL Control Blocks Passed 4-43
4.6.15.4 Data Resources 4-44
4.6.16 Ship 4-44
4.6.16.1 Possible Uses 4-44
4.6.16.2 Assembler Control Blocks Passed 4-45
4.6.16.3 COBOL Control Blocks Passed 4-45
4.6.16.4 Data Resources 4-45

Chapter 5. Package Shipment Exit 5-1
5.1 Assembler Calling Conventions 5-2
5.2 Assembler Example 5-3
5.3 COBOL Calling Conventions Example 5-4
5.4 Other Design Considerations 5-5
5.5 Sample 5-6

Chapter 6. Package Exit Control Blocks 6-1
6.1 Control Block Summaries 6-3
6.1.1 Control Blocks 6-4
6.2 Control Block Initialization and Availability 6-6
6.2.1 Summary 6-6
6.3 Package Exit Control Block 6-8
6.3.1 Primary Information Block - for Assembler 6-8
6.3.1.1 Field Descriptions 6-13
6.3.2 Primary Information Block - for COBOL 6-20
6.3.2.1 Field Descriptions 6-25
6.4 Package Request Control Block 6-34
6.4.1 Second Information Block 6-34
6.4.1.1 $PREQPDS DSECT 6-34
6.4.1.2 Field Descriptions 6-35
6.4.1.3 PACKAGE-REQUEST-BLOCK control block layout 6-38
6.4.1.4 Field Descriptions 6-39

6.5 Package Header Control Block 6-44
6.5.1 Third Information Block 6-44
6.5.1.1 $PHDRDS DSECT 6-44
6.5.1.2 Field Descriptions 6-45
6.5.1.3 PACKAGE-EXIT-HEADER-BLOCK control block layout . . . 6-46
6.5.1.4 Field Descriptions 6-47
6.6 Package File Control Block 6-50
6.6.1 Fourth Information Block 6-50
6.6.1.1 $PFILDS DSECT 6-50
6.6.1.2 Field Descriptions 6-50
6.6.1.3 PACKAGE-EXIT-FILE-BLOCK control block layout 6-51
6.6.1.4 Field Descriptions 6-52
6.7 Package Action Request Block 6-53
6.7.1 Fifth Information Block 6-53
6.7.1.1 $PACTREQ DSECT 6-53
6.7.1.2 Field Descriptions 6-56
6.7.1.3 PACKAGE-EXIT-ACTION-BLOCK control block layout 6-59
6.7.1.4 Field Descriptions 6-62
6.8 Package Approver Request Block 6-65
6.8.1 Sixth Information Block 6-65
6.8.1.1 $PAPPREQ DSECT 6-65
6.8.1.2 Field Descriptions 6-69
6.8.1.3 PACKAGE-EXIT-APPROVER-MAP control block layout . . . 6-70
6.8.1.4 Field Descriptions 6-71
6.9 Package Backout Request Block 6-72
6.9.1 Seventh Information Block 6-72
6.9.1.1 $PBODREQ DSECT 6-72
6.9.1.2 Field Descriptions 6-72
6.9.1.3 PACKAGE-EXIT-BACKOUT-BLOCK control block layout . . 6-73
6.9.1.4 Field Descriptions 6-74
6.10 Package Ship Request Block 6-76
6.10.1 Eighth Information Block 6-76
6.10.1.1 $PSHPREQ DSECT 6-76
6.10.1.2 Field Descriptions 6-77
6.10.1.3 PACKAGE-EXIT-SHIPMENT-BLOCK control block layout . 6-78
6.10.1.4 Field Descriptions 6-79
6.11 Package SCL Request Block 6-81
6.11.1 Ninth Information Block 6-81
6.11.1.1 $PSCLREQ DSECT 6-81
6.11.1.2 Field Descriptions 6-81
6.11.1.3 PACKAGE-EXIT-SCL-RECORD control block layout 6-82
6.11.1.4 Field Descriptions 6-83
6.11.2 Tenth Information Block 6-83
6.11.2.1 $PCOLREQ DSECT—Product Collection Request Block . . . 6-83
6.11.2.2 PACKAGE-EXIT-COLLECT-BLOCK control block layout . . 6-84
6.12 Package Exit Example 6-86
6.12.1 Streamlining the Package Approval Process 6-86

Chapter 7. Exit Program Examples 7-1
7.1 Overview 7-2
7.2 Exits 2 and 6 7-3

7.2.1 @CCIDTBL 7-3
7.2.2 Building the CCID Table for Exit 2 7-3
7.2.2.1 Rules 7-4
7.2.3 Sample Exit 2 Program 7-5
7.2.3.1 Overview 7-5
7.2.4 Sample Exit 6 Program 7-8
7.2.4.1 Overview 7-8
7.3 Notification Utility Exit Programs 7-10

Index X-1

Chapter 1. Using Exits with Endevor

Read this chapter for information about Endevor support for exit programs. Topics covered include:

- Using Exits with Endevor
- Identifying Exits to Endevor

To use exit programs with Endevor, you must:

- Identify the exits that you want to use.
- Write the exits.
- Identify the exits to Endevor by including them in the user exits table (C1UEXITS). The user exits table is created during Endevor implementation, then updated as necessary.

Note: During session initialization, Endevor loads the Defaults Table and the C1UEXITS table, then loads all exits defined in the C1UEXITS table into storage.

When it reaches an exit point, Endevor:

- Builds one or more control blocks.
- Checks the user exit table to determine the exit(s) to be called.
- Passes the control block(s) to the exit program.
- Receives the control block(s) from the exit program, displaying messages and modifying information as necessary based on the control block information.

Note: Exits written for package functions differ in some respects from other exits. For more information, see Chapter 4, “Package Exits” on page 4-1

Note: Throughout this book any references to:

- *AllFusion™: Endevor Change Manager*
- *eTrust™: CA-ACF2*
- *eTrust™: CA-Top Secret*

will simply be referred to as:

- Endevor
- CA-ACF2

- CA-Top Secret

1.1.1 Writing and Storing Exits

The Endevor exit interface is designed for use with exits written in either assembler or in high-level languages such as COBOL. Endevor supports standard IBM calling conventions for all user exits. There is one exception: exit programs must pass return codes back to Endevor through the exit control block, not through Register 15. They must also be re-entrant and re-usable.

Endevor has reserved storage subpools 10 to 19 for user exits. Although user exits can use any storage subpool, storage contained in subpools 10 to 19 can be freed at the subpool level without affecting the operation of Endevor.

1.1.2 Exit Names

Endevor provides a sample exit table with exit program names of C1UEXTnn, where nn is the number of the exit point where Endevor calls the program. For example, the default name for an exit program to be called at exit point 1 is C1UEXT01. Internal names for exits written in a high-level language (for example, COBOL), *must* maintain this naming convention (for more information, see 1.1.3, “High-Level Language Support”).

If you choose to use another naming convention for your exit programs, be sure to change the names in the sample exit table before building the table.

1.1.3 High-Level Language Support

Endevor provides an interface program, EPC1UEXT, that makes control blocks available to exit programs written in high-level languages such as COBOL. This program is provided as a load module in the uprfx.uqual.CONLIB library. Program EPC1UEXT does not support package function exits.

To use EPC1UEXT, link-edit it with the high-level language exit program, naming EPC1UEXT as the entry point. Sample linkage editor input is shown below:

```
INCLUDE SYSLIB(EPC1UEXT) *front-end program*
INCLUDE SYSLIB(exit program name) *real exit program*
ENTRY EPC1UEXT
NAME C1UEXTnn(R)
```

Note: Program EPC1UEXT recognizes only one exit program for each exit point.

1.1.3.1 Defining Multiple COBOL Programs for the Same Exit Point

To define multiple COBOL programs for the same exit point, do the following:

Step	Action
1	Compile the first exit program, for example C1UEXT03.
2	Rename the compiled program. For example, you might rename program C1UEXT03 as XIT3PROG1.
3	Compile the second program for the same exit. This program would also be named C1UEXT03.
4	Rename the second program. For example, you might rename the second C1UEXT03 program as XIT3PROG2.
5	Repeat the compile and rename steps as necessary.
6	Edit the C1UEXITS table to include the renamed exits.

1.1.3.2 Sample COBOL Linkage Section

A sample COBOL linkage section is provided as member EXITBLKS in the iprfx.iqual.SOURCE library. This member contains COBOL representations of most of the exit control blocks. It can be copied into exits that you are writing in COBOL.

1.1.4 Storing User Exits

We recommend that you store exit programs in an authorized LINKLIST library, using the exit name as the load module name.

Note: The authorization parameter (AUTH=NO) for the @C1UEXIT macro allows you to load the exit routines from either CONLIB or STEPLIB for testing purposes.

1.1.5 Debugging Exit Programs

Endevor includes a trace facility for use in debugging exit programs and for viewing the information passed to your exit programs by Endevor.

Exit trace output is contained in the User Exit Parameter Trace Log. To produce the exit trace log:

- In batch, either include the following statement in a batch job:

  ```
  //EN$TRXIT DD   SYSOUT=*
  ```

  or allocate a data set with following DCB parameters: LRECL=133, BLKSIZE=1330, RECFM=FB, DSORG=PS, then include the following statement in the job:

  ```
  //EN$TRXIT DD   DSN=data set name, DISP=SHR
  ```

- In foreground, allocate the data set as described above, then allocate this file to the foreground session data set using:

  ```
  ALLOC FI(EN$TRXIT) DA('data set name') SHR REUS
  ```

The exit trace log includes a listing of control block contents for the before- and after-exit points. This allows you to compare input and output information. If the information in a particular field in the after-exit control block is not what you expect, there is a bug in your program.

1.1.6 Sample Exit Trace Log

A sample exit trace log for a before-action (exit 2) program follows. For more information, see Chapter 2, “Exit Point Descriptions” on page 2-1 and Chapter 3, “Exit Control Blocks” on page 3-1 Blocks.

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:23:28     PAGE     1
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                        C1I0003I USER EXIT PARAMETER TRACE LOG

 11.23.28  EXIT TRACE HAS BEEN INITIALIZED.
 11.23.29  CONTROL BLOCK DFTSETLT AT 0000AB48
 11.23.29     SYS0988  0000 - C4            DFTSTGNM 0000 - D40STG1
 11.23.29     DFTSTGDS 0008 - BST.D40.STG1.MCF
 11.23.29     DFTUSECT 0034 -               DFTRSECT 003C -
 11.23.29     DFTSYSNM 0044 - Development Rel 4.0                       SMFSEC   006E - Y
 11.23.29     SMFACT   0070 - Y             SMFENV   0072 - N            DFTNXTST 0073 - 1
 11.23.29     DFTENVNM 0074 - D40           DFTNXTEN 007C - I40          DFTDBAVL 0084 - N
 11.23.29     DFTDBACT 0085 - N             DFTDBOP1 0086 - N            DFTDBOP2 0087 - N
 11.23.29     DFTDBOP3 0088 - N             DFTDBOP4 0089 - N            DFTST2NM 008A - D40STG2
 11.23.29     DFTST2DS 0092 - BST.D40.STG2.MCF
 11.23.29     DFTST11D 00BE - 1             DFTST1TI 00BF - STAGE 1
 11.23.29     DFTST21D 00D3 - 2             DFTST2TI 00D4 - STAGE 2
 11.23.29     DFTENVJN 00EC - 00000000
 11.23.29  CONTROL BLOCK DFTSETLT AT 0000AC48
 11.23.29     SYS0988  0000 - C9            DFTSTGNM 0000 - I40STG1
 11.23.29     DFTSTGDS 0008 - BST.I40.STG1.MCF
 11.23.29     DFTUSECT 0034 -               DFTRSECT 003C -
 11.23.29     DFTSYSNM 0044 - Integration Env.                          SMFSEC   006E - Y
 11.23.29     SMFACT   0070 - Y             SMFENV   0072 - N            DFTNXTST 0073 - 2
 11.23.29     DFTENVNM 0074 - I40           DFTNXTEN 007C - Q40          DFTDBAVL 0084 - N
 11.23.29     DFTDBACT 0085 - N             DFTDBOP1 0086 - N            DFTDBOP2 0087 - N
 11.23.29     DFTDBOP3 0088 - N             DFTDBOP4 0089 - N            DFTST2NM 008A - I40STG2
 11.23.29     DFTST2DS 0092 - BST.I40.STG2.MCF
 11.23.29     DFTST11D 00BE - 1             DFTST1TI 00BF - STAGE 1
 11.23.29     DFTST21D 00D3 - 2             DFTST2TI 00D4 - STAGE 2
 11.23.29     DFTENVJN 00EC - 00000000
 11.23.29  CONTROL BLOCK DFTSETLT AT 0000AD48
 11.23.29     SYS0988  0000 - D8            DFTSTGNM 0000 - Q40STG1
 11.23.29     DFTSTGDS 0008 - BST.Q40.STG1.MCF
 11.23.29     DFTUSECT 0034 -               DFTRSECT 003C -
 11.23.29     DFTSYSNM 0044 - QA Turnover Environment                   SMFSEC   006E - Y
 11.23.29     SMFACT   0070 - Y             SMFENV   0072 - N            DFTNXTST 0073 - 2
 11.23.29     DFTENVNM 0074 - Q40           DFTNXTEN 007C - P40          DFTDBAVL 0084 - N
 11.23.29     DFTDBACT 0085 - N             DFTDBOP1 0086 - N            DFTDBOP2 0087 - N
 11.23.29     DFTDBOP3 0088 - N             DFTDBOP4 0089 - N            DFTST2NM 008A - Q40STG2
 11.23.29     DFTST2DS 0092 - BST.Q40.STG2.MCF
 11.23.29     DFTST11D 00BE - 1             DFTST1TI 00BF - STAGE 1
 11.23.29     DFTST21D 00D3 - 2             DFTST2TI 00D4 - STAGE 2
 11.23.29     DFTENVJN 00EC - 00000000
 11.23.29  CONTROL BLOCK DFTSETLT AT 0000AE48
 11.23.29     SYS0988  0000 - D7            DFTSTGNM 0000 - P40STG1
 11.23.29     DFTSTGDS 0008 - BST.P40.STG1.MCF
 11.23.29     DFTUSECT 0034 -               DFTRSECT 003C -
 11.23.29     DFTSYSNM 0044 - PRODUCTION RELEASE 4.0                    SMFSEC   006E - Y
 11.23.29     SMFACT   0070 - Y             SMFENV   0072 - N            DFTNXTST 0073 -
 11.23.29     DFTENVNM 0074 - P40           DFTNXTEN 007C -              DFTDBAVL 0084 - Y
 11.23.29     DFTDBACT 0085 - Y             DFTDBOP1 0086 - N            DFTDBOP2 0087 - N
 11.23.29     DFTDBOP3 0088 - N             DFTDBOP4 0089 - N            DFTST2NM 008A - P40STG2
 11.23.29     DFTST2DS 0092 - BST.P40.STG2.MCF
 11.23.29     DFTST11D 00BE - 1             DFTST1TI 00BF - STAGE 1
 11.23.29     DFTST21D 00D3 - 2             DFTST2TI 00D4 - STAGE 2
 11.23.29     DFTENVJN 00EC - 00000000
011.23.29  EXIT MODULE BC1PNM00 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:23:29     PAGE     2
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                          C1I0003I USER EXIT PARAMETER TRACE LOG

011.23.29  BEFORE  EXIT 05 CALLED, R1(000760F0) - 000762A8
 11.23.29  CONTROL BLOCK $XITDS   AT 00075EF4
 11.23.29     XITLEN   0000 - 006C          XITVERS  0002 - 0001          XITID    0004 - $XIT
 11.23.29     XITNUMB  0008 - 00000005      XITFUNC  000C - 00000006      XITSMF   0010 - N
 11.23.29     XITINTNL 0011 - N             XITSECFL 0012 - 0080          XITMDPRM 0014 - 000749EC
 11.23.29     XITPARMB 0018 - 00075F18      XITPNLLV 001C - 00000000      XITARQP  0020 - 00000000
 11.23.29  CONTROL BLOCK $XPARMDS AT 00075F18  - ANCHORED AT XITPARMB
 11.23.29     XPALEN   0000 - 0112          XPAVERS  0002 - 0001          XPAID    0004 - $XPA
 11.23.29     XPA_ACT  0008 - DISPLAY       XPA_FUNC 0010 - DISPLAY       XPA_OPT  0018 - 00
 11.23.29     XPASRSV1 0019 - 000000        XPASACEE 001C - 00000000      XPAIENV  0038 -
 11.23.29     XPA1STG  0040 -               XPA1STNO 0048 - 00            XPA1STID 0049 -
 11.23.29     XPA1SYS  004A -               XPA1SBS  0052 -               XPA1ELM  005C - 00000000
 11.23.29     XPA1TYPE 0060 -               XPA1FLGS 0068 - M             XPA1LOC  0069 - C
 11.23.29     XPA1PTR1 006C - 00000000      XPA2ENV  0084 -               XPA2STG  008C -
 11.23.29     XPA2STNO 0094 - 00            XPA2STID 0095 -               XPA2SYS  0096 -
 11.23.29     XPA2SBS  009E -               XPA2ELM  00A8 - 00000000      XPA2TYPE 00AC -
 11.23.29     XPA2FLGS 00B4 - M             XPA2LOC  00B5 - C             XPA2PTR1 00B8 - 00000000
 11.23.29     XPA_PKGI 00D0 -                                             XPA_PKGT 00E0 -
 11.23.29     XPA_PKGS 00E8 -               XPA_PKGB 00F0 -               XPA_PKGS 00F1 -
 11.23.29     XPA_PKGA 00F2 -                                             XPA_PKGS 0102 -
 11.23.29     XPA_PKGD 010A -
 11.23.29  CONTROL BLOCK $ECBDS   AT 000762A8
 11.23.29     ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
 11.23.29     ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000001
 11.23.29     ECBEXTN  0014 - 00000005      ECBUSER  0018 - OLEJU01       ECBMODE  0020 - B
 11.23.29     ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - N
 11.23.29     ECBFUNC  0024 - 00000006      ECBFUNAM 0028 - DISPLAY       ECBMSGCD 0030 - 0000
 11.23.29     ECBMSGLN 0034 - 0000
 11.23.29     ECBMSG   0036 -
 11.23.29     ECBCALR  00BA - E             ECBPKG   00BB - N             ECBREQA  00BC - 00076378
 11.23.29     ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000      ECBIFUNC 00C8 - 00000000
 11.23.29     ECBHIRC  00CC - 00000000
 11.23.29  CONTROL BLOCK DFTSETLT AT 0000AB48
 11.23.29     SYS0988  0000 - C4            DFTSTGNM 0000 - D40STG1
 11.23.29     DFTSTGDS 0008 - BST.D40.STG1.MCF
 11.23.29     DFTUSECT 0034 -               DFTRSECT 003C -
 11.23.29     DFTSYSNM 0044 - Development Rel 4.0                         SMFSEC   006E - Y
 11.23.29     SMFACT   0070 - Y             SMFENV   0072 - N             DFTNXTST 0073 - 1
 11.23.29     DFTENVNM 0074 - D40           DFTNXTEN 007C - I40           DFTDBAVL 0084 - N
 11.23.29     DFTDBACT 0085 - N             DFTDBOP1 0086 - N             DFTDBOP2 0087 - N
 11.23.29     DFTDBOP3 0088 - N             DFTDBOP4 0089 - N             DFTST2NM 008A - D40STG2
 11.23.29     DFTST2DS 0092 - BST.D40.STG2.MCF
 11.23.29     DFTST1ID 00BE - 1             DFTST1TI 00BF - STAGE 1
 11.23.29     DFTST2ID 00D3 - 2             DFTST2TI 00D4 - STAGE 2
 11.23.29     DFTENVJN 00EC - 00000000
 11.23.29  CONTROL BLOCK DFTSETLT AT 0000AC48
 11.23.29     SYS0988  0000 - C9            DFTSTGNM 0000 - I40STG1
 11.23.29     DFTSTGDS 0008 - BST.I40.STG1.MCF
 11.23.29     DFTUSECT 0034 -               DFTRSECT 003C -
 11.23.29     DFTSYSNM 0044 - Integration Env.                            SMFSEC   006E - Y
 11.23.29     SMFACT   0070 - Y             SMFENV   0072 - N             DFTNXTST 0073 - 2
 11.23.29     DFTENVNM 0074 - I40           DFTNXTEN 007C - Q40           DFTDBAVL 0084 - N
 11.23.29     DFTDBACT 0085 - N             DFTDBOP1 0086 - N             DFTDBOP2 0087 - N
 11.23.29     DFTDBOP3 0088 - N             DFTDBOP4 0089 - N             DFTST2NM 008A - I40STG2
 11.23.29     DFTST2DS 0092 - BST.I40.STG2.MCF

1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:23:29     PAGE     3
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                          C1I0003I USER EXIT PARAMETER TRACE LOG

 11.23.29     DFTST1ID 00BE - 1             DFTST1TI 00BF - STAGE 1
 11.23.29     DFTST2ID 00D3 - 2             DFTST2TI 00D4 - STAGE 2
 11.23.29     DFTENVJN 00EC - 00000000
 11.23.29  CONTROL BLOCK DFTSETLT AT 0000AD48
 11.23.29     SYS0988  0000 - D8            DFTSTGNM 0000 - Q40STG1
 11.23.29     DFTSTGDS 0008 - BST.Q40.STG1.MCF
 11.23.29     DFTUSECT 0034 -               DFTRSECT 003C -
 11.23.29     DFTSYSNM 0044 - QA Turnover Environment                     SMFSEC   006E - Y
 11.23.29     SMFACT   0070 - Y             SMFENV   0072 - N             DFTNXTST 0073 - 2
 11.23.29     DFTENVNM 0074 - Q40           DFTNXTEN 007C - P40           DFTDBAVL 0084 - N
 11.23.29     DFTDBACT 0085 - N             DFTDBOP1 0086 - N             DFTDBOP2 0087 - N
 11.23.29     DFTDBOP3 0088 - N             DFTDBOP4 0089 - N             DFTST2NM 008A - Q40STG2
 11.23.29     DFTST2DS 0092 - BST.Q40.STG2.MCF
 11.23.29     DFTST1ID 00BE - 1             DFTST1TI 00BF - STAGE 1
 11.23.29     DFTST2ID 00D3 - 2             DFTST2TI 00D4 - STAGE 2
 11.23.29     DFTENVJN 00EC - 00000000
 11.23.29  CONTROL BLOCK DFTSETLT AT 0000AE48
 11.23.29     SYS0988  0000 - D7            DFTSTGNM 0000 - P40STG1
 11.23.29     DFTSTGDS 0008 - BST.P40.STG1.MCF
 11.23.29     DFTUSECT 0034 -               DFTRSECT 003C -
 11.23.29     DFTSYSNM 0044 - PRODUCTION RELEASE 4.0                      SMFSEC   006E - Y
 11.23.29     SMFACT   0070 - Y             SMFENV   0072 - N             DFTNXTST 0073 -
 11.23.29     DFTENVNM 0074 - P40           DFTNXTEN 007C -               DFTDBAVL 0084 - Y
 11.23.29     DFTDBACT 0085 - Y             DFTDBOP1 0086 - N             DFTDBOP2 0087 - N
 11.23.29     DFTDBOP3 0088 - N             DFTDBOP4 0089 - N             DFTST2NM 008A - P40STG2
 11.23.29     DFTST2DS 0092 - BST.P40.STG2.MCF
 11.23.29     DFTST1ID 00BE - 1             DFTST1TI 00BF - STAGE 1
 11.23.29     DFTST2ID 00D3 - 2             DFTST2TI 00D4 - STAGE 2
 11.23.29     DFTENVJN 00EC - 00000000
011.23.29  AFTER   EXIT 05 CALLED, R1(000760F0) - 000762A8
 11.23.29  CONTROL BLOCK $ECBDS   AT 000762A8
 11.23.29     ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
 11.23.29     ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000001
 11.23.29     ECBEXTN  0014 - 00000005      ECBUSER  0018 - OLEJU01       ECBMODE  0020 - B
 11.23.29     ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - N
 11.23.29     ECBFUNC  0024 - 00000006      ECBFUNAM 0028 - DISPLAY       ECBMSGCD 0030 - 0000
 11.23.29     ECBMSGLN 0034 - 0000
 11.23.29     ECBMSG   0036 -
 11.23.29     ECBCALR  00BA - E             ECBPKG   00BB - N             ECBREQA  00BC - 00076378
 11.23.29     ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000      ECBIFUNC 00C8 - 00000000
 11.23.29     ECBHIRC  00CC - 00000000
011.24.24  EXIT MODULE BC1PNM00 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
011.24.24  BEFORE  EXIT 02 CALLED, R1(000790A8) - 00079260 0007AD00
 11.24.24  CONTROL BLOCK $XITDS   AT 00077CDC
 11.24.24     XITLEN   0000 - 006C          XITVERS  0002 - 0001          XITID    0004 - $XIT
 11.24.24     XITNUMB  0008 - 00000002      XITFUNC  000C - 00000003      XITSMF   0010 - N
 11.24.24     XITINTNL 0011 - N             XITSECFL 0012 - 4000          XITMDPRM 0014 - 000749EC
 11.24.24     XITPARMB 0018 - 0007848C      XITPNLLV 001C - 00000000      XITARQP  0020 - 19725158
 11.24.24  CONTROL BLOCK $XPARMDS AT 0007848C  - ANCHORED AT XITPARMB
 11.24.24     XPALEN   0000 - 0112          XPAVERS  0002 - 0001          XPAID    0004 - $XPA
 11.24.24     XPA_ACT  0008 - RETRIEVE      XPA_FUNC 0010 - RETRIEVE      XPA_OPT  0018 - 00
 11.24.24     XPASRSV1 0019 - 000000        XPASACEE 001C - 00000000      XPAIENV  0038 - I40
 11.24.24     XPA1STG  0040 - I40STG1       XPA1STNO 0048 - 01            XPA1STID 0049 - 1
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:24:24     PAGE     4
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                                C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.24     XPA1SYS  004A - NDVRMVS       XPA1SBS  0052 - JUDY          XPA1ELM  005C - 197D71DE
 11.24.24     XPA1TYPE 0060 - ASMMAC        XPA1FLGS 0068 - M             XPA1LOC  0069 - C
 11.24.24     XPA1PTR1 006C - 1970A804      XPA2ENV  0084 -               XPA2STG  008C -
 11.24.24     XPA2STNO 0094 - 00            XPA2STID 0095 -               XPA2SYS  0096 -
 11.24.24     XPA2SBS  009E -               XPA2ELM  00A8 - 00000000      XPA2TYPE 00AC -
 11.24.24     XPA2FLGS 00B4 - I             XPA2LOC  00B5 - C             XPA2PTR1 00B8 - 00091410
 11.24.24     XPA_PKGI 00D0 -                                             XPA_PKGT 00E0 -
 11.24.24     XPA_PKGS 00E8 -               XPA_PKGB 00F0 -               XPA_PKGS 00F1 -
 11.24.24     XPA_PKGA 00F2 -                                             XPA_PKGS 0102 -
 11.24.24     XPA_PKGD 010A -
 11.24.24  CONTROL BLOCK $ECBDS   AT 00079260
 11.24.24     ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
 11.24.24     ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000002
 11.24.24     ECBEXTN  0014 - 00000002      ECBUSER  0018 - OLEJU01       ECBMODE  0020 - B
 11.24.24     ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - N
 11.24.24     ECBFUNC  0024 - 00000003      ECBFUNAM 0028 - RETRIEVE      ECBMSGCD 0030 -
 11.24.24     ECBMSGLN 0034 - 0000
 11.24.24     ECBMSG   0036 -
 11.24.24     ECBCALR  00BA - E             ECBPKG   00BB - N             ECBREQA  00BC - 00079330
 11.24.24     ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000      ECBIFUNC 00C8 - 00000000
 11.24.24     ECBHIRC  00CC - 00000000
 11.24.24  CONTROL BLOCK $REQPDS  AT 00079330  - ANCHORED AT ECBREQA
 11.24.24     REQLEN   0000 - 006C          REQVERS  0002 - 0002          REQID    0004 - $REQ
 11.24.24     REQCCID  0008 - RELTST40      REQCOMM  0014 - TEST EXITS - TRACE
 11.24.24     REQFLAG  003C - 00000000      REQSISOF 0040 - N             REQSISOC 0041 - N
 11.24.24     REQEXPND 0042 - N             REQOVRWR 0043 - N             REQRTCOD 0044 - 00000000
 11.24.24     REQELVER 0048 -               REQCPYBK 004A -               REQDEL   0050 -
 11.24.24     REQMOVWH 0051 -               REQUPDT  0052 -               REQBYGEN 0053 -
 11.24.24     REQONLCM 0054 -               REQSYNC  0055 -               REQIGGF  0056 -
 11.24.24     REQBYDP  0057 -               REQOPGRP 0058 -               REQSOUID 0060 -
 11.24.24     REQOSRCH 0068 - Y             REQORSGN 0069 -               REQOJUMP 006A -
 11.24.24     REQOSGNI 006B -
 11.24.24  CONTROL BLOCK $ENVDS   AT 0007AD00
 11.24.24     ENVLEN   0000 - 01C9          ENVVERS  0002 - 0003          ENVID    0004 - $ENV
 11.24.24     ENVFLAG  0008 - 0000          ENVSPEC  000A - C             ENVIOTY  000B - I
 11.24.24     ENVNEXT  000C - 0007AED0      ENVFCBA  0010 - 000793E0      ENVSITE  0014 - 0
 11.24.24     ENVSTGID 0015 - 1             ENVSTGCD 0016 - 1             ENVDATAC 0017 -
 11.24.24     ENVEVER  0018 - 0001          ENVELVL  001A - 0000          ENVENVM  001C - I40
 11.24.24     ENVSTAGE 0024 - I40STG1       ENVSYSTM 002C - NDVRMVS       ENVSYSUD 0034 -
 11.24.24     ENVSUBSY 0044 - JUDY          ENVSUBUD 004C -               ENVTYPE  0054 - ASMMAC
 11.24.24     ENVTYPUD 005C -               ENVELEMT 0064 - $ENMANCS      ENVEAOFF 00C4 - 00C8
 11.24.24     ENVBFARE 00C8 - 00085BC5D5D4C1D5C3E2000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.24  CONTROL BLOCK $ELMDS   AT 000793E0  - ANCHORED AT ENVFCBA
 11.24.24     ELMLEN   0000 - 0C8C          ELMVERS  0002 - 0005          ELMID    0004 - $ELM
 11.24.24     ELMFLAGS 0008 - 00000000      ELMMVERS 000C - 0001          ELMMLLVL 000E - 0000
 11.24.24     ELMMINS  0010 - 00000000      ELMMDEL  0014 - 00000000      ELMNAME  0018 - $ENMANCS
 11.24.24     ELMCCID  0022 - REL4.0TST     ELMM1STL 002E - 0000          ELMMBTOT 0030 - 00000077
 11.24.24     ELMMBDTE 0034 - 021106        ELMMBTIM 003A - 0857
 11.24.24     ELMBCOM  003E - test type sequence                          ELMBUSID 0066 - OLEJU01
 11.24.24     ELMPUSID 006E - OLEJU01       ELMPDSTP 0076 - 021106        ELMPTSTP 007C - 0857
 11.24.24     ELMFDSN  0080 - BST.OLEJU40.MACLIB
 11.24.24     ELMFMBR  00AC - 0ENMANCS      ELMPRFLG 00B6 - 40            ELMLPROD 00BA - 021106
 11.24.24     ELMLPROT 00C0 - 0857          ELMLPROU 00C4 - OLEJU01       ELMLPRON 00CC - *NOPROC*
 11.24.24     ELMMPRC  00D6 - FFFF          ELMMRC   00D8 - 0000
 11.24.24     ELMPCOM  00DA - test type sequence
 11.24.24     ELMCCOM  0102 - test type sequence                          ELMMLDTE 012A - 021106
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:24:24     PAGE     5
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                        C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.24     ELMMLTME 0130 - 0857          ELMLUSID 0134 - OLEJU01       ELMMLACT 013C - ADD
 11.24.24     ELMMLTOT 0144 - 00000077      ELMMVDTE 0148 -               ELMMVTIM 014E -
 11.24.24     ELMMUSID 0152 -               ELMRDSTP 015A -               ELMRTSTP 0160 -
 11.24.24     ELMRIID  0164 -               ELMRCOM  016C -
 11.24.24     ELMTDSN  0194 -
 11.24.24     ELMTMBR  01C0 -               ELMRCCID 01E8 -               ELMGCCID 01F4 - REL4.0TST
 11.24.24     ELMODACT 0200 - ADD           ELMLCCID 0208 - REL4.0TST
 11.24.24     ELMLCOMM 0214 - test type sequence                          ELMLUID  023C - OLEJU01
 11.24.24     ELMLADT  0244 - 021106        ELMLATM  024A - 0857          ELMSOID  024E - OLEJU01
 11.24.24     ELFRFLAG 0256 - 00            ELFRACT  0257 -               ELFRUSID 025F -
 11.24.24     ELFRDATE 0267 -               ELFRTIME 026D -               ELFRSITE 0271 -
 11.24.24     ELFRENV  0272 -               ELFRSYS  027A -               ELFRSBS  0282 -
 11.24.24     ELFRELM  028A -               ELFRTYPE 0294 -               ELFRSTG# 029C -
 11.24.24     ELFRVER  029E - 0000          ELFRLVL  02A0 - 0000          ELMUPDT  02A2 - ER2HZTA0
 11.24.24     ELMBASE  02AC - $ENMANCS      ELMLOWL  02B6 - 00            ELMIPCTG 02B8 - 00
 11.24.24     ELMDPCTG 02B9 - 00            ELMFPESD 02BA -               ELMEDFMT 02C2 - R
 11.24.24     ELMENPAK 02C3 - Y             ELMLPROV 02C4 - 00            ELMPROL  02C5 - 00
 11.24.24     ELMFMID  02C6 - 29            ELMXDNAM 02C7 -               ELMXBVER 02D0 - 0000
 11.24.24     ELMXBTOT 02D4 - 00000000      ELMXLTOT 02D8 - 00000000      ELMXBLVL 02DC - 0000
 11.24.24     ELMXLLVL 02DE - 0000          ELMXINS  02E0 - 0000          ELMXDLS  02E2 - 0000
 11.24.24     ELMXRGIN 02E4 - 0000          ELMXRGDL 02E6 - 0000          ELMXBDTE 02E8 -
 11.24.24     ELMXBTIM 02EE -               ELMXLDTE 02F2 -               ELMXLTIM 02F8 -
 11.24.24     ELMXFLAG 02FC - 00            ELMXDFMT 02FD -               ELMXUPDT 02FE -
 11.24.24     ELMSPKG  02FF -                                             ELMSPKGT 030F - 0000000000000000
 11.24.24     ELMOPKG  0317 -                                             ELMOPKGT 0327 - 0000000000000000
 11.24.24     ELMPRGRP 032F - *NOPROC*      ELMCPKG  0337 -
 11.24.24     ELMCPKGT 0347 - 0000000000000000                            ELMFFFLG 034F - D
 11.24.24     ELMTFFLG 0350 -               ELMFPAOF 0352 - 0000          ELMFNAOF 0354 - 0000
 11.24.24     ELMTPAOF 0356 - 0000          ELMTNAOF 0358 - 0000          ELMEAOFF 035A - 037C
 11.24.24     ELMBFARE 037C - 00085BC5D5D4C1D5C3E2000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.24  CONTROL BLOCK $ENVDS   AT 0007AED0  - ANCHORED AT ENVNEXT
 11.24.24     ENVLEN   0000 - 01C9          ENVVERS  0002 - 0003          ENVID    0004 - $ENV
 11.24.24     ENVFLAG  0008 - 0000          ENVSPEC  000A - E             ENVIOTY  000B - O
 11.24.24     ENVNEXT  000C - 00000000      ENVFCBA  0010 - 0007B508      ENVSITE  0014 - 0
 11.24.24     ENVSTGID 0015 - 0             ENVSTGCD 0016 -               ENVDATAC 0017 -
 11.24.24     ENVEVER  0018 - 0000          ENVELVL  001A - 0000          ENVENVM  001C -
 11.24.24     ENVSTAGE 0024 -               ENVSYSTM 002C -               ENVSYSUD 0034 -
 11.24.24     ENVSUBSY 0044 -               ENVSUBUD 004C -               ENVTYPE  0054 -
 11.24.24     ENVTYPUD 005C -               ENVELEMT 0064 -               ENVEAOFF 00C4 - 0000
 11.24.24     ENVBFARE 00C8 - 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.24  CONTROL BLOCK $FILDS   AT 0007B508  - ANCHORED AT ENVFCBA
 11.24.24     FILLEN   0000 - 0466          FILVERS  0002 - 0002          FILID    0004 - $FIL
 11.24.24     FILDSN   0008 - BST.OLEJU40.MACLIB
 11.24.24     FILDSMEM 0034 - $ENMANCS      FILDDN   003E - SYS00007       FILDSTY  0046 - PO
 11.24.24     FILPAOFF 004A - 0000          FILNAOFF 004C - 0000
 11.24.24     FILBFARE 0063 - 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
011.24.24  AFTER   EXIT 02 CALLED, R1(000790A8) - 00079260 0007AD00
 11.24.24  CONTROL BLOCK $ECBDS   AT 00079260
 11.24.24     ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
 11.24.24     ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000002
 11.24.24     ECBEXTN  0014 - 00000002      ECBUSER  0018 - OLEJU01       ECBMODE  0020 - B
 11.24.24     ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - N
 11.24.24     ECBFUNC  0024 - 00000003      ECBFUNAM 002B - RETRIEVE      ECBMSGCD 0030 -
 11.24.24     ECBMSGLN 0034 - 0000
 11.24.24     ECBMSG   0036 -
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:24:24     PAGE     6
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                                C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.24     ECBCALR  00BA - E             ECBPKG   00BB - N             ECBREQA  00BC - 00079330
 11.24.24     ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000      ECBIFUNC 00C8 - 00000000
 11.24.24     ECBHIRC  00CC - 00000000
 11.24.24  CONTROL BLOCK $REQPDS  AT 00079330  - ANCHORED AT ECBREQA
 11.24.24     REQLEN   0000 - 006C          REQVERS  0002 - 0002          REQID    0004 - $REQ
 11.24.24     REQCCID  0008 - RELTST40      REQCOMM  0014 - TEST EXITS - TRACE
 11.24.24     REQFLAG  003C - 00000000      REQSISOF 0040 - N             REQSISOC 0041 - N
 11.24.24     REQEXPND 0042 - N             REQOVRWR 0043 - N             REQRTCOD 0044 - 00000000
 11.24.24     REQELVER 0048 -               REQCPYBK 004A -               REQDEL   0050 -
 11.24.24     REQMOVWH 0051 -               REQUPDT  0052 -               REQBYGEN 0053 -
 11.24.24     REQONLCM 0054 -               REQSYNC  0055 -               REQIGGF  0056 -
 11.24.24     REQBYDP  0057 -               REQOPGRP 0058 -               REQSOUID 0060 -
 11.24.24     REQOSRCH 0068 - Y             REQORSGN 0069 -               REQOJUMP 006A -
 11.24.24     REQOSGNI 006B -
 11.24.24  CONTROL BLOCK $ENVDS   AT 0007AD00
 11.24.24     ENVLEN   0000 - 01C9          ENVVERS  0002 - 0003          ENVID    0004 - $ENV
 11.24.24     ENVFLAG  0008 - 0000          ENVSPEC  000A - C             ENVIOTY  000B - I
 11.24.24     ENVNEXT  000C - 0007AED0      ENVFCBA  0010 - 000793E0      ENVSITE  0014 - 0
 11.24.24     ENVSTGID 0015 - 1             ENVSTGCD 0016 - 1             ENVDATAC 0017 -
 11.24.24     ENVEVER  0018 - 0001          ENVELVL  001A - 0000          ENVENVM  001C - I40
 11.24.24     ENVSTAGE 0024 - I40STG1       ENVSYSTM 002C - NDVRMVS       ENVSYSUD 0034 -
 11.24.24     ENVSUBSY 0044 - JUDY          ENVSUBUD 004C -               ENVTYPE  0054 - ASMMAC
 11.24.24     ENVTYPUD 005C -               ENVELEMT 0064 - $ENMANCS      ENVEAOFF 00C4 - 00C8
 11.24.24     ENVBFARE 00C8 - 00085BC5D5D4C1D5C3E2000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.24  CONTROL BLOCK $ELMDS   AT 000793E0  - ANCHORED AT ENVFCBA
 11.24.24     ELMLEN   0000 - 0C8C          ELMVERS  0002 - 0005          ELMID    0004 - $ELM
 11.24.24     ELMFLAGS 0008 - 00000000      ELMMVERS 000C - 0001          ELMMLLVL 000E - 0000
 11.24.24     ELMMINS  0010 - 00000000      ELMMDEL  0014 - 00000000      ELMNAME  0018 - $ENMANCS
 11.24.24     ELMCCID  0022 - REL4.0TST     ELMM1STL 002E - 0000          ELMMBTOT 0030 - 00000077
 11.24.24     ELMMBDTE 0034 - 021106        ELMMBTIM 003A - 0857
 11.24.24     ELMBCOM  003E - test type sequence                          ELMBUSID 0066 - OLEJU01
 11.24.24     ELMPUSID 006E - OLEJU01       ELMPDSTP 0076 - 021106        ELMPTSTP 007C - 0857
 11.24.24     ELMFDSN  0080 - BST.OLEJU40.MACLIB
 11.24.24     ELMFMBR  00AC - 0ENMANCS      ELMPRFLG 00B6 - 40            ELMLPROD 00BA - 021106
 11.24.24     ELMLPROT 00C0 - 0857          ELMLPROU 00C4 - OLEJU01       ELMLPRON 00CC - *NOPROC*
 11.24.24     ELMMPRC  00D6 - FFFF          ELMMRC   00D8 - 0000
 11.24.24     ELMPCOM  00DA - test type sequence
 11.24.24     ELMCCOM  0102 - test type sequence                          ELMMLDTE 012A - 021106
 11.24.24     ELMMLTME 0130 - 0857          ELMLUSID 0134 - OLEJU01       ELMMLACT 013C - ADD
 11.24.24     ELMMLTOT 0144 - 00000077      ELMMVDTE 0148 -               ELMMVTIM 014E -
 11.24.24     ELMMUSID 0152 -               ELMRDSTP 015A -               ELMRTSTP 0160 -
 11.24.24     ELMRIID  0164 -               ELMRCOM  016C -
 11.24.24     ELMTDSN  0194 -
 11.24.24     ELMTMBR  01C0 -               ELMRCCID 01E8 -               ELMGCCID 01F4 - REL4.0TST
 11.24.24     ELMODACT 0200 - ADD           ELMLCCID 0208 - REL4.0TST
 11.24.24     ELMLCOMM 0214 - test type sequence                          ELMLUID  023C - OLEJU01
 11.24.24     ELMLADT  0244 - 021106        ELMLATM  024A - 0857          ELMSOID  024E - OLEJU01
 11.24.24     ELFRFLAG 0256 - 00            ELFRACT  0257 -               ELFRUSID 025F -
 11.24.24     ELFRDATE 0267 -               ELFRTIME 026D -               ELFRSITE 0271 -
 11.24.24     ELFRENV  0272 -               ELFRSYS  027A -               ELFRSBS  0282 -
 11.24.24     ELFRELM  028A -               ELFRTYPE 0294 -               ELFRSTG# 029C -
 11.24.24     ELFRVER  029E - 0000          ELFRLVL  02A0 - 0000          ELMUPDT  02A2 - ER2HZTA0
 11.24.24     ELMBASE  02AC - $ENMANCS      ELMLOWL  02B6 - 00            ELMIPCTG 02B8 - 00
 11.24.24     ELMDPCTG 02B9 - 00            ELMFPESD 02BA -               ELMEDFMT 02C2 - R
 11.24.24     ELMENPAK 02C3 - Y             ELMLPROV 02C4 - 00            ELMPROL  02C5 - 00
 11.24.24     ELMFMID  02C6 - 29            ELMXDNAM 02C7 -               ELMXBVER 02D0 - 0000
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:24:24     PAGE     7
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                          C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.24     ELMXBTOT 02D4 - 00000000      ELMXLTOT 02D8 - 00000000      ELMXBLVL 02DC - 0000
 11.24.24     ELMXLLVL 02DE - 0000          ELMXINS  02E0 - 0000          ELMXDLS  02E2 - 0000
 11.24.24     ELMXRGIN 02E4 - 0000          ELMXRGDL 02E6 - 0000          ELMXBDTE 02E8 -
 11.24.24     ELMXBTIM 02EE -               ELMXLDTE 02F2 -               ELMXLTIM 02F8 -
 11.24.24     ELMXFLAG 02FC - 00            ELMXDFMT 02FD -               ELMXUPDT 02FE -
 11.24.24     ELMSPKG  02FF -                                             ELMSPKGT 030F - 0000000000000000
 11.24.24     ELMOPKG  0317 -                                             ELMOPKGT 0327 - 0000000000000000
 11.24.24     ELMPRGRP 032F - *NOPROC*      ELMCPKG  0337 -
 11.24.24     ELMCPKGT 0347 - 0000000000000000                            ELMFFFLG 034F - D
 11.24.24     ELMTFFLG 0350 -               ELMFPAOF 0352 - 0000          ELMFNAOF 0354 - 0000
 11.24.24     ELMTPAOF 0356 - 0000          ELMTNAOF 0358 - 0000          ELMEAOFF 035A - 037C
 11.24.24     ELMBFARE 037C - 00085BC5D5D4C1D5C3E2000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.24  CONTROL BLOCK $ENVDS   AT 0007AED0  - ANCHORED AT ENVNEXT
 11.24.24     ENVLEN   0000 - 01C9          ENVVERS  0002 - 0003          ENVID    0004 - $ENV
 11.24.24     ENVFLAG  0008 - 0000          ENVSPEC  000A - E             ENVIOTY  000B - O
 11.24.24     ENVNEXT  000C - 00000000      ENVFCBA  0010 - 0007B508      ENVSITE  0014 - 0
 11.24.24     ENVSTGID 0015 - 0             ENVSTGCD 0016 -               ENVDATAC 0017 -
 11.24.24     ENVEVER  0018 - 0000          ENVELVL  001A - 0000          ENVENVM  001C -
 11.24.24     ENVSTAGE 0024 -               ENVSYSTM 002C -               ENVSYSUD 0034 -
 11.24.24     ENVSUBSY 0044 -               ENVSUBUD 004C -               ENVTYPE  0054 -
 11.24.24     ENVTYPUD 005C -               ENVELEMT 0064 -               ENVEAOFF 00C4 - 0000
 11.24.24     ENVBFARE 00C8 - 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.24  CONTROL BLOCK $FILDS   AT 0007B508  - ANCHORED AT ENVFCBA
 11.24.24     FILLEN   0000 - 0466          FILVERS  0002 - 0002          FILID    0004 - $FIL
 11.24.24     FILDSN   0008 - BST.OLEJU40.MACLIB
 11.24.24     FILDSMEM 0034 - $ENMANCS      FILDDN   003E - SYS00007       FILDSTY  0046 - PO
 11.24.24     FILPAOFF 004A - 0000          FILNAOFF 004C - 0000
 11.24.24     FILBFARE 0063 - 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
011.24.38  EXIT MODULE BC1PNM00 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
011.24.38  BEFORE  EXIT 03 CALLED, R1(00078D10) - 00078EC8 0007A968 0007B5D8
 11.24.38  CONTROL BLOCK $XITDS   AT 00077CDC
 11.24.38     XITLEN   0000 - 006C          XITVERS  0002 - 0001          XITID    0004 - $XIT
 11.24.38     XITNUMB  0008 - 00000003      XITFUNC  000C - 00000003      XITSMF   0010 - N
 11.24.38     XITINTNL 0011 - N             XITSECFL 0012 - 4000          XITMDPRM 0014 - 000749EC
 11.24.38     XITPARMB 0018 - 0007848C      XITPNLLV 001C - 00000000      XITARQP  0020 - 19725158
 11.24.38  CONTROL BLOCK $XPARMDS AT 0007848C  - ANCHORED AT XITPARMB
 11.24.38     XPALEN   0000 - 0112          XPAVERS  0002 - 0001          XPAID    0004 - $XPA
 11.24.38     XPA_ACT  0008 - RETRIEVE      XPA_FUNC 0010 - RETRIEVE      XPA_OPT  0018 - 00
 11.24.38     XPASRSV1 0019 - 000000        XPASACEE 001C - 00000000      XPAIENV  0038 - I40
 11.24.38     XPA1STG  0040 - I40STG1       XPA1STNO 0048 - 01            XPA1STID 0049 - 1
 11.24.38     XPA1SYS  004A - NDVRMVS       XPA1SBS  0052 - JUDY          XPA1ELM  005C - 197D71DE
 11.24.38     XPA1TYPE 0060 - ASMMAC        XPA1FLGS 0068 - M             XPA1LOC  0069 - C
 11.24.38     XPA1PTR1 006C - 1970A804      XPA2ENV  0084 -               XPA2STG  008C -
 11.24.38     XPA2STNO 0094 - 00            XPA2STID 0095 -               XPA2SYS  0096 -
 11.24.38     XPA2SBS  009E -               XPA2ELM  00A8 - 00000000      XPA2TYPE 00AC -
 11.24.38     XPA2FLGS 00B4 - I             XPA2LOC  00B5 - C             XPA2PTR1 00B8 - 00091410
 11.24.38     XPA_PKGI 00D0 -                                             XPA_PKGT 00E0 -
 11.24.38     XPA_PKGS 00E8 -               XPA_PKGB 00F0 -               XPA_PKGS 00F1 -
 11.24.38     XPA_PKGA 00F2 -                                             XPA_PKGS 0102 -
 11.24.38     XPA_PKGD 010A -
 11.24.38  CONTROL BLOCK $ECBDS   AT 00078EC8
 11.24.38     ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
 11.24.38     ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000003
 11.24.38     ECBEXTN  0014 - 00000003      ECBUSER  0018 - OLEJU01       ECBMODE  0020 - B
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:24:38     PAGE     8
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                                C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.38     ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - Y
 11.24.38     ECBFUNC  0024 - 00000003      ECBFUNAM 0028 - RETRIEVE      ECBMSGCD 0030 - 0000
 11.24.38     ECBMSGLN 0034 - 0000
 11.24.38     ECBMSG   0036 -
 11.24.38     ECBCALR  00BA - E             ECBPKG   00BB - N             ECBREQA  00BC - 00078F98
 11.24.38     ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000      ECBIFUNC 00C8 - 00000000
 11.24.38     ECBHIRC  00CC - 00000000
 11.24.38  CONTROL BLOCK $REQPDS  AT 00078F98  - ANCHORED AT ECBREQA
 11.24.38     REQLEN   0000 - 006C          REQVERS  0002 - 0002          REQID    0004 - $REQ
 11.24.38     REQCCID  0008 - RELTST40      REQCOMM  0014 - TEST EXITS - TRACE
 11.24.38     REQFLAG  003C - 00000000      REQSISOF 0040 - N             REQSISOC 0041 - N
 11.24.38     REQEXPND 0042 - N             REQOVRWR 0043 - N             REQRTCOD 0044 - 00000000
 11.24.38     REQELVER 0048 -               REQCPYBK 004A -               REQDEL   0050 -
 11.24.38     REQMOVWH 0051 -               REQUPDT  0052 -               REQBYGEN 0053 -
 11.24.38     REQONLCM 0054 -               REQSYNC  0055 -               REQIGGF  0056 -
 11.24.38     REQBYDP  0057 -               REQOPGRP 0058 -               REQSOUID 0060 -
 11.24.38     REQOSRCH 0068 - Y             REQORSGN 0069 -               REQOJUMP 006A -
 11.24.38     REQOSGNI 006B -
 11.24.38  CONTROL BLOCK $ENVDS   AT 0007A968
 11.24.38     ENVLEN   0000 - 01C9          ENVVERS  0002 - 0003          ENVID    0004 - $ENV
 11.24.38     ENVFLAG  0008 - 0000          ENVSPEC  000A - C             ENVIOTY  000B - I
 11.24.38     ENVNEXT  000C - 0007AB38      ENVFCBA  0010 - 00079048      ENVSITE  0014 - 0
 11.24.38     ENVSTGID 0015 - 1             ENVSTGCD 0016 - 1             ENVDATAC 0017 -
 11.24.38     ENVEVER  0018 - 0001          ENVELVL  001A - 0000          ENVENVM  001C - I40
 11.24.38     ENVSTAGE 0024 - I40STG1       ENVSYSTM 002C - NDVRMVS       ENVSYSUD 0034 -
 11.24.38     ENVSUBSY 0044 - JUDY          ENVSUBUD 004C -               ENVTYPE  0054 - ASMMAC
 11.24.38     ENVTYPUD 005C -               ENVELEMT 0064 - $ENMANCS      ENVEAOFF 00C4 - 00C8
 11.24.38     ENVBFARE 00C8 - 00085BC5D5D4C1D5C3E2000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK $ELMDS   AT 00079048  - ANCHORED AT ENVFCBA
 11.24.38     ELMLEN   0000 - 0C8C          ELMVERS  0002 - 0005          ELMID    0004 - $ELM
 11.24.38     ELMFLAGS 0008 - 00000000      ELMMVERS 000C - 0001          ELMMLLVL 000E - 0000
 11.24.38     ELMMINS  0010 - 00000000      ELMMDEL  0014 - 00000000      ELMNAME  0018 - $ENMANCS
 11.24.38     ELMCCID  0022 - REL4.0TST     ELMM1STL 002E - 0000          ELMMBTOT 0030 - 00000077
 11.24.38     ELMMBDTE 0034 - 021106        ELMMBTIM 003A - 0857
 11.24.38     ELMBCOM  003E - test type sequence                          ELMBUSID 0066 - OLEJU01
 11.24.38     ELMPUSID 006E - OLEJU01       ELMPDSTP 0076 - 021106        ELMPTSTP 007C - 0857
 11.24.38     ELMFDSN  0080 - BST.OLEJU40.MACLIB
 11.24.38     ELMFMBR  00AC - @ENMANCS      ELMPRFLG 00B6 - 40            ELMLPROD 00BA - 021106
 11.24.38     ELMLPROT 00C0 - 0857          ELMLPROU 00C4 - OLEJU01       ELMLPRON 00CC - *NOPROC*
 11.24.38     ELMMPRC  00D6 - FFFF          ELMMRC   00D8 - 0000
 11.24.38     ELMPCOM  00DA - test type sequence
 11.24.38     ELMCCOM  0102 - test type sequence                          ELMMLDTE 012A - 021106
 11.24.38     ELMMLTME 0130 - 0857          ELMLUSID 0134 - OLEJU01       ELMMLACT 013C - RETRIEVE
 11.24.38     ELMMLTOT 0144 - 00000077      ELMMVDTE 0148 -               ELMMVTIM 014E -
 11.24.38     ELMMUSID 0152 -               ELMRDSTP 015A - 021106        ELMRTSTP 0160 - 1124
 11.24.38     ELMRIID  0164 - OLEJU01       ELMRCOM  016C - TEST EXITS - TRACE
 11.24.38     ELMTDSN  0194 - BST.OLEJU40.MACLIB
 11.24.38     ELMTMBR  01C0 - $ENMANCS      ELMRCCID 01E8 - RELTST40      ELMGCCID 01F4 - REL4.0TST
 11.24.38     ELMODACT 0200 - ADD           ELMLCCID 0208 - REL4.0TST
 11.24.38     ELMLCOMM 0214 - test type sequence                          ELMLUID  023C - OLEJU01
 11.24.38     ELMLADT  0244 - 021106        ELMLATM  024A - 0857          ELMSOID  024E - OLEJU01
 11.24.38     ELFRFLAG 0256 - 00            ELFRACT  0257 -               ELFRUSID 025F -
 11.24.38     ELFRDATE 0267 -               ELFRTIME 026D -               ELFRSITE 0271 -
 11.24.38     ELFRENV  0272 -               ELFRSYS  027A -               ELFRSBS  0282 -
 11.24.38     ELFRELM  028A -               ELFRTYPE 0294 -               ELFRSTG# 029C -
 11.24.38     ELFRVER  029E - 0000          ELFRLVL  02A0 - 0000          ELMUPDT  02A2 - ER2HZTA0
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:24:38     PAGE     9
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                          C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.38     ELMBASE  02AC - $ENMANCS      ELMLOWL  02B6 - 00            ELMIPCTG 02B8 - 00
 11.24.38     ELMDPCTG 02B9 - 00            ELMFPESD 02BA -               ELMEDFMT 02C2 - R
 11.24.38     ELMENPAK 02C3 - Y             ELMLPROV 02C4 - 00            ELMPROL  02C5 - 00
 11.24.38     ELMFMID  02C6 - 29            ELMXDNAM 02C7 -               ELMXBVER 02D0 - 0000
 11.24.38     ELMXBTOT 02D4 - 00000000      ELMXLTOT 02D8 - 00000000      ELMXBLVL 02DC - 0000
 11.24.38     ELMXLLVL 02DE - 0000          ELMXINS  02E0 - 0000          ELMXDLS  02E2 - 0000
 11.24.38     ELMXRGIN 02E4 - 0000          ELMXRGDL 02E6 - 0000          ELMXBDTE 02E8 -
 11.24.38     ELMXBTIM 02EE -               ELMXLDTE 02F2 -               ELMXLTIM 02F8 -
 11.24.38     ELMXFLAG 02FC - 00            ELMXDFMT 02FD -               ELMXUPDT 02FE -
 11.24.38     ELMSPKG  02FF -                                             ELMSPKGT 030F - 0000000000000000
 11.24.38     ELMOPKG  0317 -                                             ELMOPKGT 0327 - 0000000000000000
 11.24.38     ELMPRGRP 032F - *NOPROC*      ELMCPKG  0337 -
 11.24.38     ELMCPKGT 0347 - 0000000000000000                            ELMFFFLG 034F - D
 11.24.38     ELMTFFLG 0350 - D             ELMFPAOF 0352 - 0000          ELMFNAOF 0354 - 0000
 11.24.38     ELMTPAOF 0356 - 0000          ELMTNAOF 0358 - 0000          ELMEAOFF 035A - 037C
 11.24.38     ELMBFARE 037C - 00085BC5D5D4C1D5C3E200000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK $ENVDS   AT 0007AB38  - ANCHORED AT ENVNEXT
 11.24.38     ENVLEN   0000 - 01C9          ENVVERS  0002 - 0003          ENVID    0004 - $ENV
 11.24.38     ENVFLAG  0008 - 0000          ENVSPEC  000A - E             ENVIOTY  000B - O
 11.24.38     ENVNEXT  000C - 00000000      ENVFCBA  0010 - 0007B170      ENVSITE  0014 - 0
 11.24.38     ENVSTGID 0015 - 0             ENVSTGCD 0016 -               ENVDATAC 0017 - N
 11.24.38     ENVEVER  0018 - 0000          ENVELVL  001A - 0000          ENVENVM  001C -
 11.24.38     ENVSTAGE 0024 -               ENVSYSTM 002C -               ENVSYSUD 0034 -
 11.24.38     ENVSUBSY 0044 -               ENVSUBUD 004C -               ENVTYPE  0054 -
 11.24.38     ENVTYPUD 005C -               ENVELEMT 0064 -               ENVEAOFF 00C4 - 0000
 11.24.38     ENVBFARE 00C8 - 00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK $FILDS   AT 0007B170  - ANCHORED AT ENVFCBA
 11.24.38     FILLEN   0000 - 0466          FILVERS  0002 - 0002          FILID    0004 - $FIL
 11.24.38     FILDSN   0008 - BST.OLEJU40.MACLIB
 11.24.38     FILDSMEM 0034 - $ENMANCS      FILDDN   003E - SYS00007       FILDSTY  0046 - PO
 11.24.38     FILPAOFF 004A - 0000          FILNAOFF 004C - 0000
 11.24.38     FILBFARE 0063 - 00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK $SMFHDDS AT 0007B5D8
 11.24.38     SMHLEN   0000 - 01C0          SMHSEGID 0002 - 0000          SMHSID   0004 - 02
 11.24.38     SMHRECTY 0005 - E6            SMHTIME  0006 - 003EAE4C      SMHDATE  000A - 0102310F
 11.24.38     SMHCPUID 000E - CA31          SMHHDLEN 0012 - 0030          SMHC1VER 0014 - 01
 11.24.38     SMHCTLTY 0015 - 02            SMHCONT  0016 - 0000          SMHSEQ   0018 - 0000
 11.24.38     SMHC1ENV 001A - I40           SMHUSER  0022 - OLEJU01
 11.24.38  CONTROL BLOCK $SMFREC2 AT 0007B608
 11.24.38     SM2RECLN 0000 - 0190          SM2RECVN 0002 - 0001          SM2FUNC  0004 - 00000003
 11.24.38     SM2FUNNM 0008 - RETRIEVE      SM2BLKS  0010 - 00000003      SM2RECHD 0014 - 0018
 11.24.38     SM2IFUNC 0016 - 0000
 11.24.38  CONTROL BLOCK SM2REQDS AT 0007B620
 11.24.38     SM2RLEN  0000 - 0060          SM2RTYP  0002 - 0004          SM2RVER  0004 - 0001
 11.24.38     SM2RFLG1 0006 -               SM2RFLG2 0007 -               SM2RCCID 0008 - RELTST40
 11.24.38     SM2RCOMM 0014 - TEST EXITS - TRACE                          SM2RFLAG 003C - 00000000
 11.24.38     SM2RSISO 0040 - N             SM2RSICO 0041 - N             SM2REXPN 0042 - N
 11.24.38     SM2ROVER 0043 - N             SM2RRTCD 0044 - 00000000      SM2RDEL  0048 -
 11.24.38     SM2RIGNF 0049 -               SM2RBGNP 004A -               SM2RBDLP 004B -
 11.24.38     SM2RSYNC 004C -               SM2RONCP 004D -               SM2RMVWH 004E -
 11.24.38     SM2RAWUP 004F -               SM2RPGRP 0050 -               SM2RSIUS 0058 -
 11.24.38  CONTROL BLOCK SM2ENVDS AT 0007B680
 11.24.38     SM2ELEN  0000 - 0082          SM2ETYP  0002 - 0001          SM2EVER  0004 - 0002
 11.24.38     SM2EFLG1 0006 -               SM2EFLG2 0007 -               SM2SITE  0008 - 0
 11.24.38     SM2STGCD 0009 - 1             SM2STGID 000A - F1            SM2VER   000C - 0001
 11.24.38     SM2LVL   000E - 0000          SM2ENVM  0010 - I40           SM2STAGE 0018 - I40STG1
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:24:38     PAGE    10
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                                C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.38     SM2SYSTM 0020 - NDVRMVS       SM2SUBSY 0028 - JUDY          SM2STYPE 0030 - ASMMAC
 11.24.38     SM2ELEMT 0038 -  L            SM2EAOFF 0038 - 0078
 11.24.38     SM2DSNAM 0042 -
 11.24.38     SM2FAOFF 0042 - 0000          SM2DSMEM 006E -               SM2NAOFF 006E - 0000
 11.24.38     SM2EAREA 0078 - 00085BC5D5D4C1D5C3E2009600010002000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK SM2ENVDS AT 0007B702
 11.24.38     SM2ELEN  0000 - 0096          SM2ETYP  0002 - 0001          SM2EVER  0004 - 0002
 11.24.38     SM2EFLG1 0006 -               SM2EFLG2 0007 -               SM2SITE  0008 -
 11.24.38     SM2STGCD 0009 -               SM2STGID 000A - 00            SM2VER   000C - 0000
 11.24.38     SM2LVL   000E - 0000          SM2ENVM  0010 -               SM2STAGE 0018 -
 11.24.38     SM2SYSTM 0020 -               SM2SUBSY 0028 -               SM2STYPE 0030 -
 11.24.38     SM2ELEMT 0038 -               SM2EAOFF 0038 - 0000
 11.24.38     SM2DSNAM 0042 -  L
 11.24.38     SM2FAOFF 0042 - 0078          SM2DSMEM 006E -  p            SM2NAOFF 006E - 008C
 11.24.38     SM2EAREA 0078 - 0012C2E2E34BD6D3C5D1E4F4F04BD4C1C3D3C9C200085BC5D5D4C1D5C3E200000000000000000000000000000000000000000
011.24.38  AFTER   EXIT 03 CALLED, R1(00078D10) - 00078EC8 0007A968 0007B5D8
 11.24.38  CONTROL BLOCK $ECBDS   AT 00078EC8
 11.24.38     ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
 11.24.38     ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000003
 11.24.38     ECBEXTN  0014 - 00000003      ECBUSER  0018 - OLEJU01       ECBMODE  0020 - B
 11.24.38     ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - Y
 11.24.38     ECBFUNC  0024 - 00000003      ECBFUNAM 0028 - RETRIEVE      ECBMSGCD 0030 - 0000
 11.24.38     ECBMSGLN 0034 - 0000
 11.24.38     ECBMSG   0036 -
 11.24.38     ECBCALR  00BA - E             ECBPKG   00BB - N             ECBREQA  00BC - 00078F98
 11.24.38     ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000      ECBIFUNC 00C8 - 00000000
 11.24.38     ECBHIRC  00CC - 00000000
 11.24.38  CONTROL BLOCK $REQPDS  AT 00078F98  - ANCHORED AT ECBREQA
 11.24.38     REQLEN   0000 - 006C          REQVERS  0002 - 0002          REQID    0004 - $REQ
 11.24.38     REQCCID  0008 - RELTST40      REQCOMM  0014 - TEST EXITS - TRACE
 11.24.38     REQFLAG  003C - 00000000      REQSISOF 0040 - N             REQSISOC 0041 - N
 11.24.38     REQEXPND 0042 - N             REQOVRWR 0043 - N             REQRTCOD 0044 - 00000000
 11.24.38     REQELVER 0048 -               REQCPYBK 004A -               REQDEL   0050 -
 11.24.38     REQMOVWH 0051 -               REQUPDT  0052 -               REQBYGEN 0053 -
 11.24.38     REQONLCM 0054 -               REQSYNC  0055 -               REQIGGF  0056 -
 11.24.38     REQBYDP  0057 -               REQOPGRP 0058 -               REQSOUID 0060 -
 11.24.38     REQOSRCH 0068 - Y             REQORSGN 0069 -               REQOJUMP 006A -
 11.24.38     REQOSGNI 006B -
 11.24.38  CONTROL BLOCK $ENVDS   AT 0007A968
 11.24.38     ENVLEN   0000 - 01C9          ENVVERS  0002 - 0003          ENVID    0004 - $ENV
 11.24.38     ENVFLAG  0008 - 0000          ENVSPEC  000A - C             ENVIOTY  000B - I
 11.24.38     ENVNEXT  000C - 0007AB38      ENVFCBA  0010 - 00079048      ENVSITE  0014 - 0
 11.24.38     ENVSTGID 0015 - 1             ENVSTGCD 0016 - 1             ENVDATAC 0017 -
 11.24.38     ENVEVER  0018 - 0001          ENVELVL  001A - 0000          ENVENVM  001C - I40
 11.24.38     ENVSTAGE 0024 - I40STG1       ENVSYSTM 002C - NDVRMVS       ENVSYSUD 0034 -
 11.24.38     ENVSUBSY 0044 - JUDY          ENVSUBUD 004C -               ENVTYPE  0054 - ASMMAC
 11.24.38     ENVTYPUD 005C -               ENVELEMT 0064 - $ENMANCS      ENVEAOFF 00C4 - 00C8
 11.24.38     ENVBFARE 00C8 - 00085BC5D5D4C1D5C3E200000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK $ELMDS   AT 00079048  - ANCHORED AT ENVFCBA
 11.24.38     ELMLEN   0000 - 0C8C          ELMVERS  0002 - 0005          ELMID    0004 - $ELM
 11.24.38     ELMFLAGS 0008 - 00000000      ELMMVERS 000C - 0001          ELMMLLVL 000E - 0000
 11.24.38     ELMMINS  0010 - 00000000      ELMMDEL  0014 - 00000000      ELMNAME  0018 - $ENMANCS
 11.24.38     ELMCCID  0022 - REL4.0TST     ELMM1STL 002E - 0000          ELMMBTOT 0030 - 00000077
 11.24.38     ELMMBDTE 0034 - 021106        ELMMBTIM 003A - 0857
 11.24.38     ELMBCOM  003E - test type sequence                          ELMBUSID 0066 - OLEJU01
```

```
1 (C) 2002 Computer Associates International, Inc              Endevor                              11/06/02  11:24:38     PAGE    11
                                                                                                          RELEASE 4.0   SERIAL B4000C
                                          C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.38     ELMPUSID 006E - OLEJU01       ELMPDSTP 0076 - 021106        ELMPTSTP 007C - 0857
 11.24.38     ELMFDSN  0080 - BST.OLEJU40.MACLIB
 11.24.38     ELMFMBR  00AC - @ENMANCS      ELMPRFLG 00B6 - 40            ELMLPROD 00BA - 021106
 11.24.38     ELMLPROT 00C0 - 0857          ELMLPROU 00C4 - OLEJU01       ELMLPRON 00CC - *NOPROC*
 11.24.38     ELMMPRC  00D6 - FFFF          ELMMRC   00D8 - 0000
 11.24.38     ELMPCOM  00DA - test type sequence
 11.24.38     ELMCCOM  0102 - test type sequence                          ELMMLDTE 012A - 021106
 11.24.38     ELMMLTME 0130 - 0857          ELMLUSID 0134 - OLEJU01       ELMMLACT 013C - RETRIEVE
 11.24.38     ELMMLTOT 0144 - 00000077      ELMMVDTE 0148 -               ELMMVTIM 014E -
 11.24.38     ELMMUSID 0152 -               ELMRDSTP 015A - 021106        ELMRTSTP 0160 - 1124
 11.24.38     ELMRIID  0164 - OLEJU01       ELMRCOM  016C - TEST EXITS - TRACE
 11.24.38     ELMTDSN  0194 - BST.OLEJU40.MACLIB
 11.24.38     ELMTMBR  01C0 - $ENMANCS      ELMRCCID 01E8 - RELTST40      ELMGCCID 01F4 - REL4.0TST
 11.24.38     ELMODACT 0200 - ADD           ELMLCCID 0208 - REL4.0TST
 11.24.38     ELMLCOMM 0214 - test type sequence                          ELMLUID  023C - OLEJU01
 11.24.38     ELMLADT  0244 - 021106        ELMLATM  024A - 0857          ELMSOID  024E - OLEJU01
 11.24.38     ELFRFLAG 0256 - 00            ELFRACT  0257 -               ELFRUSID 025F -
 11.24.38     ELFRDATE 0267 -               ELFRTIME 026D -               ELFRSITE 0271 -
 11.24.38     ELFRENV  0272 -               ELFRSYS  027A -               ELFRSBS  0282 -
 11.24.38     ELFRELM  028A -               ELFRTYPE 0294 -               ELFRSTG# 029C -
 11.24.38     ELFRVER  029E - 0000          ELFRLVL  02A0 - 0000          ELMUPDT  02A2 - ER2HZTA0
 11.24.38     ELMBASE  02AC - $ENMANCS      ELMLOWL  02B6 - 00            ELMIPCTG 02B8 - 00
 11.24.38     ELMDPCTG 02B9 - 00            ELMFPESD 02BA -               ELMEDFMT 02C2 - R
 11.24.38     ELMENPAK 02C3 - Y             ELMLPROV 02C4 - 00            ELMPROL  02C5 - 00
 11.24.38     ELMFMID  02C6 - 29            ELMXDNAM 02C7 -               ELMXBVER 02D0 - 0000
 11.24.38     ELMXBTOT 02D4 - 00000000      ELMXLTOT 02D8 - 00000000      ELMXBLVL 02DC - 0000
 11.24.38     ELMXLLVL 02DE - 0000          ELMXINS  02E0 - 0000          ELMXDLS  02E2 - 0000
 11.24.38     ELMXRGIN 02E4 - 0000          ELMXRGDL 02E6 - 0000          ELMXBDTE 02E8 -
 11.24.38     ELMXBTIM 02EE -               ELMXLDTE 02F2 -               ELMXLTIM 02F8 -
 11.24.38     ELMXFLAG 02FC - 00            ELMXDFMT 02FD -               ELMXUPDT 02FE -
 11.24.38     ELMSPKG  02FF -                                             ELMSPKGT 030F - 0000000000000000
 11.24.38     ELMOPKG  0317 -                                             ELMOPKGT 0327 - 0000000000000000
 11.24.38     ELMPRGRP 032F - *NOPROC*      ELMCPKG  0337 -
 11.24.38     ELMCPKGT 0347 - 0000000000000000                            ELMFFFLG 034F - D
 11.24.38     ELMTFFLG 0350 - D             ELMFPAOF 0352 - 0000          ELMFNAOF 0354 - 0000
 11.24.38     ELMTPAOF 0356 - 0000          ELMTNAOF 0358 - 0000          ELMEAOFF 035A - 037C
 11.24.38     ELMBFARE 037C - 00085BC5D5D4C1D5C3E200000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK $ENVDS   AT 0007AB38  - ANCHORED AT ENVNEXT
 11.24.38     ENVLEN   0000 - 01C9          ENVVERS  0002 - 0003          ENVID    0004 - $ENV
 11.24.38     ENVFLAG  0008 - 0000          ENVSPEC  000A - E             ENVIOTY  000B - O
 11.24.38     ENVNEXT  000C - 00000000      ENVFCBA  0010 - 0007B170      ENVSITE  0014 - 0
 11.24.38     ENVSTGID 0015 - 0             ENVSTGCD 0016 -               ENVDATAC 0017 - N
 11.24.38     ENVEVER  0018 - 0000          ENVELVL  001A - 0000          ENVENVM  001C -
 11.24.38     ENVSTAGE 0024 -               ENVSYSTM 002C -               ENVSYSUD 0034 -
 11.24.38     ENVSUBSY 0044 -               ENVSUBUD 004C -               ENVTYPE  0054 -
 11.24.38     ENVTYPUD 005C -               ENVELEMT 0064 -               ENVEAOFF 00C4 - 0000
 11.24.38     ENVBFARE 00C8 - 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK $FILDS   AT 0007B170  - ANCHORED AT ENVFCBA
 11.24.38     FILLEN   0000 - 0466          FILVERS  0002 - 0002          FILID    0004 - $FIL
 11.24.38     FILDSN   0008 - BST.OLEJU40.MACLIB
 11.24.38     FILDSMEM 0034 - $ENMANCS      FILDDN   003E - SYS00007       FILDSTY  0046 - PO
 11.24.38     FILPAOFF 004A - 0000          FILNAOFF 004C - 0000
 11.24.38     FILBFARE 0063 - 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK $SMFHDDS AT 0007B5D8
 11.24.38     SMHLEN   0000 - 01C0          SMHSEGID 0002 - 0000          SMHSID   0004 - 02
 11.24.38     SMHRECTY 0005 - E6            SMHTIME  0006 - 003EAE4C      SMHDATE  000A - 0102310F
```

```
1 (C) 2002 Computer Associates International, Inc             Endevor                          11/06/02  11:24:38     PAGE    12
                                                                                                     RELEASE 4.0   SERIAL B4000C
                                        C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.38    SMHCPUID 000E - CA31          SMHHDLEN 0012 - 0030          SMHC1VER 0014 - 01
 11.24.38    SMHCTLTY 0015 - 02            SMHCONT  0016 - 0000          SMHSEQ   0018 - 0000
 11.24.38    SMHC1ENV 001A - I40           SMHUSER  0022 - OLEJU01
 11.24.38  CONTROL BLOCK $SMFREC2 AT 0007B608
 11.24.38    SM2RECLN 0000 - 0190          SM2RECVN 0002 - 0001          SM2FUNC  0004 - 00000003
 11.24.38    SM2FUNNM 0008 - RETRIEVE      SM2BLKS  0010 - 00000003      SM2RECHD 0014 - 0018
 11.24.38    SM2IFUNC 0016 - 0000
 11.24.38  CONTROL BLOCK SM2REQDS AT 0007B620
 11.24.38    SM2RLEN  0000 - 0060          SM2RTYP  0002 - 0004          SM2RVER  0004 - 0001
 11.24.38    SM2RFLG1 0006 -               SM2RFLG2 0007 -               SM2RCCID 0008 - RELTST40
 11.24.38    SM2RCOMM 0014 - TEST EXITS - TRACE                          SM2RFLAG 003C - 00000000
 11.24.38    SM2RSISO 0040 - N             SM2RSICO 0041 - N             SM2REXPN 0042 - N
 11.24.38    SM2ROVER 0043 - N             SM2RRTCD 0044 - 00000000      SM2RDEL  004B -
 11.24.38    SM2RIGNF 0049 -               SM2RBGNP 004A -               SM2RBDLP 004B -
 11.24.38    SM2RSYNC 004C -               SM2RONCP 004D -               SM2RMVWH 004E -
 11.24.38    SM2RAWUP 004F -               SM2RPGRP 0050 -               SM2RSIUS 0058 -
 11.24.38  CONTROL BLOCK SM2ENVDS AT 0007B680
 11.24.38    SM2ELEN  0000 - 0082          SM2ETYP  0002 - 0001          SM2EVER  0004 - 0002
 11.24.38    SM2EFLG1 0006 -               SM2EFLG2 0007 -               SM2SITE  0008 - 0
 11.24.38    SM2STGCD 0009 - 1             SM2STGID 000A - F1            SM2VER   000C - 0001
 11.24.38    SM2LVL   000E - 0000          SM2ENVM  0010 - I40           SM2STAGE 0018 - I40STG1
 11.24.38    SM2SYSTM 0020 - NDVRMVS       SM2SUBSY 0028 - JUDY          SM2STYPE 0030 - ASMMAC
 11.24.38    SM2ELEMT 0038 -  L            SM2EAOFF 0038 - 0078
 11.24.38    SM2DSNAM 0042 -
 11.24.38    SM2FAOFF 0042 - 0000          SM2DSMEM 006E -               SM2NAOFF 006E - 0000
 11.24.38    SM2EAREA 0078 - 00085BC5D5D4C1D5C3E2009600010002000000000000000000000000000000000000000000000000000000000000000000000000000000
 11.24.38  CONTROL BLOCK SM2ENVDS AT 0007B702
 11.24.38    SM2ELEN  0000 - 0096          SM2ETYP  0002 - 0001          SM2EVER  0004 - 0002
 11.24.38    SM2EFLG1 0006 -               SM2EFLG2 0007 -               SM2SITE  0008 -
 11.24.38    SM2STGCD 0009 -               SM2STGID 000A - 00            SM2VER   000C - 0000
 11.24.38    SM2LVL   000E - 0000          SM2ENVM  0010 -               SM2STAGE 0018 -
 11.24.38    SM2SYSTM 0020 -               SM2SUBSY 0028 -               SM2STYPE 0030 -
 11.24.38    SM2ELEMT 0038 -               SM2EAOFF 0038 - 0000
 11.24.38    SM2DSNAM 0042 -  L
 11.24.38    SM2FAOFF 0042 - 0078          SM2DSMEM 006E -  p            SM2NAOFF 006E - 008C
 11.24.38    SM2EAREA 0078 - 0012C2E2E34BD6D3C5D1E4F4F04BD4C1C3D3C9C200085BC5D5D4C1D5C3E200000000000000000000000000000000000000000
011.24.53  EXIT MODULE BC1PNM00 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
011.24.53  BEFORE  EXIT 06 CALLED, R1(00076520) - 000766D8
 11.24.53  CONTROL BLOCK $XITDS   AT 00075CF4
 11.24.53    XITLEN   0000 - 006C          XITVERS  0002 - 0001          XITID    0004 - $XIT
 11.24.53    XITNUMB  0008 - 00000006      XITFUNC  000C - 00000006      XITSMF   0010 - N
 11.24.53    XITINTNL 0011 - N             XITSECFL 0012 - 0080          XITMDPRM 0014 - 000749EC
 11.24.53    XITPARMB 0018 - 00075D20      XITPNLLV 001C - 00000000      XITARQP  0020 - 00000000
 11.24.53  CONTROL BLOCK $XPARMDS AT 00075D20  - ANCHORED AT XITPARMB
 11.24.53    XPALEN   0000 - 0112          XPAVERS  0002 - 0001          XPAID    0004 - $XPA
 11.24.53    XPA_ACT  0008 - DISPLAY       XPA_FUNC 0010 - DISPLAY       XPA_OPT  0018 - 00
 11.24.53    XPA$RSV1 0019 - 000000        XPA$ACEE 001C - 00000000      XPA1ENV  0038 -
 11.24.53    XPA1STG  0040 -               XPA1STNO 0048 - 00            XPA1STID 0049 -
 11.24.53    XPA1SYS  004A -               XPA1SBS  0052 -               XPA1ELM  005C - 00000000
 11.24.53    XPA1TYPE 0060 -               XPA1FLGS 0068 - M             XPA1LOC  0069 - C
 11.24.53    XPA1PTR1 006C - 00000000      XPA2ENV  0084 -               XPA2STG  008C -
 11.24.53    XPA2STNO 0094 - 00            XPA2STID 0095 -               XPA2SYS  0096 -
 11.24.53    XPA2SBS  009E -               XPA2ELM  00A8 - 00000000      XPA2TYPE 00AC -
 11.24.53    XPA2FLGS 00B4 - M             XPA2LOC  00B5 - C             XPA2PTR1 00B8 - 00000000

1 (C) 2002 Computer Associates International, Inc             Endevor                          11/06/02  11:24:53     PAGE    13
                                                                                                     RELEASE 4.0   SERIAL B4000C
                                        C1I0003I USER EXIT PARAMETER TRACE LOG

 11.24.53    XPA_PKGI 00D0 -                                             XPA_PKGT 00E0 -
 11.24.53    XPA_PKGS 00E8 -               XPA_PKGB 00F0 -               XPA_PKGS 00F1 -
 11.24.53    XPA_PKGA 00F2 -                                             XPA_PKGS 0102 -
 11.24.53    XPA_PKGD 010A -
 11.24.53  CONTROL BLOCK $ECBDS   AT 000766D8
 11.24.53    ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
 11.24.53    ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000001
 11.24.53    ECBEXTN  0014 - 00000006      ECBUSER  0018 - OLEJU01       ECBMODE  0020 - B
 11.24.53    ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - N
 11.24.53    ECBFUNC  0024 - 00000006      ECBFUNAM 0028 - DISPLAY       ECBMSGCD 0030 -
 11.24.53    ECBMSGLN 0034 - 0000
 11.24.53    ECBMSG   0036 -
 11.24.53    ECBCALR  00BA - E             ECBPKG   00BB - N             ECBREQA  00BC - 000767A8
 11.24.53    ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000      ECB1FUNC 00C8 - 00000000
 11.24.53    ECBHIRC  00CC - 00000000
011.24.53  AFTER   EXIT 06 CALLED, R1(00076520) - 000766D8
 11.24.53  CONTROL BLOCK $ECBDS   AT 000766D8
 11.24.53    ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
 11.24.53    ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000001
 11.24.53    ECBEXTN  0014 - 00000006      ECBUSER  0018 - OLEJU01       ECBMODE  0020 - B
 11.24.53    ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - N
 11.24.53    ECBFUNC  0024 - 00000006      ECBFUNAM 0028 - DISPLAY       ECBMSGCD 0030 -
 11.24.53    ECBMSGLN 0034 - 0000
 11.24.53    ECBMSG   0036 -
 11.24.53    ECBCALR  00BA - E             ECBPKG   00BB - N             ECBREQA  00BC - 000767A8
 11.24.53    ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000      ECB1FUNC 00C8 - 00000000
 11.24.53    ECBHIRC  00CC - 00000000
```

A comparison of the after-action control block information for the modifiable fields indicates that the information has remained the same. This means that the program is functioning normally.

```
AFTER   EXIT 02 CALLED, R1(0004B658) - 0004B7D0 0004C278
CONTROL BLOCK $ECBDS   AT 0004B7D0
    ECBLEN   0000 - 00D0          ECBVERS  0002 - 0002          ECBID    0004 - $ECB
    ECBFLAG  0008 - 00000000      ECBRTCD  000C - 00000000      ECBARGS  0010 - 00000002
    ECBEXTN  0014 - 00000002      ECBUSER  0018 - DA1CC47       ECBMODE  0020 - T
    ECBINTNL 0021 - N             ECBSMFAC 0022 - Y             ECBSMFWR 0023 - N
    ECBFUNC  0024 - 00000003      ECBFUNAM 0028 - RETRIEVE      ECBMSGCD 0030 -
    ECBMSGLN 0034 - 0000
```

```
    ECBMSG   0036 -
    ECBREQA  00BC - 0004C4C8      ECBPNLT  00C0 - 00000000      ECBUEXT  00C4 - 00000000
    ECBIFUNC 00C8 - 00000000
CONTROL BLOCK $REQPDS  AT 0004C4C8  - ANCHORED AT ECBREQA
    REQLEN   0000 - 006C          REQVERS  0002 - 0002          REQID    0004 - $REQ
    REQCCID  0008 - 11            REQCOMM  0014 - LOOKING FOR PROBLEM
    REQFLAG  003C - 00000000      REQSISOF 0040 - Y             REQSISOC 0041 - N
    REQEXPND 0042 - N             REQOVRWR 0043 - Y             REQRTCOD 0044 - 00000000
    REQELVER 0048 -
    REQMOVWH 0051 -
    REQONLCM 0054 -
    REQBYDP  0057 -
    REQOSRCH 0068 - N             REQORSGN 0069 -
    REQOSGNI 006B -
```

1.2 Identifying Exits to Endevor

1.2.1 Overview

You identify exit programs to Endevor by adding an @C1UEXIT macro to the source code of the user exits table (C1UEXITS). The JCL which builds the user exit table and sample source code can be found in the BC1JXITS member in the iprfx.iqual.JCL library. During Endevor execution, within each exit number, the exits are called in the order defined in the exit point within the load module.

1.2.2 BC1JXITS Job

```
//*(JOBCARD)
//*-------------------------------------------------------------------*
//*                                                                   *
//*   (C) 1987,2000 COMPUTER ASSOCIATES INTERNATIONAL, INC.           *
//*                                                                   *
//* NAME: BC1JXITS                                                    *
//*                                                                   *
//*    BC1JXITS - THIS JOB IS USED TO ADD ADDITIONAL MULTIPLE         *
//*               USER EXITS TO ENDEVOR-MVS.  THE TABLE CREATED       *
//*               IN STEP 2 IS CALLED C1UEXITS.                       *
//*                                                                   *
//*    THE FOLLOWING UPDATES MUST BE MADE TO THIS JCL BEFORE          *
//*    IT CAN BE EXECUTED:                                            *
//*                                                                   *
//*    1. UPDATE THE JOBCARD TO REFLECT CORRECT SITE INFORMATION      *
//*    2. REVIEW THAT THE SOURCE AND LOAD LIBRARY DATA SET NAMES ARE  *
//*       CORRECT.                                                    *
//*             - iprfx.iqual.SOURCE                                  *
//*             - iprfx.iqual.AUTHLIB                                 *
//*-------------------------------------------------------------------*
//STEP1    EXEC PGM=ASMA90,PARM='NODECK,OBJECT,NOTERM'
//SYSLIB   DD  DISP=SHR,DSN=BST.SUPPORT.REL393.SOURCE
//         DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSLIN   DD  DISP=(,PASS,DELETE),DSN=&&SYSLIN,
//             UNIT=SYSDA,SPACE=(TRK,(3,5),RLSE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSPUNCH DD  DUMMY
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(5,3))
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD   *
C1UEXITS TITLE 'DEFINE USER EXITS'
*-------------------------------------------------------------------*
*   MULTIPLE USER EXIT SUPPORT                                      *
*                                                                   *
*   TYPE=START DEFINITION MUST PROCEDE ANY EXIT DEFINITION STATEMENT. *
C1UEXITS @C1UEXIT TYPE=START,XIT7BAT=YES
*-------------------------------------------------------------------*
*   SEE ENDEVOR-MVS EXITS MANUAL FOR A DESCRIPTION OF               *
```

```
*   DEFINING MULTIPLE USER EXITS.                                        *
*----------------------------------------------------------------------*
         @C1UEXIT EXIT#=1,NAME=C1UEXT01,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=2,NAME=C1UEXT02,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=3,NAME=C1UEXT03,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=4,NAME=C1UEXT04,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=5,NAME=C1UEXT05,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=6,NAME=C1UEXT06,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=7,NAME=C1UEXT07,ANCHID=0,AUTH=YES
*----------------------------------------------------------------------*
*   TYPE=DEFINITION DENOTES THE END OF THE EXIT TABLE.                 *
*----------------------------------------------------------------------*
         @C1UEXIT TYPE=END
         END
//*
//STEP2    EXEC PGM=IEWL,PARM='LIST,NCAL,RENT,XREF,SIZE=(256K,64K)',
//         COND=(0,NE)
//SYSLIN   DD  DISP=(OLD,DELETE),DSN=&&SYSLIN
//SYSLMOD  DD  DISP=SHR,DSN=BST.SUPPORTU.REL393.AUTHLIB(C1UEXITS)
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(5,3))
//SYSPRINT DD  SYSOUT=*
```

1.2.3 @C1UEXIT Macro Statements

The C1UEXITS source consists of a series of @C1UEXIT macro statements. The three varieties of @C1UEXIT macros statements are shown in the following list:

- @C1UEXIT TYPE=START...
- @C1UEXIT EXIT#=number,NAME=program...
- @C1UEXIT TYPE=END

One TYPE=START macro must be coded as the first macro in the table source. One TYPE=END macro must be coded as the last macro in the table source. One EXIT#= macro should be coded for each exit program that you identify to Endevor.

1.2.4 @C1UEXIT TYPE=START...

@C1UEXIT TYPE=START...identifies the start of the user exit table and sets a default value of Yes or No for PECBBATX (an exit 7 setup flag described in Enabling Exit Points). The format is as follows:

```
@C1UEXIT TYPE=START,XIT7BAT=YES
```

Where: *XIT7BAT=*

Sets a default value for PECBBATX. If this keyword is not coded, a severity 4 MNOTE is generated at assembly time and a default value of "No" is chosen. Acceptable values are:

PECBBATX to "N" during the Setup call.

- **No--** Indicates that all or most package exits cannot run in batch (without TSO services). Those exits that can run in batch should set PECBBATX to "Y" during the Setup call.
- **Yes--** Indicates that all or most package exits can run in batch (without TSO services). Exits that cannot run in batch must set PECBBATX to "N" during the Setup call.

1.2.5 @C1UEXIT EXIT#=number,NAME=program,ANCHID=id,AUTH=yes|no

@C1UEXIT EXIT#=number,NAME=program... identifies a program as a Endevor exit. You must code a single macro of this format for each exit module to be identified. The format is as follows:

```
@C1UEXIT EXIT#=1,NAME=C1UEXT01,ANCHID=0,AUTH=YES
```

Where:

- *EXIT#=*

  The exit point where you want Endevor to call the exit program.

  Acceptable values are:

  - **1**
  - **2**
  - **3**
  - **4**
  - **5**
  - **6**
  - **7**

- *NAME=*

  Defines the program name. This name must be the same as the name of the called program.

- *ANCHID=*—Identifies the anchor ID for this user exit. The valid range for an anchor ID is 0-9999. If an anchor ID is not coded, the default is zero (0).
- *AUTH=*—Indicates whether the exit program is stored in an authorized library.

  Acceptable values are:

  - **Yes—** Default. Indicates that the user exits are stored in an authorized library.
  - **No—** Indicates that the user exits are stored in either a STEPLIB or CONLIB. Use this option when testing exits.

1.2.6 @C1UEXIT TYPE=END

@C1UEXIT TYPE=END identifies the end of the user exit table. The format is as follows:

```
@C1UEXIT TYPE=END
```

1.2.7 Using Anchor IDs with Multiple User Exits

Anchor IDs allow you to store information produced by one user exit that needs to be passed to one or more other user exits. To do this, assign the same anchor ID to all the exits that use the information.

Anchor IDs work as follows:

- Endevor initialization logic builds an anchor ID table, associating each anchor ID with a default value of 0000. Anchor IDs can have a value of 0-9999. Endevor user exits have an anchor ID of 0.
- Before a user exit is invoked, Endevor copies the value stored in the anchor ID table into the ECBUEXT field in the $ECBDS DSECT (described in $ECBDS DSECT: Exit Control Block). Endevor then executes the user exit.
- The user exit, if necessary, can modify the ECBUEXT field in the $ECBDS DSECT. After the exit terminates, Endevor copies the value in the ECBUEXT field into the anchor ID table.

1.2.8 Example

Assume you want to specify three additional exits to be supported by Endevor, and that the information produced by exit 1 will be used by exits 3 and 6. You might specify:

- An exit 1 routine called SECEXIT1, with an anchor ID of 2.
- An exit 3 routine called RECEXIT3, with an anchor ID of 2.
- An exit 6 routine called STATEXT6, with an anchor ID of 2.

You would then modify the exit table by adding three lines to the @C1UEXIT macro in the BC1JXIT JCL, as follows:

```
*-------------------------------------------------------------------*
C1UEXITS @C1UEXIT TYPE=START,XIT7BAT=YES
*-------------------------------------------------------------------*
*   SEE ENDEVOR EXITS MANUAL FOR A DESCRIPTION OF      *
*   DEFINING MULTIPLE USER EXITS.                                   *
*-------------------------------------------------------------------*
         @C1UEXIT EXIT#=1,NAME=C1UEXT01,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=1,NAME=SECEXIT1,ANCHID=2,AUTH=YES
         @C1UEXIT EXIT#=2,NAME=C1UEXT02,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=3,NAME=C1UEXT03,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=3,NAME=RECEXIT3,ANCHID=2,AUTH=YES
         @C1UEXIT EXIT#=4,NAME=C1UEXT04,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=5,NAME=C1UEXT05,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=6,NAME=C1UEXT06,ANCHID=0,AUTH=YES
         @C1UEXIT EXIT#=6,NAME=STATEXT6,ANCHID=2,AUTH=YES
         @C1UEXIT EXIT#=7,NAME=C1UEXT07,ANCHID=0,AUTH=YES
*-------------------------------------------------------------------*
*   TYPE=DEFINITION DENOTES THE END OF THE EXIT TABLE.              *
*-------------------------------------------------------------------*
         @C1UEXIT TYPE=END
         END
```

When Endevor reaches the first exit point, both program C1UEXT01 and program SECEXIT1 will be called, in the order in which they are listed. This processing proceeds as follows:

Step	Action
1	Prior to calling program SECEXIT1, Endevor reads the value associated with anchor ID 2 in the anchor ID table, and copies this value into the ECBUEXT field of the $ECBDS DSECT.
2	Endevor then invokes SECEXIT1, passing the $ECBDS DSECT or one of the parameters.
3	Program SECEXIT1 terminates and returns to Endevor.
4	Endevor copies the value in the ECBUEXT field of the $ECBDS DSECT into the anchor ID table slot associated with anchor ID 2.

When exit point 3 is reached, Endevor calls both C1UEXT03 and RECEXIT3. Execution of program RECEXIT3 follows the four steps described above. Note that the value associated with anchor ID 2 prior to calling RECEXIT3 in the anchor ID table will be the value in the ECBUEXT field after the execution of program SECEXIT1.

When exit point 6 is reached, Endevor calls both the C1UEXT06 and STATEXT6 routines. Execution of program STATEXT6 follows the four steps described above. Note that the value associated with anchor ID 2 in the anchor ID table prior to calling STATEXT6 will be the value in the ECBUEXT field after the execution of program RECEXIT3.

1.3 Documentation Overview

This manual is part of a comprehensive documentation set that fully describes the features and functions of Endevor and explains how to perform everyday tasks. For a complete list of Endevor manuals, see the PDF Table of Contents file in the PDF directory, or the Bookmanager Bookshelf file in the Books directory.

The following section describes product conventions.

1.4 Syntax for Long File and Path Names

The following considerations apply to the Path clause for ADD, UPDATE, COPY and RETRIEVE statements:

- The PATH clause is mutually exclusive with the FILE or Data Set clauses.
- The HFSFile clause is mutually exclusive with a Member clause.
- The path name must begin with a "/" and be terminated with a "/" and cannot be followed by the file name.
- The HFS file name can be up to 255 bytes in length.
- The path name can be up to 768 bytes in length.

1.4.1 HFSFile Syntax Rules

A filename can be up to 255 characters long. To be portable, the filename should use only the characters in the POSIX portable filename character set:

- Uppercase or lowercase A to Z
- Numbers 0 to 9
- Period (.)
- Underscore (_)
- Hyphen (-)

Do not include any nulls or slash characters in a filename.

Doublebyte characters are not supported in a filename and are treated as singlebyte data. *Using doublebyte characters in a filename may cause problems*. For instance, if you use a doublebyte character in which one of the bytes is a . (dot) or / (slash), the file system treats this as a special delimiter in the pathname.

The shells are case-sensitive, and distinguish characters as either uppercase or lowercase. Therefore, **FILE1** is not the same as **file1**.

A filename can include a suffix, or *extension*, that indicates its file type. An extension consists of a period (.) and several characters. For example, files that are C code could have the extension **.c**, as in the filename **dbmod3.c**. Having groups of files with identical suffixes makes it easier to run commands against many files at once.

1.4.2 Path Name Syntax Rules

The path name value can be up to 768 characters long. It can contain only the following characters:

- Uppercase letters
- Lowercase letters
- Numbers
- National characters
- Slash (/)
- Plus (+)
- Hyphen (-)
- Period (.)

1.4.3 Element Name Syntax Rules

The Element name can be up to 255 characters long. It can contain only the following characters:

- Uppercase letters
- Lowercase letters
- Numbers
- National characters
- Period (.)
- Hyphen (-)
- Underscore(_)

Element names name include a percent sign (%) in any column as a placeholder character in most SCL. The final one or more characters may be replaced in SCL and some panels with an asterisk (*) as a wild character for selection purposes.

1.4.4 SCL Continuation Syntax Rules

All SCL parameters that span multiple lines must be enclosed in single quotes. SCL keyword parameters cannot span multiple lines—only the parameter values. The syntax required to span a paramter value should lead with an apostrophe or quotation mark at the beginning of the specification and a trailing apostrophe or quotation mark of the value. Spaces that are not surrounded by non-blank characters will not be included in the text string. Example:

```
ADD ELEMENT 'Spanned
 ElementName'  CCID 'This is the chan
  ge control number'
```

This would result in an element value of "SpannedElementName" and a CCID value of "This is the change control number".

Chapter 2. Exit Point Descriptions

This chapter describes Endevor's exit points. It contains the following sections:

- Overview
- Exit Point 1: After Endevor Security Checks
- Exit Point 2: Before Actions
- Exit Point 3: After Actions
- Exit Point 4: Data Set Validation
- Exit Point 5: At Session Initialization
- Exit Point 6: Before Session Termination
- Example of Exit Use

2.1 Overview

2.1.1 Exit Point Functions

Endevor provides seven exit points. The list below summarizes the exit point functions. Note that when Endevor is reinvoked by executing Endevor from within a processor, each exit point is also reinvoked, according to this summary.

Exit point	Occurs
1	After each Endevor security check.
2	Before an action is executed--in either foreground or batch--after all input has been validated.
3	After an action has been completed and the MCF, base library, and/or delta library have been updated for the action. If SMF recording is in effect, this exit is called after the SMF record is prepared, but before it is written out.
4	Called before exit 2 for add, update, and retrieve processing.
5	At job initialization, either before the first action is executed (batch), or before the first Endevor panel is displayed (foreground).
6	Immediately before Endevor terminates processing.
7	Before and after each package function. For more information, see Chapter 5, "Package Exits."

The sections that follow describe each exit point in detail.

2.2 Exit Point 1: After Endevor Security Checks

2.2.1 Overview

Endevor calls exit 1 programs:

- When building a menu panel, after making a separate security check to verify access to each menu option.
- Before performing an action, after checking the Access Security Table, User Security Table, and Resource Security Table to determine whether the user is authorized to perform the action.

2.2.2 How to Use Exit 1 Programs

Use exit 1 programs to:

- Validate the return code from the Endevor security check.
- Perform additional security checking. For example, you might write an exit to interface with another security package (CA-ACF2, CA-Top Secret, etc.).

2.2.3 Assembler Exit Control Blocks

Control block	Fields modifiable by exit programs
Exit Control Block ($ECBDS)	▪ ECBRTCD ▪ ECBMSGCD ▪ ECBMSG ▪ ECBUEXT
Environment Block ($ENVDS)	None

2.2.4 COBOL Exit Control Blocks

Control block	Fields modifiable by exit programs
Exit Control Block (EXIT-CONTROL-BLOCK)	▪ ECB-RETURN-CODE ▪ ECB-MESSAGE-CODE ▪ ECB-MESSAGE-TEXT ▪ ECB-EXIT-HOLD-FIELD

Control block	Fields modifiable by exit programs
Environment Block (SRC-ENVIRONMENT-BLOCK) (TGT-ENVIRONMENT-BLOCK)	None.

2.2.5 Exit Control Block Return Codes

Endevor passes the return code from its security check to exit 1, in the ECBRTCD for assembler and ECB-RETURN-CODE for COBOL, of the Exit Control block. The associated error message is ECBMSG in assembler and ECB-MESSAGE-TEXT for COBOL, on entry to the exit.

Return code	Means
0	When passed to exit: the user is authorized to perform the action. When passed back to Endevor: passed successfully.
4	The user is not defined to the Access Security Table or the User Security Table; however, the element is referenced in the Resource Security Table, and the requested action is valid.
8	When passed to exit: not passed to exit programs. When passed back to Endevor: failed. Abort the request.
12	The user is not authorized to access the system/subsystem named by the action.
16	The requested action cannot be issued against the requested element, based on the Resource Security Table.

If you are using the SMF Interface, Endevor writes an SMF Security Record out if an error is returned from the exit. For details about SMF recording, see "SMF Recording," in the *Administration Guide*.

2.3 Exit Point 2: Before Actions

2.3.1 Overview

Endevor calls exit 2 programs before executing the following actions, both in foreground and in batch: ADD, UPDATE, RETRIEVE, GENERATE, MOVE, DELETE, SIGNIN, TRANSFER, ARCHIVE, and RESTORE. The programs are called just before the action is executed, after the data for the action is validated but before the actual processing starts. Under Quick Edit, Exit 2 is invoked before to open the temporary dataset "userid.QERCVYnn.RECOVERY", before the element is edited or created.

2.3.2 How to Use Exit 2 Programs

Use exit 2 programs to:

- Cancel an action.
- Change some of the input to an action.
- To modify Endevor defaults automatically. For example, you might write an exit program to change the DELETE INPUT SOURCE option from **N** to **Y** whenever a user adds or updates an element.

2.3.3 Assembler Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block ($ECBDS)	▪ ECBRTCD ▪ ECBUEXT ▪ ECBMSGCD ▪ ECBMSG
Source Environment Block ($ENVDS)	None.
Target Environment Block ($ENVDS)	None.

Control block	Fields modifiable by exit programs
Request Information Block ($REQPDS)	■ REQLEN ■ REQVERS ■ REQID ■ REQCCID ■ REQCOMM ■ REQSFLAG ■ REQSISOF ■ REQSISOC ■ REQEXPND ■ REQOVRWR ■ REQRTCOD ■ REQELVER ■ REQCPYBK ■ REQDEL (ARCHIVE only) ■ REQMOVWH ■ REQUPDT ■ REQBYGEN ■ REQONLCMP ■ REQSYNC ■ REQIGGF ■ REQBYPD ■ REQOPGRP ■ REQSOUID ■ REQOSRCH ■ REQORSGN ■ REQOJUMP ■ REQOSGNI ■ REQSIZE
Element Master Information Block ($ELMDS) and/or File Block ($FILDS)	None.

2.3.4 ENVFCBA Field in Assembler

The Source $ENVDS block points to the source of the action. The Target $ENVDS block points to the target location. As shown in the following table, the ENVFCBA field in each $ENVDS block points to either an Element Master block ($ELMDS) or a File block ($FILDS), depending on the action requested.

For this Action	The Source $ENVDS Block Points To	And the Target $ENVDS Block Points To
Add	$FILDS	none
Update	$FILDS	$ELMDS
Retrieve	$ELMDS	$FILDS
Generate	$ELMDS	none
Move	$ELMDS	$ELMDS—when element is at target prior to move none—when element not at target prior to move
Delete	$ELMDS	none
Signin	$ELMDS	none
Transfer	$ELMDS	$ELMDS
Archive	$ELMDS	$FILDS
Restore	$FILDS	$ELMDS

2.3.5 COBOL Control Blocks

Control Block	Fields modifiable by exit programs
Exit Control Block (EXIT-CONTROL-BLOCK)	▪ ECB-RETURN-CODE ▪ ECB-EXIT-HOLD-FIELD ▪ ECB-MESSAGE-CODE ▪ ECB-MESSAGE-TEXT
Source Environment Block (SRC-ENVIRONMENT-BLOCK)	None
Target Environment Block (TGT-ENVIRONMENT-BLOCK)	None

Control Block	Fields modifiable by exit programs
Request Information Block (REQUEST-INFO-BLOCK)	REQ-CCID ■ REQ-COMMENT ■ REQ-REQUEST-FLAG ■ REQ-SISO-INDICATOR ■ REQ-COPY-INDICATOR ■ REQ-EXPINCL-INDICATOR ■ REQ-OVERWRITE-INDICATOR ■ REQ-ACTION-RC ■ REQ-NEW-VERSION ■ REQ-GEN-COPYBACK ■ REQ-DELETE-AFTER (ARCHIVE only) ■ REQ-MOVE-WITH-HISTORY ■ REQ-ADD-WITH-UPDATE ■ REQ-BYPASS-GEN-PROC ■ REQ-DEL-COMPS-ONLY ■ REQ-SYNCHRONIZE ■ REQ-IGNGEN-FAIL ■ REQ-BYPASS-DEL-PROC ■ REQ-PROCESSOR-GROUP ■ REQ-SIGNOUT-USRID ■ REQ-SEARCH-OPTION ■ REQ-RETAIN-SIGNOUT-OPT ■ REQ-JUMP-OPTION ■ REQ-SIGNIN-OPTION
Element Master Information Block (SRC-ELEMENT-MASTER-INFO-BLOCK) (TGT-ELEMENT-MASTER-INFO-BLOCK) and/or File Block (SRC-FILE-CONTROL-BLOCK) (TGT-FILE-CONTROL-BLOCK)	None

2.3.6 COBOL Source and Target Fields

The Source SRC-ENVIRONMENT-BLOCK block points to the source of the action. The Target TGT-ENVIRONMENT-BLOCK block points to the target location. As shown in the following table, each Environment block points to either an Element Master block or to a File block, depending on the action requested. file.Environment block

For this Action	The Source SRC-ENVIRONMENT-BLOCK Block points to	And the Target TGT-ENVIRONMENT-BLOCK Block points to
Add	SRC-FILE-CONTROL-BLOCK	none
Update	SRC-FILE-CONTROL-BLOCK	TGT-ELEMENT-MASTER -INFO-BLOCK
Retrieve	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-FILE-CONTROL -BLOCK
Generate	SRC-ELEMENT-MASTER-INFO-BLOCK	none
Move	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-ELEMENT-MASTER-INFO—when element is at target prior to move none—when element not at target prior to move
Delete	SRC-ELEMENT-MASTER-INFO-BLOCK	none
Signin	SRC-ELEMENT-MASTER-INFO-BLOCK	none
Transfer	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-ELEMENT-MASTER -INFO-BLOCK
Archive	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-FILE-CONTROL -BLOCK
Restore	SRC-FILE-CONTROL-BLOCK	TGT-ELEMENT-MASTER -INFO-BLOCK

2.3.7 Return Codes

You signal the results of the exit using the return code field ECBRTCD in assembler or ECB-RETURN-CODE in COBOL. Place the appropriate code in the field of the Exit Control block, before returning to Endevor.

Return code	Meaning
0	Proceed with the action. The exit did not change any action-related data.
4	Proceed with the action, but the exit changed fields in the Request Information block, and the revised fields should be used for action processing.
8	Abort the action request.

2.4 Exit Point 3: After Actions

2.4.1 Overview

Endevor calls exit 3 programs after executing the following actions, both in foreground and in batch: ADD, UPDATE, RETRIEVE, GENERATE, MOVE, DELETE, SIGNIN, TRANSFER, ARCHIVE, and RESTORE. The programs are called after each action is completed, including execution of any processors, and the master control file base and/or delta libraries have been updated.

If SMF recording is in effect, Endevor passes the SMF Action Request record to the exit programs. It is passed as the second parameter in the parameter list.

2.4.2 How to Use Exit 3 Programs

Use exit 3 programs to:

- Perform general logging and recording.
- Modify the SMF Action Request record.

2.4.3 Assembler Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block ($ECBDS)	▪ ECBRTCD ▪ ECBSMFWR ▪ ECBMSGCD ▪ ECBMSG ▪ ECBUEXT.
Request Information Block ($REQPDS)	None.
SMF Action Record	All.
Source Environment Block ($ENVDS)	None.
Target Environment Block ($ENVDS)	None.
Element Master Information Block ($ELMDS) and/or File Block ($FILDS)	None.

2.4.4 ENVFCBA Field

The Source $ENVDS block points to the source of the action. The Target $ENVDS block points to the target location. As shown in the following table, the ENVFCBA field in each $ENVDS block points to either an Element Master block ($ELMDS) or a File block ($FILDS), depending on the action requested.

For this action	The source $ENVDS block points to	And the target $ENVDS block points to
Add	$FILDS	$ELMDS
Update	$FILDS	$ELMDS
Retrieve	$ELMDS	$FILDS
Generate	$ELMDS	none
Move	$ELMDS (see note below)	$ELMDS
Delete	$ELMDS	none
Signin	$ELMDS	none
Transfer	$ELMDS (see note below)	$ELMDS
Archive	$ELMDS (see note below)	$FILDS
Restore	$FILDS	$ELMDS

Note: This applies to Move, Transfer, and Archive if the element has not been deleted by the action.

2.4.5 COBOL Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block (EXIT-CONTROL-BLOCK)	▪ ECB-RETURN-CODE ▪ ECB-SMF-WRITE-FLAG ▪ ECB-MESSAGE-CODE ▪ ECB-MESSAGE-TEXT ▪ ECB-EXIT-HOLD-FIELD
Request Information Block (REQUEST-INFO-BLOCK)	None.
SMF Action Record	All.

Control block	Fields modifiable by exit programs
Source Environment Block (SRC-ENVIRONMENT-BLOCK)	None.
Target Environment Block (TGT-ENVIRONMENT-BLOCK)	None.
Element Master Information Block (SRC-ELEMENT-MASTER-INFO-BLOCK) (TGT-ELEMENT-MASTER-INFO-BLOCK) and/or File Block (SRC-FILE-CONTROL-BLOCK) (TGT-FILE-CONTROL-BLOCK)	None.

2.4.6 COBOL Source and Target Fields

The Source SRC-ENVIRONMENT-BLOCK block points to the source of the action. The Target TGT-ENVIRONMENT-BLOCK block points to the target location. As shown in the following table, each Environment block points to either an Element Master block or to a File block, depending on the action requested.

For this Action	The Source SRC-ENVIRONMENT-BLOCK Block points to	And the Target TGT-ENVIRONMENT-BLOCK Block points to
Add	SRC-FILE-CONTROL-BLOCK	TGT-ELEMENT-MASTER-INFO-BLOCK
Update	SRC-FILE-CONTROL-BLOCK	TGT-ELEMENT-MASTER-INFO-BLOCK
Retrieve	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-FILE-CONTROL -BLOCK
Generate	SRC-ELEMENT-MASTER-INFO-BLOCK	none
Move	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-ELEMENT-MASTER-INFO-BLOCK
Delete	SRC-ELEMENT-MASTER-INFO-BLOCK	none
Signin	SRC-ELEMENT-MASTER-INFO-BLOCK	none
Transfer	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-ELEMENT-MASTER -INFO-BLOCK
Archive	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-FILE-CONTROL -BLOCK
Restore	SRC-FILE-CONTROL-BLOCK	TGT-ELEMENT-MASTER -INFO-BLOCK

Note: This applies if the element has not been deleted by the action.

2.4.7 Return Codes

Valid return codes for this exit are listed below. Place the appropriate code in the ECBRTCD field for assembler and the ECB-RETURN-CODE field for COBOL in the Exit Control block, before returning to Endevor.

Return code	Meaning
0	Successful completion.
4	Successful completion, but check the ECBSMFWR field to determine whether to write out the SMF record.

2.5 Exit Point 4: Data Set Validation

2.5.1 Overview

Endevor calls exit 4 programs before calling exit 2 programs for the following actions: ADD, UPDATE, and RETRIEVE. The exit program passes the File block back to Endevor. The File block describes the external file used during action processing.

Note: This exit is not called for ARCHIVE and RESTORE requests, even though these actions use the File block. This is because if you change the data set defined to the File block, you have to allocate the external file dynamically, which is cumbersome with tape (archive) files.

2.5.2 How to Use Exit 4 Programs

Use exit 4 programs to:

- Change external files, for example, from VSAM to PDS format.
- Change the DDname (FILDDN) used for the action. If you change the DDname, you must dynamically allocate the file identified by the new DDname field, and pass a return code of 4. (At the time the exit is called, Endevor has already allocated the data set for the DDname passed to the exit.)
- Set up controls so users can only retrieve elements into certain data sets, or add elements from certain data sets.

  Note: If you use exit 4 and change the name of the dataset being used as input, you should be aware of the following:

  - Dataset validation will still be based on the file used by exit 4 and not source DSN in the SCL.
  - The file exit 4 points to will be recorded in the MCF as the source input file.
  - If the option, To Delete Input Source, is specified, the member will be deleted from the dataset that exit 4 points to.

2.5.3 Assembler Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block ($ECBDS)	■ ECBRTCD ■ ECBMSGCD ■ ECBMSG ■ ECBUEXT
Request Information Block ($REQPDS)	None.

Control block	Fields modifiable by exit programs
Source Environment Block ($ENVDS)	None.
Target Environment Block ($ENVDS)	None.
File Block ($FILDS)	■ FILDSN (ignored) ■ FILDDN ■ FILDSMEM (ignored) ■ FILDSTY (ignored)

2.5.4 ENVFCBA Assembler Field

The Source $ENVDS block points to the source of the action. The Target $ENVDS block points to the target location. As shown in the following table, the ENVFCBA field in each $ENVDS block points to either an Element Master block ($ELMDS) or a File block ($FILDS), depending on the action requested.

For this action	The source $ENVDS block points to	And the target $ENVDS block points to
Add	$FILDS	$ELMDS
Update	$FILDS	$ELMDS
Retrieve	$ELMDS	$FILDS

2.5.5 COBOL Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block (EXIT-CONTROL-BLOCK)	■ ECB-RETURN-CODE ■ ECB-MESSAGE-CODE ■ ECB-MESSAGE-TEXT ■ ECB-EXIT-HOLD-FIELD
Request Information Block (REQUEST-INFO-BLOCK)	None.
Source Environment Block (SRC-ENVIRONMENT-BLOCK)	None.
Target Environment Block (TGT-ENVIRONMENT-BLOCK)	None.

Control block	Fields modifiable by exit programs
File Block (SRC-FILE-CONTROL-BLOCK) (TGT-FILE-CONTROL-BLOCK)	■ SRC-FIL-DATASET-NAME (ignored) ■ TGT-FIL-DATASET-NAME(ignored) ■ SRC-FIL-DDNAME ■ TGT-FIL-DDNAME ■ SRC-FIL-DATASET-MEMBER (ignored) ■ TGT-FIL-DATASET-MEMBER (ignored) ■ SRC-FIL-DATASET-TYPE (ignored) ■ TGT-FIL-DATASET-TYPE (ignored)

2.5.6 COBOL Source and Target Fields

The Source SRC-ENVIRONMENT-BLOCK block points to the source of the action. The Target TGT-ENVIRONMENT-BLOCK block points to the target location. As shown in the following table, each Environment block points to either an Element Master block or a File block depending on the action requested.

For this action	The source SRC-ENVIRONMENT-BLOCK Block points to	And the target TGT-ENVIRONMENT-BLOCK Block points To
Add	SRC-FILE-CONTROL-BLOCK	TGT-ELEMENT- MASTER-INFO-BLOCK
Update	SRC-FILE-CONTROL-BLOCK	TGT-ELEMENT- MASTER-INFO-BLOCK
Retrieve	SRC-ELEMENT-MASTER-INFO-BLOCK	TGT-FILE-CONTROL -BLOCK

2.5.7 Return Codes

Valid return codes for this exit are listed below. Place the appropriate code in the ECBRTCD field for assembler or ECB-RETURN-CODE for COBOL in the Exit Control Block, before returning to Endevor.

Return code	Meaning
0	Proceed with the action.
4	Proceed with the action, but the exit changed fields in the Request Information block and the revised fields should be used for action processing. If you change the data set used, *make sure to* allocate the file before returning a code 4.
8	Abort the action request.

2.6 Exit Point 5: At Session Initialization

2.6.1 Overview

ENDEVOR calls exit 5 programs at session initialization, immediately after Endevor allocates its internal control blocks. In batch, it is called before the first action is executed. In foreground, it is called before the first Endevor panel is displayed.

2.6.2 How to Use Exit 5 Programs

Use exit 5 programs to:

- Change the User ID.
- Make calls to SMF records.
- Create a log of activity against Endevor.

2.6.3 Assembler Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block ($ECBDS)	▪ ECBRTCD ▪ ECBUSER ▪ ECBMSGCD ▪ ECBMSG ▪ ECBUEXT

2.6.4 COBOL Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block (EXIT-CONTROL-BLOCK)	■ ECB-RETURN-CODE ■ ECB-USER-ID ■ ECB-MESSAGE-CODE ■ ECB-MESSAGE-TEXT ■ ECB-EXIT-HOLD-FIELD

2.6.5 Return Codes

Valid return codes for this exit are listed below. Place the appropriate code in the ECBRTCD field for assembler or the ECB-RETURN-CODE for COBOL in the Exit Control block, before returning to Endevor. If you change the User ID, *make sure* to pass a return code of 4.

Return code	Meaning
0	Successful completion. The exit did not change the User ID.
4	Proceed with the action, but the exit changed the USER ID field in the Exit Control block, and the revised ID should be used for subsequent processing.

2.7 Exit Point 6: Before Session Termination

2.7.1 Overview

Endevor calls exit 6 programs before it ends a session. In batch, these programs are called just before the job terminates. In foreground, they are called just before Endevor returns to ISPF.

2.7.2 How to Use Exit 6 Programs

Use exit 6 programs to:

- Collect statistics.
- Complete logging activity started by an exit 5 module.

2.7.3 Assembler Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block ($ECBDS)	■ ECBRTCD ■ ECBMSGCD ■ ECBMSG ■ ECBUEXT

2.7.4 COBOL Control Blocks Passed

Control block	Fields modifiable by exit programs
Exit Control Block EXIT-CONTROL-BLOCK	■ ECB-RETURN-CODE ■ ECB-MESSAGE-CODE ■ ECB-MESSAGE-TEXT ■ ECB-EXIT-HOLD-FIELD

2.7.5 Return Codes

Valid return codes for this exit are listed below. Place the appropriate code in the ECBRTCD field in assembler or the ECB-RETURN-CODE for COBOL in the Exit Control block, before returning to Endevor.

Return code	Meaning
0	Successful completion.

2.8 Example of Exit Use

2.8.1 When Endevor Calls Exit Programs

The following example illustrates where Endevor calls exit programs, and suggests exits that a user might write for each point.

Step	Action
1	You select Endevor on ISPF Main Menu. Endevor reads the Defaults Table, the C1UEXITS table, loads exit programs, then calls exit 5 programs. You might write a program for exit point 5 that logs Endevor activity.
2	You select an environment on the Environment Options Menu. Endevor checks your authority to select the chosen environment, then calls exit 1 programs. You might write a program for exit point 1 that also checks your security profile in CA-ACF2.
3	You submit an ADD request. Endevor calls exit 4 programs to make sure the element to be added is in PDS format, then calls exit 2 programs. You might write an exit 2 program that: ■ Changes the DELETE INPUT SOURCE option from **N** to **Y** whenever a user adds or updates an element. ■ Sends a message to the owner of a module that is signed out, informing the owner that you want to add the module.
4	Endevor adds the element, builds the SMF record, and updates the Master Control File, base and delta libraries, then calls exit 3 programs. You might write an exit 3 program that: ■ Updates the SMF record prior to writing it. ■ Sends a message to the production control supervisor that the element has been added.
5	You log off Endevor. Endevor calls exit 6 programs. You might write an exit 6 program that completes the log of Endevor activity initiated at the start of the session.

2.9 User Exit Message Handling

2.9.1 Coding Exit Programs to Fill in Fields

To issue messages from your exit programs you must code the exit programs to fill in the following fields:

2.9.1.1 Assembler Coding

Exits	Control block	Return code	Message code (optional)	Message text (batch only)
1-6	$ECBDS	ECBRTCD	ECBMSGCD	ECBMSG
7	$PECBDS	PECBRTCD	PECBMGCD	PECBMSG

2.9.1.2 COBOL Coding

Exits	Control block	Return code	Message code (optional)	Message text (batch only)
1-6	EXIT-CONTROL-BLOCK	ECB-RETURN-CODE	ECB-MESSAGE-CODE	ECB-MESSAGE-TEXT
7	PACKAGE-EXIT-BLOCK	PECB-NDVR-EXIT-RC	PECB-MESSAGE-ID	PECB-MESSAGE

The message code default is 0000. If you want the message to have an ISPF equivalent, the message code format should be 0nnn.

An example of a return code is 12.

An example of a message code is 0801.

The following message codes are reserved:

- 0001
- 09nn

2.9.2 Building Messages

In foreground, Endevor builds a message identifier by combining the message code and return code (0801E), dropping the first digit of the message code (801E), then adding CIUU to the result: CIUU801E.

In batch, Endevor builds the message itself starting with CIU, adding the message code and severity code (CIU0801E), adding the string 'exit # - ' (CIU0801E EXIT 2 -), then adding the message text. For example:

```
CIU0801E EXIT 2 - ACTION DENIED. USER IS NOT AUTHORIZED TO PERFORM THIS ACTION
```

2.9.3 Issuing Messages

In foreground, Endevor passes the message identifier (CIUU801E) to ISPF, which searches for that string in member CIUUnn in the installed message library, iprfx.iqual.ISPMLIB, where nn are the second and third digits of the message code.

You must create this member in order to issue ISPF messages. In this example, you would create member CIUU80, which can contain multiple messages. Each message must take up two lines, so member CIUU80 might look like this:

```
CIUU801I 'ACTION ALLOWED'                                .ALARM=YES
'USER IS AUTHORIZED TO PERFORM THIS ACTION            '
CIUU801E 'ACTION DENIED '                                .ALARM=YES
'USER IS NOT AUTHORIZED TO PERFORM THIS ACTION        '
CIUU802E 'INVALID PKID'                                  .ALARM=YES
'THE SPECIFIED PACKAGE ID IS INVALID FOR YOUR USE     '
```

Line 1 contains a message identifier and short message text of up to 25 characters.

Line 2 contains long message text. This text can be up to 80 characters.

For more information about building ISPF messages, see the *ISPF Dialog Management Guide*.

When ISPF finds the string CIUU801E, it displays the short message text, in this case ACTION DENIED.

In batch, Endevor writes the formatted message to the Batch Execution Report.

Chapter 3. Exit Control Blocks

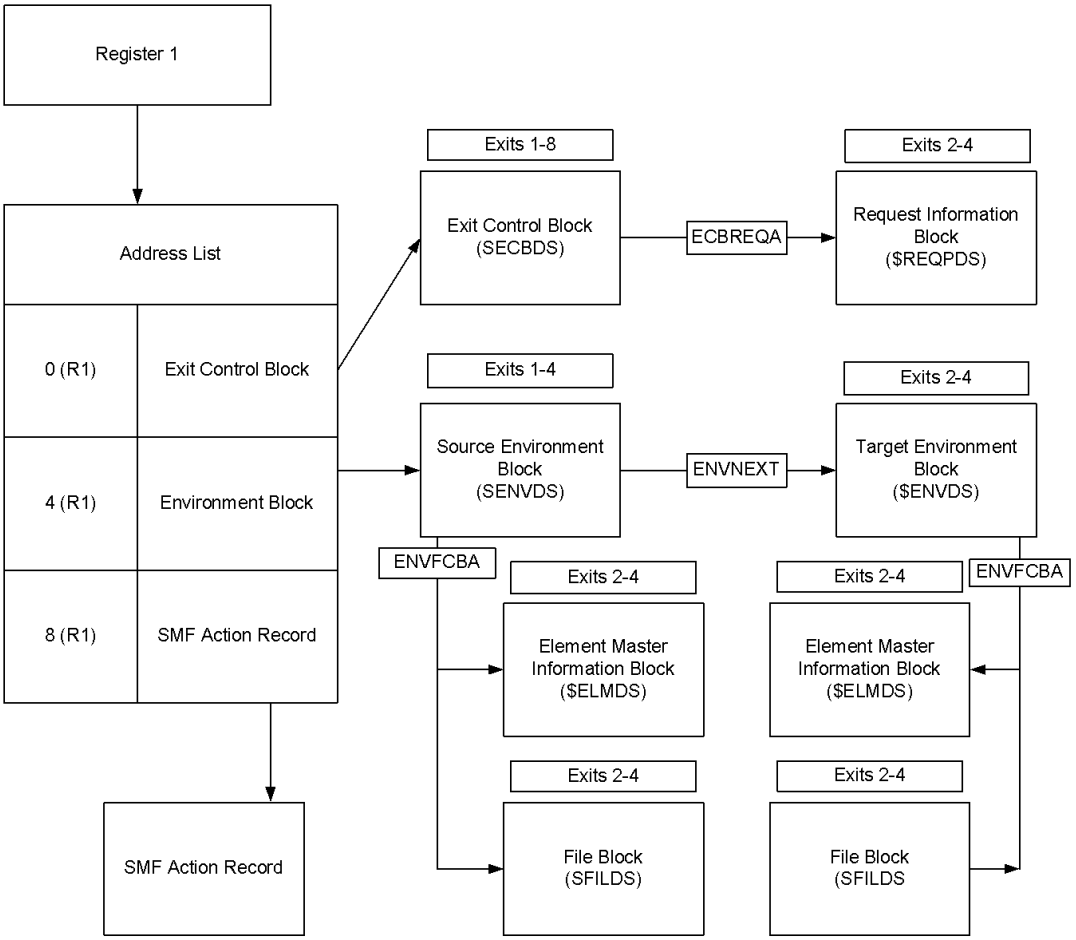
This chapter describes the exit control blocks. It contains the following sections:

- Overview
- $ECBDS DSECT: Exit Control Block
- EXIT-CONTROL-BLOCK control block layout: Exit Control Block
- $REQPDS DSECT: Request Information Block
- REQUEST-INFO-BLOCK control block layout: Request Information Block
- $ENVDS DSECT: Environment Block
- XXX-ENVIRONMENT-BLOCK control block layout: Environment Block
- $ELMDS DSECT: Element Master Information Block
- XXX-ELEMENT-MASTER-INFO-BLOCK control block layout: Element Master Information Block
- $FILDS DSECT: File Block
- XXX-FILE-CONTROL-BLOCK control block layout: File Block
- SMF Action Record

3.1 Overview

When a user exit is invoked, Endevor formats a series of control

address list, which in turn points to the control blocks



This section describes each of the control blocks in this structure. In all cases where specific fields do not apply for the processing at hand, alphabetic fields are space-filled and numeric fields are zero-filled.

The control blocks are always passed to the exit program in the following order:

Sequence	Control block name
1	Exit Control block
2	Request Information block
3	Source Environment block

Sequence	Control block name
4	Source Element Master Information block
5	Source File block
6	Target Environment block
7	Target Element Master Information block
8	Target File block
9	SMF Action Record

Note: If the control block is not relevant to the user exit, then the control block will not contain valid data.

3.2 Primary Information Block

The following sections contain sample control blocks and field field descriptions for Assembler and COBOL.

3.2.1 $ECBDS DSECT Sample

This is the first and primary information block passed to exit programs. It is pointed to by Register 1: 0(R1), and is passed using standard calling conventions.

```
$ECBDS
$ECBDS   DSECT
ECBLEN   DC    AL2(ECBSIZE)         LENGTH OF THE BLOCK
ECBVERS  DC    AL2(ECB$VERS)        CURRENT VERSION OF THE BLOCK
ECB$VERS EQU   2                        VERSION 2
ECBID    DC    CL4'$ECB'            BLOCK ID
ECBFLAG  DC    F'0'                 ANY SPECIAL FLAGS
ECBRTCD  DC    F'0'                 RETURN CODE FROM EXIT
ECB$OK   EQU   0                    RETURN 0 ALL O.K.
ECB$MOK  EQU   4                    RETURN 4 ALL O.K. WITH MODS
ECB$FAIL EQU   8                    RETURN 8 ABORT
ECBARGS  DC    F'0'                 NUMBER OF PARMS PASSED (INCLUSIVE)
ECBEXTN  DC    F'0'                 EXIT NUMBER BEING INVOKED
ECB$CON1 EQU   1                    C1UEXT01 - SECURITY
ECB$CON2 EQU   2                    C1UEXT02 - PRE-ACTION
ECB$CON3 EQU   3                    C1UEXT03 - POST-ACTION
ECB$CON4 EQU   4                    C1UEXT04 - EXT DS VALIDATION
ECB$CON5 EQU   5                    C1UEXT05 - JOB INITIATION
ECB$CON6 EQU   6                    C1UEXT06 - JOB TERMINATION
ECB$MEXT EQU   6                      MAX EXIT NUMBER DEFINED
ECBUSER  DC    CL8' '               TSO USER/BATCH JOBNAME
ECBMODE  DC    CL1' '               MODE FOR ONLINE(TSO) OR BATCH
ECB$TSO  EQU   C'T'                 TSO MODE
EXB$BAT  EQU   C'B'                 BATCH MODE
ECBINTNL DC    CL1' '               INTERNAL CALL FLAG ( PANEL BUILD)
ECB$YES  EQU   C'Y'                 YES - INTERNAL ONLY CALL
ECB$NO   EQU   C'N'                 NO  - ACTUAL ACTION REQUEST
ECBSMFAC DC    CL1' '               SMF OUTPUT ENABLED ( Y OR N)
ECBSMFWR DC    CL1' '               SMF WRITE BUFFER ( Y OR N)
ECBFUNC  DC    F'0'                 CURRENT ACTIVITY
ECB$ADD  EQU    1                   ADD
ECB$UPD  EQU    2                   UPDATE
ECB$RETR EQU    3                   RETRIEVE
ECB$DEL  EQU    4                   DELETE
ECB$GEN  EQU    5                   GENERATE
ECB$DISP EQU    6                   DISPLAY
ECB$MOVE EQU    7                   MOVE
ECB$ARCH EQU    8                   ARCHIVE
ECB$SUPV EQU    9                   **-RESERVED-**
ECB$PANL EQU    10                  **-RESERVED-**
ECB$CHGT EQU    11                  CHGTYPE
ECB$SINI EQU    12                  SIGNIN
ECB$SINO EQU    13                  SIGNOVR
ECB$ENVM EQU    14                  ENVRNMGR
ECB$PRNT EQU    15                  PRINT
ECB$TRNS EQU    16                  TRANSFER
```

```
ECB$NUSE EQU    17                  **-NOT-YET-USED-**
ECB$REST EQU    18                  RESTORE
ECB$FMAX EQU    18                  MAX-ACTION-VALUE
ECBFUNAM DC    CL8' '               ACTION NAME
ECBMSGCD DC    CL4' '               MESSAGE ID FOR ERROR MESSAGE
                                    MESSAGE CODE WILL BE C1XNNNNS
ECBMSGLN DC    H'0'                 LENGTH OF ERROR MESSAGE
ECBMSGSZ  EQU   132                 SIZE IS 132
ECBMSG    DC    CL(ECBMSGSZ)' '     MESSAGE
ECBCALR   DC    CL1' '              CALLER'S ORIGIN
ECB$NDV   EQU   C'E'                ENDEVOR
ECB$QE    EQU   C'Q'                QUICK-EDIT
ECBPKG    DC    CL1' '              RUNNING UNDER A PKG? Y/N
ECBREQA   DC    F'0'                ADDRESS OF $REQDS
ECBPNLT   DC    F'0'                PANEL TYPE IF INTERNAL
ECB$NONE  EQU   0                   NO PANEL INFORMATION
ECB$PENV  EQU   1                   ENVIRONMENT LEVEL PANEL

ECB$PPRM  EQU   2                   TOP/MAIN    LEVEL PANEL
ECB$PFOR  EQU   3                   FOREGROUND  LEVEL PANEL
ECBUEXT   DC    F'0'                USER EXIT HOLD FIELD
ECBIFUNC  DC    F'0'                INTERNAL SECURITY FUNCTIONS
                                    SEE EQUATES FOLLOWING ECBFUNC
ECBHIRC   DC    F'0'                HIGHEST RC SET BY PREVIOUS EXITS
ECBSIZE   EQU   *-$ECBDS            LENGTH OF $ECBDS BLOCK
```

3.2.2 Assembler Field Descriptions

The following list contains descriptions of the fields in the $ECBDS DSECT. Unless otherwise noted, exit programs cannot modify these fields.

Field	Description
ECBLEN	Size of the Exit Control block ($ECBDS), in bytes.
ECBVERS	Version number to identify the Exit Control block ($ECBDS) written by the exit handler. This should always be 1. Use the ECB$VERS label to verify that the correct version of this DSECT was passed to your exit.
ECBID	Code that identifies this as the Exit Control block, for use in rapid identification when you are scanning your modules visually. Value is always "$ECB".
ECBFLAG	Not used.

Field	Description
ECBRTCD	Modifiable by exits 1-6. This field contains the return code. On entry to the exit, this field is generally set to 0. An exception occurs with exit 1, if Endevor has detected a security violation in its own checking (in which case this is passed as 4, 12, or 16, as described for exit 1). On return from the exit, this field has meaning as follows. (Any value other than those listed below is set to 8.) Refer to the individual write-up of each exit for further detail. Values are: Value Label and Meaning ■ Value: 0 – Label: ECB$OK. – Meaning: Successful (or continue processing). Applies for all exits. ■ Value: 4 – Label: ECB$MOK. – Meaning: Successful (continue processing), but some control block data was modified by the exit. Applies for exits 1-5. Make sure to modify the changed data if necessary before proceeding. ■ Value: 8 – Label: ECB$FAIL. – Meaning: Stop action processing for exits 1, 2, and 4. ■ Value: 12 – Label: none – Meaning: Used only on entry to an exit 1 program. ■ Value: 16 – Label: none – Meaning: Used only on entry to an exit 1 program.
ECBARGS	Number of parameters passed to the exit in the address list (R1). The last word of the list is X'80', indicating end-of-list.

Field	Description
ECBEXTN	Number identifying the exit being invoked: ■ Value: 1 – Label: ECB$CON1 – Meaning: Exit 1 invoked. ■ Value: 2 – Label: ECB$CON2. – Meaning: Exit 2 invoked ■ Value: 3 – Label: ECB$CON3 – Meaning: Exit 3 invoked ■ Value: 4 – Label: ECB$CON4 – Meaning: Exit 4 invoked ■ Value: 5 – Label: ECB$CON5 – Meaning: Exit 5 invoked ■ Value: 6 – Label: ECB$CON6. – Meaning: Exit 6 invoked
ECBUSER	User ID associated with the current processing. Modifiable by exit 5.
ECBMODE	Indicates whether the current action was requested in foreground or batch. ■ Value: T – Label: TSO – Meaning: Action was requested in foreground. ■ Value: B – Label: ECB$BAT – Meaning: Action was requested in batch.

Field	Description
ECBINTNL	Indicates whether the current processing relates to building a panel dynamically or to an action request. ■ Value: Y – Label: ECB$YES – Meaning: Processing involves dynamic TSO panel building ■ Value: N – Label: ECB$NO – Meaning: Processing involves an action request
ECBSMFAC	Indicates whether Endevor was installed to support SMF recording, as defined by the SMFREC# parameter of the Defaults Table: **Y** (yes) or **N** (no).
ECBSMFWR	Applicable if ECBSMFAC is **Y**. Indicates whether Endevor is set up to write out SMF action and/or security records, as defined by the SMFACT and SMFSEC parameters of the Defaults Table: **Y** (yes) or **N** (no). You can suppress writing SMF records by setting this field to **N** using exit 3.

Field	Description
ECBFUNC	Action code. Value Action 1 -- Add 2 -- Update 3 -- Retrieve 4 -- Delete 5 -- Generate 6 -- Display 7 -- Move 8 -- Archive 9 -- Reserved 10 -- Reserved 11 -- CHGTYPE 12 -- Signin 13 -- Override Signout 14 -- ENVRNMGR 15 -- Print 16 -- Transfer 17 -- Not used 18 -- Restore
ECBFUNAM	Literal to describe the current action (ADD, UPDATE, and so forth). Passed to exits 1-4.
ECBMSGCD	Modifiable by exits 1-6. Default value is 0000. Used to construct the message code written to the Endevor Execution Report by the exit handler. A message code precedes each message written to the report, and applies if the ECBMSG field contains a non-blank value. For messages produced by user exits, the error code is formatted as C1Unnnns, where nnnn is defined by this field, and s is a severity code associated with ECBRTCD (**I** for return codes of less than 8, **E** for return codes of 8 and above).
ECBMSGLN	Not used.
ECBMSGSZ	Not used.

Field	Description
ECBMSG	Modifiable by exits 1-6. Error message associated with the current activity. For exit 1, this is set on entry if a security violation has occurred. For the other exits, this is blank on entry. 103 characters in this field are available for messages. The first 29 positions are for Endevor-generated information. If there is a message in this field when you return from the user exit, the exit handler records the message in the Endevor Execution Report. The message is preceded in that report with an error code (C1Unnnns format, described above for the ECBMSGCD field).
ECBCALR	Exit is executing under Endevor or Quick-Edit. ■ Value: E – Label: ECB$NDV – Meaning: Exit is running under Endevor ■ Value: Q – Label: ECB$QE – Meaning: Exit is running under Quick-Edit ■ Value: Y – Label: ECBPKG – Meaning: Exit is running under a package
ECBREQA	A pointer to the Request Information block (DSECT $REQPDS).

Field	Description
ECBPNLT	Indicates the type of panel being processed when the exit was invoked. ■ Value: 0 – Label: ECB$NONE – Meaning: No panel information is available ■ Value: 1 – Label: ECB$PENV – Meaning: Exit was called while processing an Environment Definition panel ■ Value: 2 – Label: ECB$PPRM – Meaning: Exit was called while processing the Primary Options Menu ■ Value: 3 – Label: ECB$PFOR – Meaning: Exit was called while processing a foreground panel
ECBUEXT	Modifiable by exits 1-6. User field, used to save (and pass) information between user exits. Generally, this field contains an address that points to a block of user data. When you are using anchor IDs, this field is used to store address values read by exit programs from the anchor ID table. Any value placed in this field remains there until the end of the Endevor session, or until another user exit loads a new value in this field.
ECBIFUNC	Same as ECBFUNC (above).
ECBHIRC	Highest return code set by previously executed system and user exits for this exit number.
ECBSIZE	Size of the $ECBDS DSECT.

3.2.3 EXIT-CONTROL-BLOCK control block layout sample

This is the first and primary information block passed to exit programs. It is pointed to by Register 1: 0(R1), and is passed using standard calling conventions.

```
01  EXIT-CONTROL-BLOCK.                                          00010000


    05  ECB-LENGTH              PIC 9(4) COMP SYNC.              00020000
    05  ECB-CURRENT-VERSION     PIC 9(4) COMP SYNC.              00030000
    05  ECB-BLOCK-ID            PIC X(4).                        00040007
    05  ECB-FLAG                PIC 9(8) COMP SYNC.              00050005
```

```
05  ECB-RETURN-CODE         PIC 9(8) COMP SYNC.          00060005
    88  RETURN-OK           VALUE 0.                     00070005
    88  RETURN-OK-MODS      VALUE 4.                     00080005
    88  RETURN-ABORT        VALUE 8.                     00090005
05  ECB-ARGUMENTS           PIC 9(8) COMP SYNC.          00100005
05  ECB-EXIT-NUMBER         PIC 9(8) COMP SYNC.          00110005
    88  C1IEXT01            VALUE 1.                     00120005
    88  C1IEXT02            VALUE 2.                     00130005
    88  C1IEXT03            VALUE 3.                     00140005
    88  C1IEXT04            VALUE 4.                     00150005
    88  C1IEXT05            VALUE 5.                     00160005
    88  C1IEXT06            VALUE 6.                     00170005
    88  HIGH-EXIT-NUMBER    VALUE 6.                     00170005
05  ECB-USER-ID             PIC X(8).                    00180005
05  ECB-TSO-BATCH-MODE      PIC X.                       00190005
    88  TSO                 VALUE 'T'.                   00200005
    88  BATCH               VALUE 'B'.                   00210005
05  ECB-CALL-FLAG           PIC X.                       00220005
    88  INTERNAL-CALL       VALUE 'Y'.                   00230005
    88  ACTION-CALL         VALUE 'N'.                   00240005
05  ECB-SMF-FLAG            PIC X.                       00250005
    88  SMF-ENABLED         VALUE 'Y'.                   00260005
    88  SMF-DISABLED        VALUE 'N'.                   00270005
05  ECB-SMF-WRITE-FLAG      PIC X.                       00280005
    88  SMF-WRITE-BUFFER    VALUE 'Y'.                   00290005
    88  SMF-NO-WRITE-BUF    VALUE 'N'.                   00300005
05  ECB-ACTION-CODE         PIC 9(8) COMP SYNC.          00310005
    88  ADD-ACTION          VALUE 1.                     00320005
    88  UPDATE-ACTION       VALUE 2.                     00330005
    88  RETRIEVE-ACTION     VALUE 3.                     00340005
    88  DELETE-ACTION       VALUE 4.                     00350005
    88  GENERATE-ACTION     VALUE 5.                     00360005
    88  DISPLAY-ACTION      VALUE 6.                     00370005
    88  MOVE-ACTION         VALUE 7.                     00380005
    88  ARCHIVE-ACTION      VALUE 8.                     00390005
    88  CHGTYPE-ACTION      VALUE 11.                    00400000
    88  SIGNIN-ACTION       VALUE 12.                    00400000
    88  SO-OVERRIDE-ACTION  VALUE 13.                    00401000
    88  ENV-MGR-ACTION      VALUE 14.                    00410005
    88  PRINT-ACTION        VALUE 15.                    00420009
    88  TRANSFER-ACTION     VALUE 16.                    00430009
    88  RESTORE-ACTION      VALUE 18.                    00440005
05  ECB-ACTION-NAME         PIC X(8).                    00450005
05  ECB-MESSAGE-CODE        PIC X(4).                    00460005
05  ECB-MESSAGE-LENGTH      PIC 9(4) COMP SYNC.          00470005
05  ECB-MESSAGE-TEXT        PIC X(132).                  00480005
05  ECB-CALLER-ORIGIN       PIC X(1).
    88  CALLER-ENDEVOR      VALUE 'E'.
    88  CALLER-QUICKEDIT    VALUE 'Q'.
05  ECB-UNDER-PACKAGE       PIC X(1).
    88  PACKAGE-YES         VALUE 'Y'.
    88  PACKAGE-NO          VALUE 'N'.
05  ECB-REQUEST-POINTER     PIC 9(8) COMP SYNC.          00490005
05  ECB-PANEL-TYPE          PIC 9(8) COMP SYNC.          00500005
    88  NO-PANEL-INFO       VALUE 0.                     00510005
    88  ENVIRONMENT-PANEL   VALUE 1.                     00520005
    88  MAIN-PANEL          VALUE 2.                     00530005
    88  FOREGROUND-PANEL    VALUE 3.                     00540005
05  ECB-EXIT-HOLD-FIELD     PIC 9(8) COMP SYNC.          00550005
```

```
05  ECB-SECURITY-FUNCTION   PIC 9(8) COMP SYNC.             00560005
05  ECB-HIGH-RC             PIC 9(8) COMP SYNC.             00560005
```

3.2.4 COBOL Field Descriptions

The following list contains descriptions of the fields in the EXIT-CONTROL-BLOCK. Unless otherwise noted, exit programs cannot modify these fields.

Field	Description
ECB-LENGTH	Size of the Exit Control block (EXIT-CONTROL-BLOCKS), in bytes.
ECB-CURRENT-VERSION	Version number to identify the Exit Control block (EXIT-CONTROL-BLOCKS) written by the exit handler. This should always be 1. Use the (ECB-CURRENT-VERSION) label to verify that the correct version of this DSECT was passed to your exit.
ECB-BLOCK-ID	Code that identifies this as the Exit Control block, for use in rapid identification when you are scanning your modules visually. Value is always "$ECB".
ECB-FLAG	Not used.

Field	Description
ECB-RETURN-CODE	Modifiable by exits 1-6. This field contains the return code. On entry to the exit, this field is generally set to 0. An exception occurs with exit 1, if Endevor has detected a security violation in its own checking (in which case this is passed as 4, 12, or 16, as described for exit 1). On return from the exit, this field has meaning as follows. (Any value other than those listed below is set to 8.) Refer to the individual write-up of each exit for further detail. ■ Value: 0 – Label: RETURN-OK – Meaning: Successful (or continue processing). Applies for all exits ■ Value: 4 – Label: RETURN-OK-MODS – Meaning: Successful (continue processing), but some control block data was modified by the exit. Applies for exits 1-5. Make sure to modify the changed data if necessary before proceeding. ■ Value: 8 – Label: RETURN-ABORT – Meaning: Stop action processing for exits 1, 2, and 4. ■ Value: 12 – Label: none – Meaning: Used only on entry to an exit 1 program. ■ Value: 16 – Label: none – Meaning: Used only on entry to an exit 1 program.
ECB-ARGUMENTS	Number of parameters passed to the exit in the address list (R1). The last word of the list is X'80', indicating end-of-list.

Field	Description
ECB-EXIT-NUMBERS	Number identifying the exit being invoked: ■ Value: 1 – Meaning: C1IEXT01 – Exit 1 invoked ■ Value: 2 – Label: C1IEXT02 – Meaning: Exit 2 invoked ■ Value: 3 – Label: C1IEXT03 – Meaning: Exit 3 invoked ■ Value: 4 – Label: C1IEXT04 – Meaning: Exit 4 invoked. ■ Value: 5 – Label: C1IEXT05 – Meaning: Exit 5 invoked. ■ Value: 6 – Label: C1IEXT06 – Meaning: Exit 6 invoked.
ECB-USER-ID	User ID associated with the current processing. Modifiable by exit 5.
ECB-TSO-BATCH-MODE	Indicates whether the current action was requested in foreground or batch. ■ Value: T – Label: TSO – Meaning: Action was requested in foreground ■ Value: B – Label: BATCH – Action was requested in batch

Field	Description
ECB-CALL-FLAG	Indicates whether the current processing relates to building a panel dynamically or to an action request. ■ Value: Y – Label: INTERNAL-CALL – Meaning: Processing involves dynamic TSO panel building ■ Value: N – ACTION-CALL – Processing involves an action request
ECB-SMF-FLAG	Indicates whether Endevor was installed to support SMF recording, as defined by the SMFREC# parameter of the Defaults Table: Y (yes) or N (no).
ECB-SMF-WRITE-FLAG	Applicable if ECB-SMF-FLAG is Y. Indicates whether Endevor is set up to write out SMF action and/or security records, as defined by the SMFACT and SMFSEC parameters of the Defaults Table: Y (yes) or N (no). You can suppress writing SMF records by sorting this field to N using exit 3.

Field	Description
ECB-ACTION-CODE	Action code. Value Action 1 -- Add 2 -- Update 3 -- Retrieve 4 -- Delete 5 -- Generate 6 -- Display 7 -- Move 8 -- Archive 9 -- Reserved 10 -- Reserved 11 -- CHGTYPE 12 -- Signin 13 -- Override Signout 14 -- ENVRNMGR 15 -- Print 16 -- Transfer 17 -- Not used 18 -- Restore
ECB-ACTION-NAME	Literal to describe the current action (ADD, UPDATE, and so forth). Passed to exits 1-4.
ECB-MESSAGE-CODE	Modifiable by exits 1-6. Default value is 0000. Used to construct the message code written to the Endevor Execution Report by the exit handler. A message code precedes each message written to the report, and applies if the ECB-MESSAGE-TEXT field contains a non-blank value. For messages produced by user exits, the error code is formatted as C1Unnnns, where nnnn is defined by this field, and is a severity code associated with ECB-RETURN-CODE (I for return codes less than 8, E for return codes of 8 and above).
ECB-MESSAGES-LENGTH	Not used.

Field	Description
ECB-MESSAGE-TEXT	Modifiable by exits 1-6. Error message associated with the current activity. For exit 1, this is set on entry if a security violation has occurred. For the other exits, this is blank on entry. The first 29 positions are for Endevor-generated information. If there is a message in this field when you return from the user exit, the exit handler records the message in the Endevor Execution Report. The message is preceded in that report with an error code (C1Unnnns format, described above for the ECB-MESSAGE-CODE field).
ECB-CALLER-ORIGIN	Action is executing under Endevor or Quick-Edit. ■ Value: E – Label: CALLER-ENDEVOR – Meaning: Exit is running under Endevor ■ Value: Q – Label: CALLER-QUICKEDIT – Meaning: Exit is running under Quick-Edit
ECB-UNDER-PACKAGE	Action is running under a package ■ Value: Y – Label: PACKAGE-YES – Meaning: Action is running under a package ■ Value: N – Label: PACKAGE-NO – Meaning: Action is not running under a package
ECB-REQUEST-POINTER	A pointer to the Request Information block (paragraph REQUEST-INFO-BLOCK).

Field	Description
ECB-PANEL-TYPE	Indicates the type of panel being processed when the exit was invoked. ■ Value: 0 – Label: NO-PANEL-INFO – Meaning: No panel information is available ■ Value: 1 – Label: ENVIRONMENT-PANEL – Meaning: Exit was called while processing an Environment Definition panel ■ Value: 2 – Label: MAIN-PANEL – Meaning: Exit was called while processing the Primary Options Menu ■ Value: 3 – Label: FOREGROUND-PANEL – Meaning: Exit was called while processing a foreground panel
ECB-EXIT-HOLD-FIELD	Modifiable by exits 1-6. User field, used to save (and pass) information between user exits. Generally, this field contains an address that points to a block of user data. When you are using anchor IDs, this field is used to store address values read by exit programs from the anchor ID table. Any value placed in this field remains there until the end of the Endevor session, or until another user exit loads a new value in this field.
ECB-SECURITY-FUNCTION	Same as ECB-ACTION-CODE (above).
ECB-HIGH-RC	Highest return code set by previously executed system and user exits for this exit number.

3.3 Request Information Block

Endevor passes this block to programs for exits 1-4. Exit 2 programs can change the information in this block.

3.3.1 $REQPDS DSECT sample

```
$REQPDS
$REQPDS  DSECT
REQLEN    DC    AL2(REQSIZE)            LENGTH OF THE BLOCK
REQVERS   DC    AL2(REQ$VERS)           CURRENT VERSION OF THE BLOCK
REQ$VERS  EQU   2                       VERSION OF THE BLOCK
REQID     DC    CL4'$REQ'               BLOCK ID
REQCCID   DC    CL12' '                 CCID FOR ACTION
REQCOMM   DC    CL40' '                 COMMENT FOR ACTION
REQFLAG   DC    F'0'                    SPECIAL FLAGS
REQSISOF  DC    CL1' '                  SIGN-OUT OVERRIDE FOR ACTION
REQ$YES   EQU   C'Y'                    YES
REQ$NO    EQU   C'N'                    NO
REQSISOC  DC    CL1' '                  RETRIEVE COPY ONLY ( Y OR N)
REQEXPND  DC    CL1' '                  RETRIEVE EXPAND INCLUDES ( Y OR N)
REQOVRWR  DC    CL1' '                  RETRIEVE WRITE OVER EXISTING MBR
REQRTCOD  DC    F'0'                    ACTION RETURN CODE
REQELVER  DC    CL2' '                  NEW VERSION
REQCPYBK  DC    CL1' '                  GENERATE COPYBACK OPTION
          DC    CL5' '                   ** RESERVED **
*                                       BLANKS (X'40') IF NOT SIGNED OUT
REQDEL    DC    CL1' '                  DELETE ELEMENT AFTER ACTION (Y/N)
REQMOVWH  DC    CL1' '                  MOVE WITH HISTORY (Y/N)
REQUPDT   DC    CL1' '                  ADD WITH OPTION UPDATE (Y/N)
REQBYGEN  DC    CL1' '                  BYPASS GENERATE PROCESSOR (Y/N)
REQONLCMP DC    CL1' '                  DELETE ONLY COMPONENT (Y/N)
REQSYNC   DC    CL1' '                  SYNCHRONIZE (Y/N)
REQIGGF   DC    CL1' '                  IGNORE GENERATE FAILED (Y/N)
REQBYDP   DC    CL1' '                  BYPASS DELETE PROCESSOR
REQOPGRP  DC    CL8' '                  PROCESSOR GROUP OPTION
REQSOUID  DC    CL8' '                  SIGN-OUT TO USER ID
REQOSRCH  DC    CL1' '                  SEARCH ENVIRONMENT MAP
REQORSGN  DC    CL1' '                  RETAIN SIGNOUT
REQOJUMP  DC    CL1' '                  JUMP
REQOSGNI  DC    CL1' '                  SIGNIN
REQSIZE   EQU   *-$REQPDS               LENGTH OF $REQPDS BLOCK
```

3.3.2 $REQPDS DSECT Field Descriptions

The following list contains brief descriptions of the fields in the $REQPDS DSECT. Unless otherwise noted, exit 2 programs can modify these fields.

Field	Description
REQLEN	Size of the Request Information block ($REQPDS), in bytes. Exit 2 programs cannot modify this field.

Field	Description
REQVERS	Version number of the Request Information block. This should always be 2. Use the REQ$VERS label to verify that the correct version of this DSECT was passed to your exit. Exit 2 programs cannot modify this field.
REQID	Code that identifies this as the Request Information block, for use in rapid identification when you are scanning your modules visually. Value is always "$REQ." Exit 2 programs cannot modify this field.
REQCCID	CCID associated with the current action request.
REQCOMM	Comments associated with the current action request.
REQFLAG	Not used.
REQSISOF	Indicates whether the current action requested a sign-out override. ■ Value: 1 – Label: REQ$YES – Meaning: Action requested a signout override. ■ Value: 2 – Label: REQ$NO. – Meaning: Action did not request a signout override.
REQSISOC	Applicable for a RETRIEVE action. Indicates whether the current action specified NO SIGNOUT: Y (yes) or N (no).
REQEXPND	Applicable for a RETRIEVE action. Indicates whether the current action requested that INCLUDEs be expanded: **Y** (yes) or N (no).
REQOVRWR	Applicable for a RETRIEVE action. Indicates whether the current action requested an overwrite if a member by the same name exists already in the output library: Y (yes) or **N** (no).
REQRTCOD	Action return code.
REQELVER	Applicable for an ADD or TRANSFER action. Indicates the version number of the element being added or transferred. The version number defaults to **01** for an element being added or to the from location version number of the element being transferred. You can change the version number under certain circumstances to any value from 01-99.
REQCPYBK	Applicable for a GENERATE action. Indicates whether you want to copy the element from up the map and then generate it in the stage specified in the SCL.

Field	Description
REQDEL	Applicable for all actions where the deletion of a member/element applies. This includes ADD, ARCHIVE, DELETE, MOVE, TRANSFER, and UPDATE. Indicates whether the member (for an ADD or UPDATE) or element (for an ARCHIVE, DELETE, MOVE, or TRANSFER) will be deleted upon successful completion of the requested action: Y (yes) or N (no). This value is X'40' if a delete does not apply for the current action. Can only be modified during ADD, ARCHIVE, MOVE, TRANSFER, and UPDATE actions.
REQMOVWH	Applicable for a MOVE action. Indicates whether the current action requested a MOVE WITH HISTORY: Y (yes) or N (no).
REQUPDT	Applicable for an ADD action. Indicates whether you want to update the element if it exists already: Y (yes) or N (no).
REQBYGEN	Applicable for an ADD, UPDATE, TRANSFER, and RESTORE action. Indicates whether you want the generate processor to execute after the source management has been performed: Y (yes--bypass the processor) or N (no--execute the processor).
REQONLCMP	Applicable for Endevor ACM users only. Indicates whether you want to delete both the element component list and the element, or the element component list only. Y (yes--delete just the element component list) or N (no--delete the element as well as the element component list). If you specify COMPONENT ONLY, REQDEL will be blank and this (REQONLCMP) flag will be set to y (yes).
REQSYNC	Applicable for a TRANSFER or MOVE action. Indicates whether you want to create a sync level at the at the target when the current level of the element differs from the base level of the FROM location element. See the *SCL Reference Guide* and the *User Guide* for more information.
REQIGGF	Applicable for a TRANSFER action. Indicates whether you want to ignore the *FAILED* flag for the action and perform the action, whether or not the element was previously generated or moved successfully.
REQBYDP	Applicable for a MOVE or TRANSFER action. Indicates whether you want the delete processor to execute for this element. Y—The element will be deleted but the delete processor will not be executed, bypassing the delete of any outputs associated with this element. N—The delete processor will be executed and the elements and all outputs will be deleted.
REQOPGRP	Applicable for an ADD, UPDATE, or GENERATE action. The name of the processor group to be used. If you leave this field blank, the system defaults to the processor group last used for this element (that is, when the element was last generated).

Field	Description
REQSOUID	Applicable for a SIGNIN action. The user ID of the user to whom you are signing out or reassigning the element.
REQOSRCH	Applicable for GENERATE and RETRIEVE actions. Indicates if Endevor is to search the map when building a list.
REQORSGN	Applicable for MOVE and TRANSFER actions. Indicates whether to retain signout at the target location of the action.
REQOJUMP	Applicable for the MOVE action. Indicates whether Endevor allows you to move elements when a version of the element exists at an intermediary stage that is not on the map.
REQOSGNI	Applicable for the MOVE and TRANSFER actions. Indicates whether to sign in elements at the target location of the action.
REQSIZE	Size of the $REQPDS DSECT. Exit 2 cannot modify this field.

3.3.3 REQUEST-INFO-BLOCK control block layout sample

Endevor passes this block to programs for exits 1-4. Exit 2 programs can change the information in this block.

```
01  REQUEST-INFO-BLOCK.                                          00600000
    05  REQ-LENGTH              PIC 9(4) COMP SYNC.              00610000
    05  REQ-CURRENT-VERSION     PIC 9(4) COMP SYNC.              00620000
    05  REQ-BLOCK-ID            PIC X(4).                        00630007
    05  REQ-CCID                PIC X(12).                       00640005
    05  REQ-COMMENT             PIC X(40).                       00650005
    05  REQ-REQUEST-FLAG        PIC 9(8) COMP SYNC.              00660005
    05  REQ-SISO-INDICATOR      PIC X.                           00670005
        88  SIGNOUT-OVERRIDE    VALUE 'Y'.                       00680000
        88  NO-SIGNOUT-OVERRIDE VALUE 'N'.                       00690000
    05  REQ-COPY-INDICATOR      PIC X.                           00700005
        88  RETRIEVE-COPY-ONLY  VALUE 'Y'.                       00710000
        88  RETRIEVE-FOR-UPDATE VALUE 'N'.                       00720000
    05  REQ-EXPINCL-INDICATOR   PIC X.                           00730005
        88  EXPAND-INCLUDES     VALUE 'Y'.                       00740000
        88  NO-EXPAND-INCLUDES  VALUE 'N'.                       00750000
    05  REQ-OVERWRITE-INDICATOR PIC X.                           00760000
        88  WRITE-OVER-MEMBER    VALUE 'Y'.                      00770000
        88  NO-WRITE-OVER-MEMBER VALUE 'N'.                      00780000
    05  REQ-ACTION-RC           PIC 9(8) COMP SYNC.              00790005
    05  REQ-NEW-VERSION         PIC X(2).                        00800000
    05  REQ-GEN-COPYBACK        PIC X.                           00810000
        88  GEN-COPYBACK           VALUE 'Y'.                    00811000
        88  NO-GEN-COPYBACK        VALUE 'N'.                    00812000
    05  FILLER                  PIC X(5).                        00813000
    05  REQ-DELETE-AFTER        PIC X.                           00820005
        88  DELETE-AFTER-ACTION    VALUE 'Y'.                    00830000
        88  NO-DELETE-AFTER-ACTION VALUE 'N'.                    00840000
    05  REQ-MOVE-WITH-HISTORY   PIC X.                           00850005
        88  MOVE-WITH-HISTORY    VALUE 'Y'.                      00860000
```

```
    88  MOVE-WITHOUT-HISTORY VALUE 'N'.                      00870000
 05 REQ-ADD-WITH-UPDATE     PIC X.                           00880005
    88  ADD-WITH-UPDATE     VALUE 'Y'.                       00890000
    88  ADD-WITHOUT-UPDATE  VALUE 'N'.                       00900000
 05 REQ-BYPASS-GEN-PROC     PIC X.                           00910000
    88  BYPASS-GEN-PROC     VALUE 'Y'.                       00911000
    88  NO-BYPASS-GEN-PROC  VALUE 'N'.                       00912000
 05 REQ-DEL-COMPS-ONLY      PIC X.                           00913000
    88  DEL-COMPS-ONLY      VALUE 'Y'.                       00914000
    88  NO-DEL-COMPS-ONLY   VALUE 'N'.                       00915000
 05 REQ-SYNCHRONIZE         PIC X.                           00916000
    88  YES-SYNCHRONIZE     VALUE 'Y'.                       00917000
    88  NO-SYNCHRONIZE      VALUE 'N'.                       00918000
 05 REQ-IGNGEN-FAIL         PIC X.                           00919000
    88  IGNORE-GEN-FAIL     VALUE 'Y'.                       00919100
    88  NO-INGORE-GEN-FAIL  VALUE 'N'.                       00919200
 05 REQ-BYPASS-DEL-PROC     PIC X.                           00919300
    88  BYPASS-DEL-PROC     VALUE 'Y'.                       00919400
    88  NO-BYPASS-DEL-PROC  VALUE 'N'.                       00919500
 05 REQ-PROCESSOR-GROUP     PIC X(8).                        00919600
 05 REQ-SIGNOUT-USRID       PIC X(8).                        00919700
 05 REQ-SEARCH-OPTION       PIC X.                           00919300
    88  SEARCH-ENV-MAP      VALUE 'Y'.                       00919400
    88  NO-SEARCH-ENV-MAP   VALUE 'N'.                       00919500
 05 REQ-RETAIN-SIGNOUT-OPT  PIC X.                           00919300
    88  RETAIN-SIGNOUT      VALUE 'Y'.                       00919400
    88  NO-RETAIN-SIGNOUT   VALUE 'N'.                       00919500
 05 REQ-JUMP-OPTION         PIC X.                           00919300
    88  JUMP-IS-OK          VALUE 'Y'.                       00919400
    88  NO-JUMP-SPECIFIED   VALUE 'N'.                       00919500
 05 REQ-SIGNIN-OPTION       PIC X.                           00919300
    88  SIGNIN-IS-SPECIFIED VALUE 'Y'.                       00919400
    88  NO-SIGNIN-SPECIFIED VALUE 'N'.                       00919500
```

3.3.4 COBOL REQUEST-INFO-BLOCK Field Descriptions

The following list contains brief descriptions of the fields in the REQUEST-INFO-BLOCK PARAGRAPH. Unless otherwise noted, exit 2 programs can modify these fields.

Field	Description
REQ-LENGTH	Size of the Request Information block (REQUEST-INFO-BLOCK), in bytes. Exit 2 programs cannot modify this field.
REQ-CURRENT-VERSION	Version number of the Request Information block. This should always be 2. Use this label to verify that the correct version of this control block layout was passed to your exit. Exit 2 programs cannot modify this field.
REQ-BLOCK-ID	Code that identifies this as the Request Information block, for use in rapid identification when you are scanning your modules visually. Value is always "$REQ." Exit 2 programs cannot modify this field.
REQ-CCID	CCID associated with the current action request.

Field	Description
REQ-COMMENT	Comments associated with the current action request.
REQ-REQUEST-FLAG	Not used.
REQ-SISO-INDICATOR	Indicates whether the current action requested a sign-out override ■ Value: Y – Label: SIGNOUT-OVERRIDE – Meaning: Action requested a signout override ■ Value: N – Label: NO-SIGNOUT-OVERRIDE – Meaning: Action did not request a signout override
REQ-COPY-INDICATOR	Applicable for a RETRIEVE action. Indicates whether the current action specified NO SIGNOUT: Y (yes) or N (no). ■ Value: Y – Label: RETRIEVE-COPY-ONLY – Meaning: Action did not request a signout ■ Value: N – Label: RETRIEVE-FOR-UPDATE – Meaning: Action requested a signout
REQ-OVERWRITE-INDICATOR	Applicable for a RETRIEVE action. Indicates whether the current action requested an overwrite if a member by the same name exists already in the output library: Y (yes) or N (no). ■ Value: Y – Label: WRITE-OVER-MEMBER – Meaning: Action requested overwrite ■ Value: N – Label: NO-WRITE-OVER-MEMBER – Meaning: Action did not request overwrite
REQ-ACTION-RC	Action return code.
REQ-NEW-VERSION	Applicable for an ADD or TRANSFER action. Indicates the version number of the element being added or transferred. The version number defaults to 01 for an element being added or to the from location version number of the element being transferred. You can change the version number under certain circumstances to any value from 01-99.

Field	Description
REQ-GEN-COPYBACK	Applicable for a GENERATE action. Indicates whether you want to copy the element from up the map and then generate it in the stage specified in the SCL. ■ Value: Y – Label: GEN-COPYBACK – Meaning: Action requested that element be copied ■ Value: N – Label: NO-GEN-COPYBACK – Meaning: Action did not request copy
REQ-DELETE-AFTER	Applicable for all actions where the deletion of a member/element applies. This includes ADD, ARCHIVE, DELETE, MOVE, TRANSFER, and UPDATE. Indicates whether the member (for an ADD or UPDATE) or element (for an ARCHIVE, DELETE, MOVE, or TRANSFER) will be deleted upon successful completion of the requested action: Y (yes) or N (no). This value is X'40' if a delete does not apply for the current action. Can only be modified during ADD, UPDATE, TRANSFER, and ARCHIVE actions. ■ Value: – Label: DELETE-AFTER-ACTION – Meaning: Action requested delete ■ Value: N – Label: NO-DELETE-AFTER-ACTION – Meaning: Action did not request delete
REQ-MOVE-WITH-HISTORY	Applicable for a MOVE action. Indicates whether the current action requested a MOVE WITH HISTORY: Y (yes) or N (no). ■ Value: Y – Label: MOVE-WITH-HISTORY – Meaning: Option requested ■ Value: N – Label: MOVE-WITHOUT-HISTORY – Meaning: Option not requested

Field	Description
REQ-ADD-WITH-UPDATE	Applicable for an ADD action. Indicates whether you want to update the element if it exists already: Y (yes) or N (no). ■ Value: Y – Label: ADD-WITH-UPDATE – Meaning: Option requested. ■ Value: N – Label: ADD-WITHOUT-UPDATE – Meaning: Option not requested.
REQ-BYPASS-GEN-PROC	Applicable for an ADD, UPDATE, TRANSFER, and RESTORE action. Indicates whether you want the generate processor to execute after source management has been performed: Y—The element will be deleted but the delete processor will not be executed, bypassing the delete of any outputs associated with this element. N—The delete processor will be executed and the elements and all outputs will be deleted. ■ Value: Y – Label: BYPASS-GEN-PROC – Meaning: Option requested ■ Value: N – Label: NO-BYPASS-GEN-PROC – Meaning: Option not requested.
REQ-DEL-COMPS-ONLY	Applicable for Endevor ACM users only. Indicates whether you want to delete both the element component list and the element, or the element component list only. Y (yes--delete just the element component list) or N (no--delete the element as well as the element component list). If you specify COMPONENT ONLY, REQ-DELETE-AFTER will be blank and this (REQ-DEL-COMPS-ONLY) flag will be set to y (yes). ■ Value: Y – Label: DEL-COMPS-ONLY – Meaning: Option requested ■ Value: N – Label: NO-DEL-COMPS-ONLY – Meaning: Option not requested

Field	Description
REQ-SYNCHRONIZE	Applicable for a TRANSFER or MOVE action. Indicates whether you want Endevor to create a sync level at the target when the current level of the element differs from the base level of the FROM location element. ■ Value: Y – Label: YES-SYNCHRONIZATION – Meaning: Option requested ■ Value: N – Label: NO-SYNCHRONIZATION – Meaning: Option not requested.
REQ-IGNGEN-FAIL	Applicable for a TRANSFER action. Indicates whether you want to ignore the *FAILED* flag for the element and perform the action, whether or not the element was previously generated or moved successfully. ■ Value: Y – Label: IGNORE-GEN-FAIL – Meaning: Option requested. ■ Value: N – Label: NO-IGNORE-GEN-FAIL – Meaning: Option not requested.
REQ-BYPASS-DEL-PROC	Applicable for a MOVE or TRANSFER action. Indicates whether you want the delete processor to execute for this element. Y—The element will be deleted but the delete processor will not be executed, bypassing the delete of any outputs associated with this element. N—The delete processor will be executed and the elements and all outputs will be deleted. ■ Value: Y – Label: BYPASS-DEL-PROC – Meaning: Option requested. ■ Value: N – Label: NO-BYPASS-DEL-PROC – Meaning: Option not requested.
REQ-PROCESSOR-GROUP	Applicable for an ADD, UPDATE, or GENERATE action. The name of the processor group to be used. If you leave this field blank, the system defaults to the processor group last used for this element (that is, when the element was last generated).

Field	Description
REQ-SIGNOUT-USRID	Applicable for a SIGNIN action. The user ID of the user to whom you are signing out or reassigning the element.
REQ-SEARCH-OPTION	Applicable for GENERATE and RETRIEVE actions. Indicates if Endevor is to search the map when building a list. ■ Value: Y – Label: SEARCH-ENV-MAP – Meaning: Option requested ■ Value: N – Label: NO-SEARCH-ENV-MAP – Meaning: Option not requested.
REQ-RETAIN-SIGNOUT-OPT	Applicable for MOVE and TRANSFER actions. Indicates whether to retain signout at the target location of the action. ■ Value: Y – Label: RETAIN-SIGNOUT – Meaning: Option requested ■ Value: N – Label: NO-RETAIN-SIGNOUT – Meaning: Option not requested.
REQ-JUMP-OPTION	Applicable for the MOVE action. Indicates whether Endevor allows you to move elements when a version of the element exists at an intermediary stage that is not on the map. ■ Value: Y – Label: JUMP-IS-OK – Meaning: Option requested ■ Value: N – Label: NO-JUMP-SPECIFIED – Meaning: Option not requested

Field	Description
REQ-SIGNIN-OPTION	Applicable for the MOVE and TRANSFER actions. Indicates whether to sign in elements at the target location of the action. ■ Value: Y – Label: SIGNIN-IS-SPECIFIED – Meaning: Option requested ■ Value: N – Label: NO-SIGNIN-SPECIFIED – Meaning: Option not requested.

3.4 Assembler Environment Block

3.4.1 $ENVDS DSECT sample

Endevor passes this control block to programs written for exits 2-4. It describes the environment used by the current action request. Two copies of this block are passed to each exit.

- The first copy contains information describing the source location used by the action, and is pointed to by Register 1: 4(R1).
- The second copy contains information describing the target location for the action, and applies for ADD, UPDATE, RETRIEVE, MOVE, TRANSFER, ARCHIVE, and RESTORE actions. The second copy is pointed to by the ENVNEXT field in the first copy.

The Environment block serves as an anchor to more detailed information. Depending on the action, each block points to either an Element Master Information block (for environments internal to Endevor) or a File block (for environments external to Endevor).

```
         MACRO
&NAME    $ENVDS &DSCT=YES
         AIF   ('&ampersan.DSCT' NE 'YES').SKPDSCT
$ENVDS   DSECT
         AGO   .SKPEQU
.SKPDSCT ANOP
$ENVDS   DS    0D
.SKPEQU  ANOP
ENVLEN    DC    AL2(ENVSIZE)       LENGTH OF THE BLOCK
ENVVERS   DC    AL2(ENV$VERS)      VERSION OF THE BLOCK
ENV$VERS  EQU   3                      VERSION 3
ENVID     DC    CL4'$ENV'          BLOCK ID
ENVFLAG   DC    H'0'               SPECIAL FLAGS
ENVSPEC   DC    CL1' '             TYPE OF ENVIRONMENT BLOCK            00103114
ENV$C1    EQU   C'C'                        INTERNAL TO ENDEVOR-C1      00103215
ENV$EXT   EQU   C'E'                        EXTERNAL INFORMATION        00103315
ENV$ARCH  EQU   C'A'                        ENDEVOR-C1 ARCHIVE ELEM     00103315
ENVIOTY   DC    CL1' '             I/O  TYPE FOR BLOCK                  00103414
ENV$INP   EQU   C'I'                        INPUT (SOURCE LOCATION)     00103515
ENV$OUT   EQU   C'O'                        OUTPUT (TARGET LOCATION)    00103615
ENVNEXT   DC    F'0'               ADDRESS OF TARGET $ENVDS BLOCK       00103714
ENVFCBA   DC    A(0)               ADDRESS OF THE FILE INFORMATION      00103814
*                                  INTERNAL - C1 ELEMENT INFO           00104014
*                                  EXTERNAL - FILE INFO                 00103914
ENVSITE   DC    CL1' '             CURRENT SITE NAME
ENVSTGID  DC    CL1' '                     STAGE ID ('1' OR '2')
ENVSTGCD  DC    CL1' '                     STAGE CODE (FROM C1DEFLTS)
ENVDATAC  DC    CL1' '             EXIT3, DATA CHANGED SINCE XIT2 (Y/N) 00106017
ENVEVER   DC    H'0'                       VERSION OF THE ELEMENT       00250218
ENVELVL   DC    H'0'                       LEVEL OF THE ELEMENT         00250318
ENVENVM   DC    CL8' '                     ENVIRONMENT NAME
```

```
ENVSTAGE  DC    CL8' '                      STAGE NAME
ENVSYSTM  DC    CL8' '                      SYSTEM NAME
ENVSYSUD  DC    CL8' '                      USER DATA - SYSTEM RECORD    00270111
          DC    CL8' '                      RESERVED
ENVSUBSY  DC    CL8' '                      SUBSYSTEM NAME
ENVSUBUD  DC    CL8' '                      USER DATA - SUBSYSTEM RECORD 00280111
ENVTYPE   DC    CL8' '                      TYPE NAME
ENVTYPUD  DC    CL8' '                      USER DATA - ELM TYPE RECORD  00282011
ENVELEMT  DC    CL10' '                     ELEMENT NAME - 1ST 10 CHARS
          DC    CL8' '                      RESERVED                     00290111
          DC    19F'0'                      RESERVED                     00294014
*                                                                        00104014
*    OFFSET (FROM BEGINNING OF ENVDS) TO THE ELEMENT NAME AREA.          00104014
*    FIRST 2 BYTES IN AREA = THE LENGTH OF THE NAME, THIS IS THEN        00103914
*    FOLLOWED BY THE NAME.                                               00103914
*                                                                        00103914
ENVEAOFF  DC    H'0'                        ELEMENT AREA OFFSET
          DC    H'0'                        RESERVED
ENVBFAREA DS    XL257                       AREA BUFFER SPACE
ENVSIZE   EQU   *-$ENVDS           LENGTH OF THE $ENVDS BLOCK
         MEND
```

3.4.2 $ENVDS DSECT Field Descriptions

The following list contains brief descriptions of the fields in the $ENVDS DSECT. Unless otherwise noted, exit programs cannot modify these fields.

Field	Description
ENVLEN	Size of the Environment block ($ENVDS), in bytes.
ENVVERS	Version number to identify the Environment block ($ENVDS). This should always be 3. Use the ENV$VERS label to verify that the correct version of this DSECT was passed to your exit.
ENVID	Identifies this as an Environment block. Value is always "$ENV".
ENVFLAG	Not used.

Field	Description
ENVSPEC	Indicates whether the location is defined to Endevor. Values are: ■ Value: C – Label: EVNV$C1. – Meaning: The location is not defined to Endevor. ENVFCBA points to an Element Master Information block ■ Value: E – Label: ENV$EXT – Meaning: The location is not defined to Endevor. ENVFCBA points to a File block ■ Value: A – Label: ENV$ARCH – Meaning: The location is defined to Endevor. ENVFCBA points to an Element Master Information block.
ENVIOTY	Indicates whether this block describes a source (from) or target (to) location. Values are: ■ Value: I – Label: ENV$INP. – Meaning: First environment block for source ■ Value: O – ENV$OUT. – Meaning: Second environment block for target
ENVNEXT	Applicable for the first $ENVDS block. Address of the second (target location) $ENVDS block.
ENVFCBA	Address of the Element Master Information block ($ELMDS) or File block ($FILDS) pointed to by this Environment block. If ENVSPEC is C, this field points to the $ELMDS. If ENVSPEC is E, this field points to the $FILDS.
ENVSITE	Endevor site ID, as defined to the Defaults Table.
ENVSTGID	Stage number for the current action request: 1 or 2.
ENVSTGCD	Stage ID for the current action request.
ENVDATAC	Indicates whether source management found that the data had changed. N means no source change took place; Y means change took place. This field is valid only for Exit 3, and is only in the target (second) copy of $ENVDS.

Field	Description
ENVEVER	Version number associated in the MCF with the element. Applicable if the value in the ENVSPEC field is **C**.
ENVELVL	Current level of the element. Applicable if the value in the ENVSPEC field is **C**.
ENVENVM	Environment name for the current action request. Applicable if the value in the ENVSPEC field is **C**.
ENVSTAGE	Stage name for the current action request. Applicable if the value in the ENVSPEC field is **C**.
ENVSYSTM	System name for the current action request. Applicable if the value in the ENVSPEC field is **C**.
ENVSYSUD	Not used.
ENVSOPTS	Not used.
ENVSUBSY	Subsystem name for the current action request. Applicable if the value in the ENVSPEC field is **C**.
ENVSUBUD	Not used.
ENVTYPE	Element type for the current action request. Applicable if the value in the ENVSPEC field is **C**.
ENVTYPUD	Not used.
ENVELEMT	First 10 charactersof the element name specified for the current action request.
ENVEAOFF	Offset from the beginning of the block. This points to the element name area. This area is contained within ENBFAREA. The element name area is composed of 2 parts; a 2 byte length field followed by the element name field. The length value field contains the binary length of the element name.
ENBFAREA	Area Buffer, contains the element name area
ENVSIZE	Size of the $ENVDS DSECT.

3.5 COBOL Environment Block

Endevor passes this control block to programs written for exits 2-4. It describes the environment used by the current action request. Two copies of this block are passed to each exit.

- The copy SRC-ENVIRONMENT-BLOCK contains information describing the source location used by the action.
- The copy TGT-ENVIRONMENT-BLOCK contains information describing the target location for the action and applies for ADD, UPDATE, RETRIEVE, MOVE, TRANSFER, ARCHIVE, and RESTORE actions. This copy is pointed to by the SRC-ENV-NEXT-ENV-POINTER field in the first copy.

The Environment block serves as an anchor to more detailed information. Depending on the action, each block points to either an Element Master Information block (for environments internal to Endevor) or a File block (for environments external to Endevor).

3.5.1 XXX-ENVIRONMENT-BLOCK control block layout sample

```
01  SRC-ENVIRONMENT-BLOCK.                                      00950010
    05  SRC-ENV-LENGTH              PIC 9(4) COMP SYNC.         00960000
    05  SRC-ENV-CURRENT-VERSION     PIC 9(4) COMP SYNC.         00970000
    05  SRC-ENV-BLOCK-ID            PIC X(4).                   00980010
    05  SRC-ENV-FLAGS               PIC 9(4) COMP SYNC.         00990010
    05  SRC-ENV-TYPE-OF-BLOCK       PIC X.                      01000010
        88  SRC-INTERNAL-C1-BLOCK   VALUE 'C'.                  01010010
        88  SRC-EXTERNAL-ENV-BLOCK  VALUE 'E'.                  01020010
        88  SRC-ARCHIVE-FILE        VALUE 'A'.                  01021000
    05  SRC-ENV-IO-TYPE             PIC X.                      01030010
        88  SRC-SOURCE-LOCATION     VALUE 'I'.                  01040010
        88  SRC-TARGET-LOCATION     VALUE 'O'.                  01050010
    05  SRC-ENV-NEXT-ENV-POINTER    PIC 9(8) COMP SYNC.         01060010
    05  SRC-ENV-FILE-POINTER        PIC 9(8) COMP SYNC.         01070010
    05  SRC-ENV-SITE-NAME           PIC X.                      01080010
    05  SRC-ENV-STAGE-ID            PIC X.                      01090010
        88  SRC-STAGE-ONE           VALUE '1'.                  01100010
        88  SRC-STAGE-TWO           VALUE '2'.                  01110010
    05  SRC-ENV-STAGE-CODE          PIC X.                      01120010
    05  FILLER                  PIC X.                          01130005
    05  SRC-ENV-ELEMENT-VERSION     PIC 9(4) COMP SYNC.         01140010
    05  SRC-ENV-ELEMENT-LEVEL       PIC 9(4) COMP SYNC.         01150010
    05  SRC-ENV-ENVIRONMENT-NAME    PIC X(8).                   01160010
    05  SRC-ENV-STAGE-NAME          PIC X(8).                   01170010
    05  SRC-ENV-SYSTEM-NAME         PIC X(8).                   01180010
    05  SRC-ENV-SYSTEM-DATA         PIC X(8).                   01190010
    05  FILLER                      PIC X(8).                   01200010
    05  SRC-ENV-SUBSYSTEM-NAME      PIC X(8).                   01250010
    05  SRC-ENV-SUBSYSTEM-DATA      PIC X(8).                   01260010
    05  SRC-ENV-TYPE-NAME           PIC X(8).                   01270010
    05  SRC-ENV-TYPE-DATA           PIC X(8).                   01280010
    05  SRC-ENV-ELEMENT-NAME        PIC X(10).                  01290010
    05  FILLER             PIC X(8).                            01300005
```

```
    05  FILLER             PIC 9(8) COMP SYNC OCCURS 19 TIMES.
    05  SRC-ENV-ELEMENT-AREA-OFFSET PIC 9(4) COMP SYNC.
    05  FILLER             PIC 9(4) COMP SYNC.
    05  SRC-ENV-AREA-BUFFER         PIC X(257).
01  TGT-ENVIRONMENT-BLOCK.                                          01960010
    05  TGT-ENV-LENGTH              PIC 9(4) COMP SYNC.             01970000
    05  TGT-ENV-CURRENT-VERSION     PIC 9(4) COMP SYNC.             01980000
    05  TGT-ENV-BLOCK-ID            PIC X(4).                       01990010
    05  TGT-ENV-FLAGS               PIC 9(4) COMP SYNC.             02000010
    05  TGT-ENV-TYPE-OF-BLOCK       PIC X.                          02010010
        88  TGT-INTERNAL-C1-BLOCK   VALUE 'C'.                      02020010
        88  TGT-EXTERNAL-ENV-BLOCK  VALUE 'E'.                      02030010
        88  TGT-ARCHIVE-FILE        VALUE 'A'.                      02031000
    05  TGT-ENV-IO-TYPE             PIC X.                          02040010
        88  TGT-SOURCE-LOCATION     VALUE 'I'.                      02050010
        88  TGT-TARGET-LOCATION     VALUE 'O'.                      02060010
    05  TGT-ENV-NEXT-ENV-POINTER    PIC 9(8) COMP SYNC.             02070010
    05  TGT-ENV-FILE-POINTER        PIC 9(8) COMP SYNC.             02080010
    05  TGT-ENV-SITE-NAME           PIC X.                          02090010
    05  TGT-ENV-STAGE-ID            PIC X.                          02100010
        88  TGT-STAGE-ONE           VALUE '1'.                      02110010
        88  TGT-STAGE-TWO           VALUE '2'.                      02120010
    05  TGT-ENV-STAGE-CODE          PIC X.                          02130010
    05  TGT-ENV-ELEMENT-CHANGES     PIC X.
        88  TGT-SOURCE-CHANGES      VALUE 'Y'.
        88  TGT-NO-SOURCE-CHANGES   VALUE 'N'.
    05  TGT-ENV-ELEMENT-VERSION     PIC 9(4) COMP SYNC.             02150010
    05  TGT-ENV-ELEMENT-LEVEL       PIC 9(4) COMP SYNC.             02160010
    05  TGT-ENV-ENVIRONMENT-NAME    PIC X(8).                       02170010
    05  TGT-ENV-STAGE-NAME          PIC X(8).                       02180010
    05  TGT-ENV-SYSTEM-NAME         PIC X(8).                       02190010
    05  TGT-ENV-SYSTEM-DATA         PIC X(8).                       02200010
    05  FILLER                      PIC X(8).                       02210010
    05  TGT-ENV-SUBSYSTEM-NAME      PIC X(8).                       02260010
    05  TGT-ENV-SUBSYSTEM-DATA      PIC X(8).                       02270010
    05  TGT-ENV-TYPE-NAME           PIC X(8).                       02280010
    05  TGT-ENV-TYPE-DATA           PIC X(8).                       02290010
    05  TGT-ENV-ELEMENT-NAME        PIC X(10).                      02300010
    05  FILLER             PIC X(8).                                02310010
    05  FILLER             PIC 9(8) COMP SYNC OCCURS 19 TIMES.
    05  TGT-ENV-ELEMENT-AREA-OFFSET PIC 9(4) COMP SYNC.
    05  FILLER             PIC 9(4) COMP SYNC.
    05  TGT-ENV-AREA-BUFFER         PIC X(257).
*                                                                    02320010
```

3.5.2 SRC-ENVIRONMENT-BLOCK Field Descriptions

The following list contains brief descriptions of the fields in the SRC-ENVIRONMENT-BLOCK control block layout. Unless otherwise noted, exit programs cannot modify these fields.

Field	Description
SRC-ENV-LENGTH	Size of the Environment block (SRC-ENVIRONMENT-BLOCK), in bytes.

Field	Description
SRC-ENV-CURRENT-VERSION	Version number to identify the Environment block (SRC-ENVIRONMENT-BLOCK). This should always be 3.
SRC-EVN-BLOCK-ID	Identifies this as an Environment block. Value is always "$ENV".
SRC-ENV-FLAGS	Not used.
SRC-ENV-TYPE-OF-BLOCK	Indicates whether the location is defined to Endevor. Values are: ■ Value: C – Label: SRC-INTERNAL-C1-BLOCK – Meaning: The location is defined to Endevor SRC-ENV-FILE-POINTER points to an Element Master Information block. ■ Value: E – Label: SRC-EXTERNAL-ENV-BLOCK – Meaning: The location is not defined to Endevor SRC-ENV-FILE-POINTER points to a File block ■ Value: A – Label: SRC-ARCHIVE-FILE – Meaning: The location is defined to Endevor. SRC-ENV-FILE-POINTER points to an Element Master Information block.
SRC-ENV-IO-TYPE	Indicates whether this block describes a source (from) or target (to) location. Values are: ■ Value: I – Label: SRC-SOURCE-LOCATION – Meaning: Block describes an input or source/from location ■ Value: O – Label: SRC-TARGET-LOCATION – Meaning: Block describes an output or target/to location
SRC-ENV-NEXT-ENV-POINTER	Applicable for the first SRC-ENVIRONMENT-BLOCK block. Address of the second (target location) TGT-ENVIRONMENT-BLOCK block.

Field	Description
SRC-ENV	Address of the Element Master Information block (SRC-ELEMENT-MASTER-INFO-BLOCK) or File block (SRC-FILE-CONTROL-BLOCK) pointed to by this Environment block. If SRC-ENV-TYPE-OF-BLOCK is C, this field points to the SRC-ELEMENT-MASTER-INFO-BLOCK. If SRC-ENV-TYPE-OF-BLOCK is E, this field points to the SRC-FILE-CONTROL-BLOCK.
SRC-ENV-SITE-NAME	Endevor site ID, as defined to the Defaults Table.
SRC-ENV-STAGE-ID	Stage number for the current action request: 1 or 2. ■ 1 -- SRC-STAGE-ONE ■ 2 -- SRC-STAGE-TWO
SRC-ENV-STAGE-CODE	Stage ID for the current action request.
FILLER	Not used
SRC-ENV-ELEMENT-VERSION	Version number associated in the MCF with the element. Applicable if the value in the ENVSPEC field is C.
SRC-ENV-ELEMENT-LEVEL	Current level of the element. Applicable if the value in the SRC-ENV-TYPE-OF-BLOCK field is C.
SRC-ENV-ENVIRONMENT-NAME	Environment name for the current action request. Applicable if the value in the SRC-ENV-TYPE-OF-BLOCK field is C.
SRC-ENV-STAGE-NAME	Stage name for the current action request. Applicable if the value in the SRC-ENV-TYPE-OF-BLOCK field is C.
SRC-ENV-SYSTEM-NAME	System name for the current action request. Applicable if the value in the SRC-ENV-TYPE-OF-BLOCK field is C.
SRC-ENV-SYSTEM-DATA	Not used.
SRC-ENV-SUBSYSTEM-NAME	Subsystem name for the current action request. Applicable if the value in the SRC-ENV-TYPE-OF-BLOCK field is C.
SRC-ENV-SUBSYSTEM-DATA	Not used.
SRC-ENV-TYPE-NAME	Element type for the current action request. Applicable if the value in the SRC-ENV-TYPE-OF-BLOCK field is C.
SRC-ENV-TYPE-DATA	Not used.
SRC-ENV-ELEMENT-NAME	First 10 characters of the element name specified for the current action request.
SRC-ENV-ELEMENT-NAME-OFFSET	Binary offset value from the beginning of the block which points to the actions element name area. The element name area is made up of 2 parts; a 2 byte binary length field, followed by the element name. The length field contains the actual length of the element name which immediately

Field	Description
SRC-ENV-ELEMENT-AREA-OFFSET	Binary offset value from the beginning of the block which points to the actions element name area. The element name area is made up of 2 parts; a 2 byte binary length field, followed by the element name. The length field contains the actual length of the element name which immediately follows.
SRC-ENV-AREA-BUFFER	Area buffer space which contains the element name area.

3.5.3 TGT-ENVIRONMENT-BLOCK Field Descriptions

The following list contains brief descriptions of the fields in the TGT-ENVIRONMENT-BLOCK control block layout. Unless otherwise noted, exit programs cannot modify these fields.

Field	Description
TGT-ENV-LENGTH	Size of the Environment block (TGT-ENVIRONMENT-BLOCK), in bytes.
TGT-ENV-CURRENT-VERSION	Version number to identify the Environment block (TGT-ENVIRONMENT-BLOCK). This should always be 3.
TGT-ENV-BLOCK-ID	Identifies this as an Environment block. Value is always "$ENV".
TGT-ENV-FLAGS	Not used.
TGT-ENV-TYPE-OF-BLOCK	Indicates whether the location is defined to Endevor. Values are: ■ Value: C – Label: TGT-INTERNAL-C1-BLOCK – Meaning: The location is defined to Endevor TGT-ENV-FILE-POINTER points to an Element Master Information block ■ Value: E – Label: TGT-EXTERNAL-ENV-BLOCK – Meaning: The location is not defined to Endevor. TGT-ENV-FILE-POINTER points to a File block ■ Value: A – Label: TGT-ARCHIVE-FILE – Meaning: The location is defined to Endevor TGT-ENV-FILE-POINTER points to an Element Master Information block

Field	Description
TGT-ENV-IO-TYPE	Indicates whether this block describes a source (from) or target (to) location. Values are: ■ Value: I – Label: TGT-SOURCE-LOCATION – Meaning: Block describes an input source/from location ■ Value: O – Label: TGT-TARGET-LOCATION – Meaning: Block describes an output target/to location
TGT-ENV-NEXT-ENV-POINTER.	Applicable for the first TGT-ENVIRONMENT-BLOCK block.
TGT-ENV-FILE-POINTER	Address of the Element Master Information block (TGT-ELEMENT-MASTER-INFO-BLOCK) or File block (TGT-FILE-CONTROL-BLOCK) pointed to by this Environment block. If TGT-ENV-TYPE-OF-BLOCK is C, this field points to the TGT-ELEMENT-MASTER-INFO-BLOCK. If TGT-ENV-TYPE-OF-BLOCK is E, this field points to the TGT-FILE-CONTROL-BLOCK.
TGT-ENV-SITE-NAME	Endevor site ID, as defined to the Defaults Table.
TGT-ENV-STAGE-ID	Stage number for the current action request: 1 or 2 ■ 1 -- TGT-STAGE-ONE ■ 2 -- TGT-STAGE-TWO
TGT-ENV-ELEMENT-CHANGES	Indicates whether source management found that the data had changed. N means no source change took place. This field is valid only for Exit 3, and is only in the TGT-ENVIRONMENT-BLOCK block. ■ 1 -- TGT-SOURCE-CHANGES ■ 2 -- TGT-NO-SOURCE-CHANGES
TGT-ENV-ELEMENT-VERSION	Version number associated in the MCF with the element. Applicable if the value in the ENVSPEC field is C.
TGT-ENV-ELEMENT-LEVEL	Current level of the element. Applicable if the value in the TGT-ENV-TYPE-OF-BLOCK field is C.
TGT-ENV-ENVIRONMENT-NAME	Environment name for the current action request. Applicable if the value in the TGT-ENV-TYPE-OF-BLOCK field is C.
TGT-ENV-STAGE-NAME	Stage name for the current action request. Applicable if the value in the TGT-ENV-TYPE-OF-BLOCK field is C.
TGT-ENV-SYSTEM-NAME	System name for the current action request. Applicable if the value in the TGT-ENV-TYPE-OF-BLOCK field is C.

Field	Description
TGT-ENV-SYSTEM-DATA	Not used.
TGT-ENV-SUBSYSTEM-NAME	Subsystem name for the current action request. Applicable if the value in the TGT-ENV-TYPE-OF-BLOCK field is C.
TGT-ENV-SUBSYSTEM-DATA	Not used.
TGT-ENV-TYPE-NAME	Element type for the current action request. Applicable if the value in the TGT-ENV-TYPE-OF-BLOCK field is C.
TGT-ENV-TYPE-DATA	Not used.
TGT-ENV-ELEMENT-NAME	First 10 characters of the element name specified for the current action request.
TGT-ENV-ELEMENT-NAME-AREA-OFFSET	Binary offset value from the beginning of the block which points to the action element name area. The element name area is made up of 2 parts; a 2 byte binary length field, followed by the element name. The length field contains the actual length of the element name which immediately follows.
TGT-ENV-AREA-BUFFER	Area buffer space which contains the element name area.

3.6 Assembler Element Master Information Block

3.6.1 $ELMDS DSECT sample

Endevor passes this block to programs written for exits 2-4. It describes the element specified by the current action request. It is pointed to by the Environment block, which describes a location internal to Endevor (field ENVFCBA).

Note: On a Transfer action, this control block is not filled out if the element does not exist at the target.

```
         MACRO                                                          00010000
&NAME    $ELMDS &DSCT=YES                                               00020000
         AIF   ('&ampersan.DSCT' NE 'YES').SKPDSCT                               00030000
$ELMDS   DSECT                                                          00040000
         AGO   .SKPEQU                                                  00050000
.SKPDSCT ANOP                                                           00060000
$ELMDS   DS    0F                                                       00070000
.SKPEQU  ANOP                                                           00071000
ELMLEN   DC     AL2(ELMSIZE)       LENGTH OF THE BLOCK                  00160001
ELMVERS  DC     AL2(ELM$VERS)      VERSION OF THE BLOCK                 00170001
ELM$VERS EQU    5                      VERSION 5                        00180001
ELMID    DC     CL4'$ELM'          BLOCK ID                             00190001
ELMFLAGS DC     F'0'               FLAGS                                00200001
ELMMVERS DS     H                  ELEMENT VERSION                      01511000
ELMMLLVL DS     H                  ELEMENT LEVEL NUMBER                 01512000
ELMMINS  DS     F                  NUMBER OF INSERTS - HIGH DELTA       01513100
ELMMDEL  DS     F                  NUMBER OF DELETES - HIGH DETLA       01513200
ELMNAME  DS     CL10               ELEMENT NAME                         01513301
ELMCCID  DS     CL12               USER'S CHANGE CONTROL NUMBER         01513401
ELMM1STL DS     H                  BASE LEVEL (PROD=0,STAGE=PROD LL)    01513501
*                                                                       01514000
*         BASE  INFORMATION                                             01515000
*                                                                       01516000
ELMMBTOT DS     F                  BASE # STATEMENTS                    01610000
ELMMBDTE DS     CL6                BASE DATE (YYMMDD)                   01620000
ELMMBTIM DS     CL4                BASE TIME (HHMM)                     01630000
ELMBCOM  DS     CL40               BASE COMMENT                         01631000
ELMBUSID DS     CL8                BASE USER ID                         01632000
*                                                                       01632100
*         ADD/UPDATE/GEN INFORMATION                                    01632200
*                                                                       01632300
ELMPUSID DS     CL8                GENERATE USER ID                     01632400
ELMPDSTP DS     CL6                GENERATE DATE (YYMMDD)               01632500
ELMPTSTP DS     CL4                GENERATE TIME (HHMM)                 01632600
ELMFDSN  DS     CL44               ADDED/UPDATED/RESTORED FROM - DSN    01632700
ELMFMBR  DS     CL10               ADDED/UPDATED/RESTORE  FROM - MEMBER 01632800
*                                                                       01633000
*         PROCESS INFORMATION                                           01634000
*                                                                       01635000
ELMPRFLG DS     BL1                PROCESSOR FLAG                       01640000
ELMPRONX EQU    B'00000000'            NO PROCESSOR EXECUTED            01641000
ELMPROFL EQU    B'10000000'            PROCESSOR FAILED
```

```
ELMPROGN EQU    B'01000000'            LAST PROCESSOR WAS GENERATE
ELMPROMV EQU    B'00100000'            LAST PROCESSOR WAS MOVE
ELMPRODL EQU    B'00010000'            LAST PROCESSOR WAS DELETE
ELMPROUN EQU    B'00001000'            RESTORE PROCESS NOT SPECIFIED
         DS     CL3                ** RESERVED **                      01645101
ELMLPROD DS     CL6                PROCESSOR LAST DATE
ELMLPROT DS     CL4                PROCESSOR LAST TIME
ELMLPROU DS     CL8                PROCESSOR USER ID
ELMLPRON DS     CL10               PROCESSOR NAME                      01649000
ELMMPRC  DS     H                  PROCESSOR RETURN CODE
ELMMRC   DS     H                  PROCESSOR C1 RETURN CODE
ELMMPRSA EQU    B'00100000'             PROCESSOR HAD A SYSTEM ABEND
ELMMPRUA EQU    B'00010000'             PROCESSOR HAD A USER  ABEND
ELMPCOM  DS     CL40               PROCESSOR - LAST COMMENT
*                                                                       01649800
*         LAST CHANGE INFORMATION                                       01649900
*                                                                       01650000
ELMCCOM  DS     CL40               LEVEL COMMENT                       01650100
ELMMLDTE DS     CL6                LEVEL DATE (YYMMDD)                 01650200
ELMMLTME DS     CL4                LEVEL TIME (HHMM)                   01650300
ELMLUSID DS     CL8                LEVEL USER ID                       01650401
ELMMLACT DS     CL8                LAST ACTION                         01650501
ELMMLTOT DS     F                  LAST LEVEL TOTAL                    01650600
*                                                                       01651000
*         MOVE  INFORMATION                                             01652000
*                                                                       01653000
*  (NOTE:  THE FOLLOWING THREE FIELDS ARE VALID ONLY FOR                00750000
*          RELEASES OF ENDEVOR/MVS PRIOR TO 3.5)                        00760000
*                                                                       00770000
ELMMVDTE DS     CL6                MOVE DATE (STAGE II ONLY)           01660000
ELMMVTIM DS     CL4                MOVE TIME (STAGE II ONLY)           01670000
ELMMUSID DS     CL8                MOVE - USER ID                      01680000
*                                                                       01840000
*         RETRIEVE INFORMATION                                          01850000
*                                                                       01860000
ELMRDSTP DS     CL6                RETRIEVE DATE (YYMMDD)              01990000
ELMRTSTP DS     CL4                RETRIEVE TIME (HHMM)                02000000
ELMRIID  DS     CL8                RETRIEVE USER ID                    02010000
ELMRCOM  DS     CL40               RETRIEVE COMMENT                    02020000
ELMTDSN  DS     CL44               RETRIEVE - TO DSN                   02030000
ELMTMBR  DS     CL10               RETRIEVE - TO MEM                   02040000
         DS     CL30               ** RESERVED **                      02050001
*                                                                       01840000
*         NEW 3.5 FIELDS                                                01850000
*                                                                       01860000
ELMRCCID DS     CL12               RETRIEVE CCID
ELMGCCID DS     CL12               GENERATE CCID
*                                                                       00960000
*     THE FOLLOWING LAST ACTION FIELDS CONTAIN INFORMATION              00970000
*     ASSOCIATED WITH THE LAST ELEMENT-MODIFYING ACTION                 00980000
*                                                                       00990000
ELMODACT DS     CL8                LAST ELEMENT-MODIFYING ACTION       01000000
ELMLCCID DS     CL12               LAST ACTION CCID
ELMLCOMM DS     CL40               LAST ACTION COMMENT
ELMLUID  DS     CL8                LAST ACTION USER
ELMLADT  DS     CL6                LAST ACTION DATE
ELMLATM  DS     CL4                LAST ACTION TIME
*                                                                       01060000
ELMSOID  DS     CL8                SIGNOUT ID
```

```
*                                                                       01080000
*     THE FOLLOWING FROM ENDEVOR LOCATION FIELDS CONTAIN                01090000
*     INFORMATION ASSOCIATED WITH THE LAST MOVE, TRANSFER               01100000
*     OR RESTORE ACTION                                                 01110000
*                                                                       01120000
ELFRFLAG DS     X                 FROM ENDEVOR LOCATION FLAG FIELD     01130000
ELFR$VAL EQU    X'80'             FIELDS CONTAIN INFORMATION           01140000
ELFRACT  DS     CL8               ACTION NAME                          01150000
ELFRUSID DS     CL8               USERID ASSOCIATED WITH ACTION        01160000
ELFRDATE DS     CL6               DATE ASSOCIATED WITH ACTION          01170000
ELFRTIME DS     CL4               TIME ASSOCIATED WITH ACTION          01180000
ELFRSITE DS     CL1               FROM ENDEVOR LOCATION SITE           01190000
ELFRENV  DS     CL8               FROM ENDEVOR LOCATION ENVIRONMENT    01200000
ELFRSYS  DS     CL8               FROM ENDEVOR LOCATION SYSTYEM        01210000
ELFRSBS  DS     CL8               FROM ENDEVOR LOCATION SUBSYSTEM      01220000
ELFRELM  DS     CL10              FROM ENDEVOR LOCATION ELEMENT        01230000
ELFRTYPE DS     CL8               FROM ENDEVOR LOCATION TYPE           01240000
ELFRSTG# DS     CL1               FROM ENDEVOR LOCATION STAGE NBR      01250000
ELFRVER  DS     H                 FROM ENDEVOR LOCATION VERSION        01260000
ELFRLVL  DS     H                 FROM ENDEVOR LOCATION LEVEL          01270000
*                                                                       01120000
* ADDITIONAL ELEMENT FIELDS WHICH WERE ADDED FOR E/MVS 3.6 FOLLOW       01120000
*                                                                       01120000
ELMUPDT  DS     CL10              UPDATE NAME
ELMBASE  DS     CL10              BASE NAME
ELMLOWL  DS     CL2               LOWER LEVEL (STAGE) REACHED LEVEL
ELMIPCTG DS     XL1               PERCENT OF PREV INSERTS DELETED
ELMDPCTG DS     XL1               PERCENT OF PREV DELETES RE-INSERTED
ELMFPESD DS     CL8               ESD NAME FOOTPRINTED (IF OBJECT)
ELMEDFMT DS     CL1               ELEMENT DELTA FORMAT
ELMENPAK DS     CL1               Y=ELE BASE NOT PACKED
ELMLPROV DS     XL1               LAST PROCESSOR VERSION NUMBER
ELMPROL  DS     XL1               LAST PROCESSOR LEVEL NUMBER
ELMFMID  DS     XL1               RECORD FORMAT ID
ELMV2R1  EQU     21               RELEASE 2.1 FORMAT ID
ELMV2R5  EQU     25               RELEASE 2.5 FORMAT ID
ELMV3R5  EQU     35               RELEASE 3.5 FORMAT ID
ELMV3R6  EQU     36               RELEASE 3.6 FORMAT ID
ELMCFMID EQU     36               CURRENT RECORD FORMAT ID
ELMXDNAM DS     CL8               CONFIGURATION DELTA MEMBER NAME
ELMXBVER DS     H                 CONFIGURATION DELTA VERSION NUMBER
ELMXBTOT DS     F                 CONFIGURATION BASE TOTAL
ELMXLTOT DS     F                 CONFIGURATION LAST LEVEL TOTAL
ELMXBLVL DS     H                 CONFIGURATION BASE LEVEL NUMBER
ELMXLLVL DS     H                 CONFIGURATION LAST LEVEL NUMBER
ELMXINS  DS     H                 INSERTS AT LAST LEVEL
ELMXDLS  DS     H                 DELETES AT LAST LEVEL
ELMXRGIN DS     H                 CONFIGURATION REGRESSION INSERT PCT
ELMXRGDL DS     H                 CONFIGURATION REGRESSION DELS   PCT
ELMXBDTE DS     CL6               CONFIGURATION BASE DATE (YYMMDD)
ELMXBTIM DS     CL4               CONFIGURATION BASE TIME (HHMM)
ELMXLDTE DS     CL6               CONFIGURATION LAST LVL DATE (YYMMDD)
ELMXLTIM DS     CL4               CONFIGURATION LAST LVL TIME (HHMM)
ELMXFLAG DS     XL1               CONFIGURATION COMPONENT STAT FLAG
ELMX$MON EQU    X'80'             CONFIGURATION BUILT BY MONITOR=...
ELMX$CPY EQU    X'40'             CONFIGURATION BUILT BY COPY/RESTORE
ELMXDFMT DS     CL1               CONFIGURATION DELTA FORMAT
ELMXUPDT DS     CL1               Y = COMP BASE IN DELTA LIB
ELMSPKG  DS     CL16              LAST PROCESSED PACKAGE ID
```

```
ELMSPKGT DS     FL8                 TIMESTAMP ASSOC WITH PACKAGE
ELMOPKG  DS     CL16                PACKAGE ASSOC WITH ELEMENT OUTPUTS
ELMOPKGT DS     FL8                 TIMESTAMP ASSOC WITH PACKAGE
ELMPRGRP DS     CL8                 PROCESSOR GROUP NAME
ELMCPKG  DS     CL16                PACKAGE ASSOC WITH ELEMENT OUTPUTS
ELMCPKGT DS     FL8                 TIMESTAMP ASSOC WITH PACKAGE
ELMFFFLG DS     CL1                 "FROM" FLAG FILE TYPE
ELMFFNONE EQU   C' '                NO "FROM" D/S INFO
ELMFFPATH EQU   C'P'                P = PATH, DATA IN PATH AREA
ELMFFDSN  EQU   C'D'                D = DSN, DATA IN ELMFDSN/ELMFMBR
ELMTFFLG DS     CL1                 "TO" FLAG FILE TYPE
ELMTFNONE EQU   C' '                NO "TO" D/S INFO
ELMTFPATH EQU   C'P'                P = PATH, DATA IN PATH AREA
ELMTFDSN  EQU   C'D'                D = DSN, DATA IN ELMTDSN/ELMTMBR
*                                                                       01840000
*   OFFSET (FROM BEGINNING OF ELMDS) TO THE PATH AND FILE NAME AREAS.   01850000
*   FIRST 2 BYTES IN AREA = THE LENGTH OF THE NAME, THIS IS THEN        01860000
*   FOLLOWED BY THE NAME.                                               01860000
*                                                                       01860000
ELMFPAOFF DS    H                   "FROM" PATH AREA OFFSET
ELMFNAOFF DS    H                   "FROM" FILE NAME AREA OFFSET
ELMTPAOFF DS    H                   "TO" PATH AREA OFFSET
ELMTNAOFF DS    H                   "TO" FILE NAME AREA OFFSET
*                                                                       01840000
*   OFFSET (FROM BEGINNING OF ELMDS) TO THE ELEMENT NAME AREA.          01850000
*   FIRST 2 BYTES IN AREA = THE LENGTH OF THE NAME, THIS IS THEN        01860000
*   FOLLOWED BY THE NAME.                                               01860000
*                                                                       01860000
ELMEAOFF  DS    H                   ELEMENT AREA OFFSET
         DS     CL32                    ** RESERVED **
ELMBFAREA DS    XL2320              AREA BUFFER SPACE
ELMSIZE   EQU   *-$ELMDS            LENGTH OF THE $ELMDS BLOCK          02150000
         MEND                                                           02160000
```

3.6.2 $ELMDS DSECT Field Descriptions

This section contains brief descriptions of the fields in the $ELMDS DSECT. None of these fields are modifiable by exits.

Field	Description
ELMLEN	Size of the Element Master Information block ($ELMDS), in bytes.
ELMVERS	Version number to identify the Element Master Information block ($ELMDS). This should always be 5. Use the ELM$VERS label to verify that the correct version of this DSECT was passed to your exit.
ELMID	Identifies this as the Element Master Information block. Value is always "$ELM".
ELMFLAGS	Not used.
ELMMVERS	Version number of the element specified in the action request. This field has the same value as the ENVEVER field in the Environment block.

Field	Description
ELMMLLVL	The current (highest) level of the element. This field has the same value as the ENVELVL field in the Environment block.
ELMMINS	Count of statements added to the element source by the last action that invoked source management.
ELMMDEL	Count of statements deleted from the element source by the last action that invoked source management.
ELMNAME	Name of the MCF element name. Element names longer than 10, or names containing special characters or mixed case characters will have a generated MCF element name. The actual name will be placed in the buffer area. See also ELMEAOFF.
ELMCCID	Last-specified CCID for the element.
ELMM1STL	The base level of the element.
ELMMBTOT	Count of source statements in the base level of the element in the current stage.
ELMMBDTE	Base date for the element.
ELMMBTIM	Base time for the element.
ELMBCOM	Base comment for the element.
ELMBUSID	Base user ID for the element.
ELMPUSID	Generate processor user ID for the element.
ELMPDSTP	Generate processor date for the element.
ELMPTSTP	Generate processor time for the element.
ELMFDSN	Name of the data set from which the element was last added, updated, or restored.
ELMFMBR	Name of the member from which the element was last added, updated, or restored. Applicable if the source data set was a library; blank otherwise.

Field	Description
ELMPRFLG	Flag used to describe the last processor run for the element (and related information). This field may reference a delete processor (in the case where a delete processor was the last processor run for the current action). Values are: ■ Value: X'00' – Label: ELMPRONX – Meaning: No processor has been run yet for the element ■ Value: X'80' – Label: ELMPROFL – Meaning: The last processor failed ■ Value: X'40' – Label: ELMPROGN – Meaning: The last processor was a generate processor ■ Value: X'20' – Label: ELMPROMV – Meaning: The last processor was a move processor ■ Value: X'10' – Label: ELMPRODL – Meaning: The last processor was a delete processor ■ Value: X'08' – Label: ELMPROUN – Meaning: The element has been restored but not yet processed. Therefore, processor statistics are unchanged from the archived statistics for the element
ELMLPROD	Processor date for the element.
ELMLPROT	Processor time for the element.
ELMLPROU	Processor user ID for the element.
ELMLPRON	(Element) name of the processor last run for the element.
ELMMPRC	Processor return code for the element.

Field	Description
ELMMRC	Endevor return code for the element. Values are: ▪ Value: X'10' – Label: ELMMPRSA. – Meaning: Processor had a system abend ▪ Value: X'20' – Label: ELMMPRUS – Meaning: Processor had a user abend
ELMMPRSA	Not used.
ELMMPRUA	Not used.

Field	Description
ELMPCOM	Generate processor comment for the element.
ELMCCOM	Current-level comment for the element.
ELMMLDTE	Current-level date for the element.
ELMMLTME	Current-level time for the element.
ELMLUSID	Current-level user ID for the element.
ELMMLACT	Most recent action executed for the element. Once executed, each action is recorded as the last action--except ARCHIVE, LIST, PRINT, and DISPLAY.
ELMMLTOT	Count of source statements for the element, as of the current level.
ELMMVDTE	Date the element was last moved.
ELMMVTIM	Time the element was last moved.
ELMMUSID	User ID associated with the last move of the element.
ELMRDSTP	Date the element was last retrieved.
ELMRTSTP	Time the element was last retrieved.
ELMRIID	User ID associated with the last retrieve of the element.
ELMRCOM	Comment associated with the last retrieve of the element.
ELMTDSN	Name of the data set to which the element was last retrieved.
ELMTMBR	Name of the member to which the element was last retrieved. Applicable if the target data set for the last retrieve request was a library; blank otherwise.
ELMRCCID	CCID associated with the last retrieve of the element.

Field	Description
ELMGCCID	CCID associated with the last generate of the element.
ELMODACT	Last action that modified the element. For example, an ADD or UPDATE would modify an element. A RETRIEVE would not.
ELMLCCID	CCID associated with the last action that modified the element.
ELMLCOMM	Comment associated with the last action that modified the element.
ELMLUID	User ID of the user who performed the last action that modified an element.
ELMLADT	Date of last action that modified the element.
ELMLATM	Time when the last modifying action was performed.
ELMSOID	ID of the user to whom the modified element is signed out.
ELFRFLAG	Flag used to indicate whether or not there is information in the following fields which describe the FROM information in a MOVE or TRANSFER action. For example, the value X'80' with the label ELFR$VAL means the flag is set to ON.
ELFRACT	The last action. Either MOVE, TRANSFER, or RESTORE.
ELFRUSID	The user ID of the person performing the last MOVE, TRANSFER, or RETRIEVE.
ELFRDATE	The date of the last MOVE, TRANSFER, or RETRIEVE.
ELFRTIME	The time of the last MOVE, TRANSFER, or RETRIEVE.
ELFRSITE	The site of the last MOVE, TRANSFER, or RETRIEVE.
ELFRENV	The environment of the last MOVE, TRANSFER, or RETRIEVE.
ELFRSYS	The system of the last MOVE, TRANSFER, or RETRIEVE.
ELFRSBS	The subsystem of the last MOVE, TRANSFER, or RETRIEVE.
ELFRELM	The MCF name of the element in the FROM location.
ELFRTYPE	The type of the element in the FROM location.
ELFRSTG#	The stage number of the element in the FROM location.
ELFRVER	The version of the element in the FROM location.
ELFRLVL	The level of the element in the FROM location.
ELMUPDT	Element name in delta library.
ELMBASE	Element name in base library.
ELMLOWL	Internal use only.
ELMIPCTG	Percentage of previous inserts deleted.

Field	Description
ELMDPCTG	Percentage of previous deletes reinserted.
ELMFPESD	Name of footprinted object (ESD) module.
ELMEDFMT	Element delta format.
ELMENPAK	Element base is not compressed when value is **Y**.
ELMLPROV	Version number of last processor.
ELMPROL	Level number of last processor.
ELMFMID	ID of record format. For Release 3.6, all elements returned should have the value 36 in this field.
ELMV2R1	21 - Release 2.1 format ID.
ELMV2R5	25 - Release 2.5 format ID.
ELMV3R5	35 - Release 3.5 format ID.
ELMV3R6	36 - Release 3.6 format ID.
ELMCFMID	Current record format ID for all releases 3.6 or greater this is equal to 36.
ELMXDNAM	Component list delta member name.
ELMXBVER	Component list delta member version number.
ELMXBTOT	Component list base total.
ELMXLTOT	Component list last level total.
ELMXBLVL	Component list base level number.
ELMXLLVL	Component list last level number.
ELMXINS	Inserts at last level.
ELMXDLS	Deletes at last level.
ELMXRGIN	Component list regression insert percent.
ELMXRGDL	Component list regression deletes percent.
ELMXBDTE	Component list base date.
ELMXBTIM	Component list base time.
ELMXLTIM	Component list last level time.
ELMXLDAT	Component list last level date.
ELMXFLAG	Component list status flag.
ELMX$MON	Component list built using MONITOR=COMPONENT statement in processor.
ELMX$CPY	Component list built by copy or restore.

Field	Description
ELMXDFMT	Component list delta format.
ELMXUPDT	Internal use only.
ELMSPKG	ID of last package processed.
ELMSPKGT	Time stamp on last processed package.
ELMOPKG	Package associated with element outputs.
ELMOPKGT	Time stamp on package outputs.
ELMPRGRP	Processor group name.
ELMSIZE	Size of the $ELMDS DSECT.
ELMCPKG	Element locked to package.
ELMCPKGT	Element locked to package time stamp.
ELMFFFLG	■ From data set flag type. ■ Blank—no from the data set ■ P—Dataset is a path name; data can be found in the path or filename data areas. See also the fields ELMFPAOFF and ELMFNAOFF. ■ D—Dataset is a DSN. See also the fields ELMFDSN and ELMFMBK.
ELMTFFLG	To data set flag type. ■ Blank—No from data set ■ P—Dataset is a path name. Path and file names are contained in data areas. See also fields ELMTPAOFF and ELMTNAOFF. ■ D—Dataset is a DSN. See fields ELMTDSN and ELMTMBR.
ELMFPAOFF	When ELMFFFLG indicates a path, the field will contain the offset value to the "FROM" path name area from the beginning of $ELMDS. The path area is composed of a 2 byte binary path name length, followed by the name.
ELMFNAOFF	When ELMFFFLG indicates a path, the field will contain the offset value to the "FROM" file name area from the beginning of $ELMDS. The name area is composed of a 2 byte binary file name followed by the name.
ELMTPAOFF	When ELMTFFLG indicates a path. This field will contain the offset value to the "TO" path name area from the beginning of the $ELMDS. The path name is composed of 2 byte binary path name length value, followed by the name.

Field	Description
ELMTNAOFF	When ELMTFFLG indicates a path. This field will contain the offset value to the "TO" path name area from the beginning of the $ELMDS. The path name is composed of 2 byte binary path name length value, followed by the name.
ELMEAOFF	Contains the offset to the element name areas from the beginning of $ELMDS. The element area is composed of a 2 byte binary element name length value, followed by the name. This area contains the true name of the element which can be up to 255 character.
ELMBFAREA	This buffer area contains the space where all areas are contained.

3.7 COBOL Element Master Information Block

Endevor passes this block to programs written for exits 2-4. It describes the element specified by the current action request. It is pointed to by the Environment block, which describes a location internal to Endevor (field SRC-ENV-FILE-POINTER).

Note: On a Transfer action, this control block is not filled out if the element does not exist at the target.

3.7.1 ELEMENT-MASTER-INFO-BLOCK control block layout sample

```
01  SRC-ELEMENT-MASTER-INFO-BLOCK.                                  01340010

    05  SRC-ELM-LENGTH              PIC 9(4) COMP SYNC.             01350000
    05  SRC-ELM-CURRENT-VERSION     PIC 9(4) COMP SYNC.             01360000
    05  SRC-ELM-BLOCK-ID            PIC X(4).                       01370010
    05  SRC-ELM-FLAGS               PIC 9(8) COMP SYNC.             01380010
    05  SRC-ELM-ELEMENT-VERSION     PIC 9(4) COMP SYNC.             01390010
    05  SRC-ELM-ELEMENT-LEVEL       PIC 9(4) COMP SYNC.             01400010
    05  SRC-ELM-NUMBER-INSERTS      PIC 9(8) COMP SYNC.             01410010
    05  SRC-ELM-NUMBER-DELETES      PIC 9(8) COMP SYNC.             01420010
    05  SRC-ELM-ELEMENT-NAME        PIC X(10).                      01430010
    05  SRC-ELM-CURRENT-CCID        PIC X(12).                      01440010
    05  SRC-ELM-BASE-LEVEL-NUMBER   PIC 9(4) COMP SYNC.             01450010
    05  SRC-ELM-BASE-TOTAL-STMTS    PIC 9(8) COMP SYNC.             01460010
    05  SRC-ELM-BASE-DATE           PIC 9(6).                       01470010
    05  SRC-ELM-BASE-TIME           PIC 9(4).                       01480010
    05  SRC-ELM-BASE-COMMENT        PIC X(40).                      01490010
    05  SRC-ELM-BASE-USERID         PIC X(8).                       01500010
    05  SRC-ELM-GEN-USERID          PIC X(8).                       01510010
    05  SRC-ELM-GEN-DATE            PIC 9(6).                       01520010
    05  SRC-ELM-GEN-TIME            PIC 9(4).                       01530010
    05  SRC-ELM-FROM-DSN            PIC X(44).                      01540010
    05  SRC-ELM-FROM-MEMBER         PIC X(10).                      01550010
    05  SRC-ELM-PROCESSOR-FLAG      PIC X.                          01560010
    05  FILLER                  PIC X(3).                           01570005
    05  SRC-ELM-PROCESSOR-LAST-DATE PIC 9(6).                       01580010
    05  SRC-ELM-PROCESSOR-LAST-TIME PIC 9(4).                       01590010
    05  SRC-ELM-PROCESSOR-USERID    PIC X(8).                       01600010
    05  SRC-ELM-PROCESSOR-NAME      PIC X(10).                      01610010
    05  SRC-ELM-PROCESSOR-RC        PIC 9(4) COMP SYNC.             01620010
    05  SRC-ELM-PROCESSOR-C1-RC     PIC 9(4) COMP SYNC.             01630010
    05  SRC-ELM-PROCESSOR-LAST-COMMENT PIC X(40).                   01640010
    05  SRC-ELM-LEVEL-COMMENT       PIC X(40).                      01650010
    05  SRC-ELM-LEVEL-DATE          PIC 9(6).                       01660010
    05  SRC-ELM-LEVEL-TIME          PIC 9(4).                       01670010
    05  SRC-ELM-LEVEL-USERID        PIC X(8).                       01680010
    05  SRC-ELM-LAST-ACTION         PIC X(8).                       01690010
    05  SRC-ELM-LAST-LEVEL-TOTAL    PIC 9(8) COMP SYNC.             01700010
    05  SRC-ELM-MOVE-DATE           PIC 9(6).                       01710010
    05  SRC-ELM-MOVE-TIME           PIC 9(4).                       01720010
    05  SRC-ELM-MOVE-USERID         PIC X(8).                       01730010
    05  SRC-ELM-RETRIEVE-DATE       PIC 9(6).                       01740010
    05  SRC-ELM-RETRIEVE-TIME       PIC 9(4).                       01750010
```

```
     05  SRC-ELM-RETRIEVE-USERID     PIC X(8).                    01760010
     05  SRC-ELM-RETRIEVE-COMMENT    PIC X(40).                   01770010
     05  SRC-ELM-RETRIEVE-TO-DSN     PIC X(44).                   01780010
     05  SRC-ELM-RETRIEVE-TO-MEMBER  PIC X(10).                   01790010
     05  FILLER                  PIC X(30).                       01800005
     05  SRC-ELM-RET-CCID            PIC X(12).                   01801000
     05  SRC-ELM-GEN-CCID            PIC X(12).                   01802000
     05  SRC-ELM-MOD-ACTION          PIC X(8).                    01803000
     05  SRC-ELM-ACTION-CCID         PIC X(12).                   01804000
     05  SRC-ELM-ACTION-COMM         PIC X(40).                   01805000
     05  SRC-ELM-ACTION-USER         PIC X(8).                    01806000
     05  SRC-ELM-ACTION-DATE         PIC 9(6).                    01807000
     05  SRC-ELM-ACTIOM-TIME         PIC 9(4).                    01808000
     05  SRC-ELM-SIGNOUT-ID          PIC X(8).                    01809000
     05  SRC-ELM-LOC-FLAG            PIC X.                       01809100
     05  SRC-ELM-FR-ACTION           PIC X(8).                    01809200
     05  SRC-ELM-FR-ACT-USERID       PIC X(8).                    01809300
     05  SRC-ELM-FR-ACT-DATE         PIC 9(6).                    01809400
     05  SRC-ELM-FR-ACT-TIME         PIC 9(4).                    01809500
     05  SRC-ELM-FR-SITE             PIC X.                       01809600
     05  SRC-ELM-FR-ENV              PIC X(8).                    01809700
     05  SRC-ELM-FR-SYS              PIC X(8).                    01809800
     05  SRC-ELM-FR-SUBSYS           PIC X(8).                    01809900
     05  SRC-ELM-FR-ELEMENT          PIC X(10).                   01810000
     05  SRC-ELM-FR-TYPE             PIC X(8).                    01810100
     05  SRC-ELM-FR-STG-NBR          PIC X.                       01810200
     05  SRC-ELM-FR-VER              PIC 9(4) COMP SYNC.          01810300
     05  SRC-ELM-FR-LEVEL            PIC 9(4) COMP SYNC.          01810400
                                                                   01810500
* NEW ELEMENT FIELDS FOR E/MVS 3.6 FOLLOW                          01810006
                                                                   01810500
     05  SRC-ELM-UPDATE-MEMBER-NAME  PIC X(10).                   01810400
     05  SRC-ELM-BASE-MEMBER-NAME    PIC X(10).                   01810400
     05  SRC-ELM-LOWER-LEVEL-IND     PIC X(2).                    01810400
     05  SRC-ELM-PCT-PREV-INSERT-DEL PIC X.                       01810400
     05  SRC-ELM-PCT-PREV-DELS-REINS PIC X.                       01810400
     05  SRC-ELM-ESD-FP-NAME-IF-OBJ  PIC X(8).                    01810400
     05  SRC-ELM-DELTA-FORMAT        PIC X.                       01810400
         88  SRC-ELM-FORWARD-DELTA   VALUE 'F'.                   01100010
         88  SRC-ELM-REVERSE-DELTA  VALUE 'R'.                    01110010
     05  SRC-ELM-PACKED-INDICATOR    PIC X.                       01810400
         88  SRC-ELM-BASE-NOT-PACKED VALUE 'Y'.                   01100010
     05  SRC-ELM-LAST-PROC-VERSION   PIC X.                       01810400
     05  SRC-ELM-LAST-PROC-LEVEL     PIC X.                       01810400
     05  SRC-ELM-RECORD-FORMAT       PIC X.                       01810400
     05  SRC-ELM-CONF-DELTA-MEM-ID   PIC X(8).                    01810400
     05  SRC-ELM-CONF-DELTA-VERSION  PIC S9(4) COMP SYNC.         01810400
     05  SRC-ELM-CONF-BASE-TOTAL     PIC S9(8) COMP SYNC.         01810400
     05  SRC-ELM-CONF-LAST-LVL-TOTAL PIC S9(8) COMP SYNC.         01810400
     05  SRC-ELM-CONF-BASE-LVL-NBR   PIC S9(4) COMP SYNC.         01810400
     05  SRC-ELM-CONF-LAST-LVL-NBR   PIC S9(4) COMP SYNC.         01810400
     05  SRC-ELM-INSERTS-LAST-LEVEL  PIC S9(4) COMP SYNC.         01810400
     05  SRC-ELM-DELETES-LAST-LEVEL  PIC S9(4) COMP SYNC.         01810400
     05  SRC-ELM-CONF-REGRES-INS-PCT PIC S9(4) COMP SYNC.         01810400
     05  SRC-ELM-CONF-REGRES-DEL-PCT PIC S9(4) COMP SYNC.         01810400
     05  SRC-ELM-CONF-BASE-DATE      PIC 9(6).                    01810400
     05  SRC-ELM-CONF-BASE-TIME      PIC 9(4).                    01810400
     05  SRC-ELM-CONF-LAST-LVL-DATE  PIC 9(6).                    01810400
     05  SRC-ELM-CONF-LAST-LVL-TIME  PIC 9(4).                    01810400
```

```
            05  SRC-ELM-CONF-COMP-STAT-FLAG PIC X.                      01810400
            05  SRC-ELM-CONF-DELTA-FORMAT   PIC X.                      01810400
            05  SRC-ELM-CONF-DELTA-BASE-FLG PIC X.                      01810400
                88  SRC-ELM-CONF-BASE-IN-DELTA VALUE 'Y'.               01100010
            05  SRC-ELM-LAST-PROC-PACKAGE   PIC X(16).                  01810400
            05  SRC-ELM-PACKAGE-TIMESTAMP1  PIC X(4).                   01810400
            05  SRC-ELM-PACKAGE-TIMESTAMP2  PIC X(4).                   01810400
            05  SRC-ELM-PACKAGE-WITH-OUTPUT PIC X(16).                  01810400
            05  SRC-ELM-OPACKAGE-TIMESTAMP1 PIC X(4).                   01810400
            05  SRC-ELM-OPACKAGE-TIMESTAMP2 PIC X(4).                   01810400
            05  SRC-ELM-PROCESSOR-GROUP     PIC X(8).                   01810400
            05  SRC-ELM-PACKAGE-RESERVED    PIC X(16).
            05  SRC-ELM-PKG-RES-TIMESTAMP1  PIC X(4).                   01810400
            05  SRC-ELM-PKG-RES-TIMESTAMP2  PIC X(4).                   01810400
            05  SRC-ELM-FROM-FILE-TYPE-FLG  PIC X.                      01810400
                88  SRC-ELM-FROM-NA-TYPE  VALUE ' '.                    01100010
                88  SRC-ELM-FROM-DSN-TYPE  VALUE 'D'.                   01100010
                88  SRC-ELM-FROM-PATH-TYPE VALUE 'P'.                   01100010
            05  SRC-ELM-TO-FILE-TYPE-FLG  PIC X.                        01810400
                88  SRC-ELM-TO-NA-TYPE  VALUE ' '.                      01100010
                88  SRC-ELM-TO-DSN-TYPE  VALUE 'D'.                     01100010
                88  SRC-ELM-TO-PATH-TYPE VALUE 'P'.                     01100010
            05  SRC-ELM-FROM-PATH-AREA-OFFSET  PIC S9(4) COMP SYNC.     01810400
            05  SRC-ELM-FROM-NAME-AREA-OFFSET  PIC S9(4) COMP SYNC.     01810400
            05  SRC-ELM-TO-PATH-AREA-OFFSET    PIC S9(4) COMP SYNC.     01810400
            05  SRC-ELM-TO-NAME-AREA-OFFSET    PIC S9(4) COMP SYNC.     01810400
            05  SRC-ELM-NAME-AREA-OFFSET       PIC S9(4) COMP SYNC.     01810400
            05  FILLER                    PIC X(32).                    01810006
            05  SRC-ELM-AREA-BUFFER       PIC X(2320).                  01810006
        *                                                               01810006
*
        01  TGT-ELEMENT-MASTER-INFO-BLOCK.                              02350010

            05  TGT-ELM-LENGTH              PIC 9(4) COMP SYNC.         02360000
            05  TGT-ELM-CURRENT-VERSION     PIC 9(4) COMP SYNC.         02370000
            05  TGT-ELM-BLOCK-ID            PIC X(4).                   02380010
            05  TGT-ELM-FLAGS               PIC 9(8) COMP SYNC.         02390010
            05  TGT-ELM-ELEMENT-VERSION     PIC 9(4) COMP SYNC.         02400010
            05  TGT-ELM-ELEMENT-LEVEL       PIC 9(4) COMP SYNC.         02410010
            05  TGT-ELM-NUMBER-INSERTS      PIC 9(8) COMP SYNC.         02420010
            05  TGT-ELM-NUMBER-DELETES      PIC 9(8) COMP SYNC.         02430010
            05  TGT-ELM-ELEMENT-NAME        PIC X(10).                  02440010
            05  TGT-ELM-CURRENT-CCID        PIC X(12).                  02450010
            05  TGT-ELM-BASE-LEVEL-NUMBER   PIC 9(4) COMP SYNC.         02460010
            05  TGT-ELM-BASE-TOTAL-STMTS    PIC 9(8) COMP SYNC.         02470010
            05  TGT-ELM-BASE-DATE           PIC 9(6).                   02480010
            05  TGT-ELM-BASE-TIME           PIC 9(4).                   02490010
            05  TGT-ELM-BASE-COMMENT        PIC X(40).                  02500010
            05  TGT-ELM-BASE-USERID         PIC X(8).                   02510010
            05  TGT-ELM-GEN-USERID          PIC X(8).                   02520010
            05  TGT-ELM-GEN-DATE            PIC 9(6).                   02530010
            05  TGT-ELM-GEN-TIME            PIC 9(4).                   02540010
            05  TGT-ELM-FROM-DSN            PIC X(44).                  02550010
            05  TGT-ELM-FROM-MEMBER         PIC X(10).                  02560010
            05  TGT-ELM-PROCESSOR-FLAG      PIC X.                      02570010
            05  FILLER                  PIC X(3).                       02580010
            05  TGT-ELM-PROCESSOR-LAST-DATE PIC 9(6).                   02590010
            05  TGT-ELM-PROCESSOR-LAST-TIME PIC 9(4).                   02600010
            05  TGT-ELM-PROCESSOR-USERID    PIC X(8).                   02610010
```

```
     05  TGT-ELM-PROCESSOR-NAME      PIC X(10).                  02620010
     05  TGT-ELM-PROCESSOR-RC        PIC 9(4) COMP SYNC.         02630010
     05  TGT-ELM-PROCESSOR-C1-RC     PIC 9(4) COMP SYNC.         02640010
     05  TGT-ELM-PROCESSOR-LAST-COMMENT PIC X(40).               02650010
     05  TGT-ELM-LEVEL-COMMENT       PIC X(40).                  02660010
     05  TGT-ELM-LEVEL-DATE          PIC 9(6).                   02670010
     05  TGT-ELM-LEVEL-TIME          PIC 9(4).                   02680010
     05  TGT-ELM-LEVEL-USERID        PIC X(8).                   02690010
     05  TGT-ELM-LAST-ACTION         PIC X(8).                   02700010
     05  TGT-ELM-LAST-LEVEL-TOTAL    PIC 9(8) COMP SYNC.         02710010
     05  TGT-ELM-MOVE-DATE           PIC 9(6).                   02720010
     05  TGT-ELM-MOVE-TIME           PIC 9(4).                   02730010
     05  TGT-ELM-MOVE-USERID         PIC X(8).                   02740010
     05  TGT-ELM-RETRIEVE-DATE       PIC 9(6).                   02750010
     05  TGT-ELM-RETRIEVE-TIME       PIC 9(4).                   02760010
     05  TGT-ELM-RETRIEVE-USERID     PIC X(8).                   02770010
     05  TGT-ELM-RETRIEVE-COMMENT    PIC X(40).                  02780010
     05  TGT-ELM-RETRIEVE-TO-DSN     PIC X(44).                  02790010
     05  TGT-ELM-RETRIEVE-TO-MEMBER  PIC X(10).                  02800010
     05  FILLER                  PIC X(30).                      02810010
     05  TGT-ELM-RET-CCID            PIC X(12).                  02811000
     05  TGT-ELM-GEN-CCID            PIC X(12).                  02812000
     05  TGT-ELM-MOD-ACTION          PIC X(8).                   02813000
     05  TGT-ELM-ACTION-CCID         PIC X(12).                  02814000
     05  TGT-ELM-ACTION-COMM         PIC X(40).                  02815000
     05  TGT-ELM-ACTION-USER         PIC X(8).                   02816000
     05  TGT-ELM-ACTION-DATE         PIC 9(6).                   02817000
     05  TGT-ELM-ACTIOM-TIME         PIC 9(4).                   02818000
     05  TGT-ELM-SIGNOUT-ID          PIC X(8).                   02819000
     05  TGT-ELM-LOC-FLAG            PIC X.                      02819100
     05  TGT-ELM-FR-ACTION           PIC X(8).                   02819200
     05  TGT-ELM-FR-ACT-USERID       PIC X(8).                   02819300
     05  TGT-ELM-FR-ACT-DATE         PIC 9(6).                   02819400
     05  TGT-ELM-FR-ACT-TIME         PIC 9(4).                   02819500
     05  TGT-ELM-FR-SITE             PIC X.                      02819600
     05  TGT-ELM-FR-ENV              PIC X(8).                   02819700
     05  TGT-ELM-FR-SYS              PIC X(8).                   02819800
     05  TGT-ELM-FR-SUBSYS           PIC X(8).                   02819900
     05  TGT-ELM-FR-ELEMENT          PIC X(10).                  02820000
     05  TGT-ELM-FR-TYPE             PIC X(8).                   02820100
     05  TGT-ELM-FR-STG-NBR          PIC X.                      02820200
     05  TGT-ELM-FR-VER              PIC 9(4) COMP SYNC.         02820300
     05  TGT-ELM-FR-LEVEL            PIC 9(4) COMP SYNC.         02820400

* NEW ELEMENT FIELDS FOR E/MVS 3.6 FOLLOW                        01810006
                                                                 01810500
     05  TGT-ELM-UPDATE-MEMBER-NAME  PIC X(10).                  01810400
     05  TGT-ELM-BASE-MEMBER-NAME    PIC X(10).                  01810400
     05  TGT-ELM-LOWER-LEVEL-IND     PIC X(2).                   01810400
     05  TGT-ELM-PCT-PREV-INSERT-DEL PIC X.                      01810400
     05  TGT-ELM-PCT-PREV-DELS-REINS PIC X.                      01810400
     05  TGT-ELM-ESD-FP-NAME-IF-OBJ  PIC X(8).                   01810400
     05  TGT-ELM-DELTA-FORMAT        PIC X.                      01810400
         88  TGT-ELM-FORWARD-DELTA   VALUE 'F'.                  01100010
         88  TGT-ELM-REVERSE-DELTA  VALUE 'R'.                   01110010
     05  TGT-ELM-PACKED-INDICATOR    PIC X.                      01810400
         88  TGT-ELM-BASE-NOT-PACKED VALUE 'Y'.                  01100010
     05  TGT-ELM-LAST-PROC-VERSION   PIC X.                      01810400
     05  TGT-ELM-LAST-PROC-LEVEL     PIC X.                      01810400
```

```
     05  TGT-ELM-RECORD-FORMAT       PIC X.                      01810400
     05  TGT-ELM-CONF-DELTA-MEM-ID   PIC X(8).                   01810400
     05  TGT-ELM-CONF-DELTA-VERSION  PIC S9(4) COMP SYNC.        01810400
     05  TGT-ELM-CONF-BASE-TOTAL     PIC S9(8) COMP SYNC.        01810400
     05  TGT-ELM-CONF-LAST-LVL-TOTAL PIC S9(8) COMP SYNC.        01810400
     05  TGT-ELM-CONF-BASE-LVL-NBR   PIC S9(4) COMP SYNC.        01810400
     05  TGT-ELM-CONF-LAST-LVL-NBR   PIC S9(4) COMP SYNC.        01810400
     05  TGT-ELM-INSERTS-LAST-LEVEL  PIC S9(4) COMP SYNC.        01810400
     05  TGT-ELM-DELETES-LAST-LEVEL  PIC S9(4) COMP SYNC.        01810400
     05  TGT-ELM-CONF-REGRES-INS-PCT PIC S9(4) COMP SYNC.        01810400
     05  TGT-ELM-CONF-REGRES-DEL-PCT PIC S9(4) COMP SYNC.        01810400
     05  TGT-ELM-CONF-BASE-DATE      PIC 9(6).                   01810400
     05  TGT-ELM-CONF-BASE-TIME      PIC 9(4).                   01810400
     05  TGT-ELM-CONF-LAST-LVL-DATE  PIC 9(6).                   01810400
     05  TGT-ELM-CONF-LAST-LVL-TIME  PIC 9(4).                   01810400
     05  TGT-ELM-CONF-COMP-STAT-FLAG PIC X.                      01810400
     05  TGT-ELM-CONF-DELTA-FORMAT   PIC X.                      01810400
     05  TGT-ELM-CONF-DELTA-BASE-FLG PIC X.                      01810400
         88  TGT-ELM-CONF-BASE-IN-DELTA VALUE 'Y'.               01100010
     05  TGT-ELM-LAST-PROC-PACKAGE   PIC X(16).                  01810400
     05  TGT-ELM-PACKAGE-TIMESTAMP1  PIC X(4).                   01810400
     05  TGT-ELM-PACKAGE-TIMESTAMP2  PIC X(4).                   01810400
     05  TGT-ELM-PACKAGE-WITH-OUTPUT PIC X(16).                  01810400
     05  TGT-ELM-OPACKAGE-TIMESTAMP1 PIC X(4).                   01810400
     05  TGT-ELM-OPACKAGE-TIMESTAMP2 PIC X(4).                   01810400
     05  TGT-ELM-PROCESSOR-GROUP     PIC X(8).                   01810400
     05  TGT-ELM-PACKAGE-RESERVED    PIC X(16).
     05  TGT-ELM-PKG-RES-TIMESTAMP1  PIC X(4).                   01810400
     05  TGT-ELM-PKG-RES-TIMESTAMP2  PIC X(4).                   01810400
     05  TGT-ELM-FROM-FILE-TYPE-FLG  PIC X.                      01810400
         88  TGT-ELM-FROM-NA-TYPE  VALUE ' '.                    01100010
         88  TGT-ELM-FROM-DSN-TYPE  VALUE 'D'.                   01100010
         88  TGT-ELM-FROM-PATH-TYPE VALUE 'P'.                   01100010
     05  TGT-ELM-TO-FILE-TYPE-FLG  PIC X.                        01810400
         88  TGT-ELM-TO-NA-TYPE  VALUE ' '.                      01100010
         88  TGT-ELM-TO-DSN-TYPE  VALUE 'D'.                     01100010
         88  TGT-ELM-TO-PATH-TYPE VALUE 'P'.                     01100010
     05  TGT-ELM-FROM-PATH-AREA-OFFSET  PIC S9(4) COMP SYNC.     01810400
     05  TGT-ELM-FROM-NAME-AREA-OFFSET  PIC S9(4) COMP SYNC.     01810400
     05  TGT-ELM-TO-PATH-AREA-OFFSET    PIC S9(4) COMP SYNC.     01810400
     05  TGT-ELM-TO-NAME-AREA-OFFSET    PIC S9(4) COMP SYNC.     01810400
     05  TGT-ELM-NAME-AREA-OFFSET       PIC S9(4) COMP SYNC.     01810400
     05  FILLER                    PIC X(32).                    01810006
     05  TGT-ELM-AREA-BUFFER       PIC X(2320).                  01810006
*                                                                02820010
*                                                                02830010
```

3.7.2 SRC-ELEMENT-MASTER-INFO-BLOCK Field Descriptions

This section contains brief descriptions of the fields in the SRC-ELEMENT-MASTER-INFO-BLOCK. None of these fields are modifiable by exits.

Field	Description
SRC-ELM-LENGTH	Size of the Element Master Information block SRC-ELEMENT-MASTER-INFO-BLOCKS), in bytes.
SRC-ELM-CURRENT-VERSION	Version number to identify the Element Master Information block (SRC-ELEMENT-MASTER-INFO-BLOCK). This should always be 5.
SRC-ELM-BLOCK-ID	Identifies this as the Element Master Information block. Value is always "$ELM".
SRC-ELM-FLAGS	Not used.
SRC-ELM-ELEMENT-VERSION	Version number of the element specified in the action request. This field has the same value as the SRC-ENV-ELEMENT-LEVEL field in the Environment block.
SRC-ELM-ELEMENT-LEVEL	The current (highest) level of the element. This field has the same value as the SRC-ENV-ELEMENT-LEVEL field in the Environment block.
SRC-ELM-NUMBER-INSERTS	Count of statements added to the element source by the last action that invoked source management.
SRC-ELM-NUMBER-DELETES	Count of statements deleted from the element source by the last action that invoked source management.
SRC-ELM-ELEMENT-NAME	Name of the MCF element name. Element names longer than 10, or names containing special characters or mixed case characters will have a generated MCF element name. The actual name will be placed in the buffer area. See also SRC-ELM-NAME-ARE-OFFSET.
SRC-ELM-CURRENT-CCID	Last-specified CCID for the element.
SRC-ELM-BASE-LEVEL-NUMBER	The base level of the element.
SRC-ELM-BASE-TOTAL-STMTS	Count of source statements in the base level of the element in the current stage.
SRC-ELM-BASE-DATE	Base date for the element.
SRC-ELM-BASE-TIME	Base time for the element.
SRC-ELM-BASE-COMMENT	Base comment for the element.
SRC-ELM-BASE-USERID	Base user ID for the element.
SRC-ELM-GEN-USERID	Generate processor user ID for the element.
SRC-ELM-GEN-DATE	Generate processor date for the element.
SRC-ELM-GEN-TIME	Generate processor time for the element.
SRC-ELM-FROM-DSN	Name of the data set from which the element was last added, updated, or restored.

Field	Description
SRC-ELM-FROM-MEMBER	Name of the member from which the element was last added, updated, or restored. Applicable if the source data set was a library; blank otherwise.
SRC-ELM-PROCESSOR-FLAG	Flag used to describe the last processor run for the element (and related information). This field may reference a delete processor (in the case where a delete processor was the last processor run for the current action). Values are: ■ Value: X'00' – Meaning: No processor has been run yet for the element ■ Value: X'80' – Meaning: The last processor failed ■ Value: X'40' – Meaning: The last processor was a generate processor ■ Value: X'20' – Meaning: The last processor was a move processor ■ Value: X'10' – Meaning: The last processor was a delete processor ■ Value: X'08' – Meaning: The element has been restored but not yet processed. Therefore, processor statistics are unchanged from the archived statistics for the element
SRC-ELM-PROCESSOR-LAST-DATE	Processor date for the element.
SRC-ELM-PROCESSOR-LAST-TIME	Processor time for the element.
SRC-ELM-PROCESSOR-USERID	Processor user ID for the element.
SRC-ELM-PROCESSOR-NAME	(Element) name of the processor last run for the element.
SRC-ELM-PROCESS-RC	Processor return code for the element.
SRC-ELM-PROCESSOR-C1-RC	Endevor return code for the element. Values are: ■ Value: X'10' – Meaning: Processor had a system abend ■ Value: X'20' – Meaning: Processor had a user abend

Field	Description
SRC-ELM-PROCESSOR-LAST-COMMENTS	Generate processor comment for the element.
SRC-ELM-LEVEL-COMMENT	Current-level comment for the element.
SRC-ELM-LEVEL-DATE	Current-level date for the element.
SRC-ELM-LEVEL-TIME	Current-level time for the element.
SRC-ELM-LEVEL-USERID	Current-level user ID for the element.
SRC-ELM-LAST-ACTION	Most recent action executed for the element. Once executed, each action is recorded as the last action--except ARCHIVE, LIST, PRINT, and DISPLAY.
SRC-ELM-LAST-LEVEL-TOTAL	Count of source statements for the element, as of the current level.
SRC-ELM-MOVE-DATE	Date the element was last moved.
SRC-ELM-MOVE-TIME	Time the element was last moved.
SRC-ELM-MOVE-USERID	User ID associated with the last move of the element.
SRC-ELM-RETRIEVE-DATE	Date the element was last retrieved.
SRC-ELM-RETRIEVE-TIME	Time the element was last retrieved.
SRC-ELM-RETRIEVE-USERID	User ID associated with the last retrieve of the element.
SRC-ELM-RETRIEVE-COMMENT	Comment associated with the last retrieve of the element.
SRC-ELM-RETRIEVE-TO-DSN	Name of the data set to which the element was last retrieved.
SRC-ELM-RETRIEVE-TO-MEMBER	Name of the member to which the element was last retrieved. Applicable if the target data set for the last retrieve request was a library; blank otherwise.
SRC-ELM-RET-CCID	CCID associated with the last retrieve of the element.
SRC-ELM-GEN-CCID	CCID associated with the last generate of the element.
SRC-ELM-MOD-ACTION	Last action that modified the element. For example, an ADD or UPDATE would modify an element. A RETRIEVE would not.
SRC-ELM-ACTION-CCID	CCID associated with the last action that modified the element.
SRC-ELM-ACTION-COMM	Comment associated with the last action that modified the element.
SRC-ELM-ACTION-USER	User ID of the user who performed the last action that modified an element.
SRC-ELM-ACTION-DATE	Date of last action that modified the element.
SRC-ELM-ACTION-TIME	Time when the last modifying action was performed.
SRC-ELM-SIGNOUT-ID	ID of the user to whom the modified element is signed out.

Field	Description
SRC-ELM-LOC-FLAG	Flag used to indicate whether or not there is information in the following fields which describe the FROM information in a MOVE or TRANSFER action. For example, the value X'80' means the flag is set to ON.
SRC-ELM-FR-ACTION	The last action. Either MOVE, TRANSFER, or RESTORE.
SRC-ELM-FR-ACT-USERID	The user ID of the person performing the last MOVE, TRANSFER, or RETRIEVE.
SRC-ELM-FR-ACT-DATE	The date of the last MOVE, TRANSFER, or RETRIEVE.
SRC-ELM-FR-ACT-TIME	The time of the last MOVE, TRANSFER, or RETRIEVE.
SRC-ELM-FR-SITE	The site of the last MOVE, TRANSFER, or RETRIEVE.
SRC-ELM-FR-ENV	The environment of the last MOVE, TRANSFER, or RETRIEVE.
SRC-ELM-FR-SYS	The system of the last MOVE, TRANSFER, or RETRIEVE.
SRC-ELM-FR-SUBSYS	The subsystem of the last MOVE, TRANSFER, or RETRIEVE.
SRC-ELM-FR-ELEMENT	The name of the MCF element in the FROM location.
SRC-ELM-FR-TYPE	The type of the element in the FROM location.
SRC-ELM-FR-STG-NBR	The stage number of the element in the FROM location.
SRC-ELM-FR-VER	The version of the element in the FROM location.
SRC-ELM-FR-LEVEL	The level of the element in the FROM location.
SRC-ELM-UPDATE-MEMBER-NAME	Element name in delta library.
SRC-ELM-BASE-MEMBER-NAME	Element name in base library.
SRC-ELM-LOWER-LEVEL-IND	Internal use only.
SRC-ELM-PCT-PREV-INSERT-DEL	Percentage of previous inserts deleted.
SRC-ELM-PCT-PREV-DELS-REINS	Percentage of previous deletes reinserted.
SRC-ELM-ESD-FP-NAME-IF-OBJ	Name of footprinted object (ESD) module.
SRC-ELM-DELTA-FORMAT	Element delta format. ■ Value: F – Label: SRC-ELM-FORWARD-DELTA – Meaning: Forward format ■ Value: R – Label: SRC-ELM-REVERSE-DELTA – Meaning: Reverse format

Field	Description
SRC-ELM-PACKED-INDICATOR	Element base is not compressed when value is Y. Y -- SRC-ELM-BASE-NOT-PACKED
SRC-ELM-LAST-PROC-VERSION	Version number of last processor.
SRC-ELM-LAST-PROC-LEVEL	Level number of last processor.
SRC-ELM-RECORD-FORMAT	ID of record format. For all releases 3.6 or greater, all elements returned should have the value x'36' in this field. ■ x'21' -- Release 2.1 format ID. ■ x'25' -- Release 2.5 format ID. ■ x'35' -- Release 3.5 format ID. ■ x'36' -- Release 3.6 format ID. ■ x'36' --Current Record format ID.
SRC-ELM-CONF-DELTA-MEM-ID	Component list delta member name.
SRC-ELM-CONF-DELTA-VERSION	Component list delta member version number.
SRC-ELM-CONF-BASE-TOTAL	Component list base total.
SRC-ELM-CONF-LAST-LVL-TOTAL	Component list last level total.
SRC-ELM-CONF-BASE-LVL-NBR	Component list base level number.
SRC-ELM-CONF-LAST-LVL-NBR	Component list last level number.
SRC-ELM-INSERTS-LAST-LEVEL	Inserts at last level.
SRC-ELM-DELETES-LAST-LEVEL	Deletes at last level.
SRC-ELM-CONF-REGRES-INS-PCT	Component list regression insert percent.
SRC-ELM-CONF-REGRES-DEL-PCT	Component list regression deletes percent.
SRC-ELM-CONF-BASE-DATE	Component list base date.
SRC-ELM-CONF-BASE-TIME	Component list base time.
SRC-ELM-CONF-LAST-LVL-TIME	Component list last level time.
SRC-ELM-CONF-LAST-LVL-DATE	Component list last level date.
SRC-ELM-CONF-COMP-STAT-FLAG	Component list status flag. ■ Value: X'80' – Meaning: Component list built using MONITOR=COMPONENT statement in processor ■ Value: X'40' – Meaning: Component list built by copy or restore

Field	Description
SRC-ELM-CONF-DELTA-FORMAT	Component list delta format. Y -- SRC-ELM-CONF-BASE-IN-DELTA
SRC-LAST-PROC-PACKAGE	ID of last packaged processed.
SRC-ELM-PACKAGE-TIMESTAMP1	Time stamp on last processed package.
SRC-ELM-PACKAGE-TIMESTAMP2	Time stamp on last processed package.
SRC-ELM-PACKAGE-WITH-OUTPUT	Package associated with element outputs.
SRC-ELM-OPACKAGE-TIMESTAMP1	Time stamp on package outputs.
SRC-ELM-OPACKAGE-TIMESTAMP2	Time stamp on package outputs.
SRC-PROCESSOR-GROUP-NAME	Processor group name.
SRC-ELM-PACKAGE-RESERVED	Element locked to package
SRC-ELM-PKG-RES-TIMESTAMP1	Timestamp when element is locked to the package
SRC-ELM-PKG-RES-TIMESTAMP2	Timestamp when element is locked to the package
SRC-ELM-FROM-FILE-TYPE-FLG	"FROM" data set flag type ■ Blank—No "FROM" data set type ■ P—Data set is a path. Path and file names are contained in data areas. See SRC-ELM-FROM-PATH-AREA-OFFSET and SRC-ELM-FROM-NAME-AREA-OFFSET. ■ D—Data set is a DSN. See fields SRC-ELM-FROM-DSN and SRC-ELM-FROM-MEMBER.
SRC-ELM-TO-FILE-TYPE-FLG	TO data set flag type.
"TO" data set flag type ■ Blank—No "TO" data set type ■ P—Data set is a path. Path and file names are contained in data areas. See SRC-ELM-TO-PATH-AREA-OFFSET and SRC-ELM-TO-NAME-AREA-OFFSET. ■ D—Data set is a DSN. See fields SRC-ELM-RETRIEVE-TO-DSN and SRC-ELM-RETRIEVE-TO-MEMBER.	
SRC-ELM-FROM-PATH-AREA-OFFSET	When SRC-ELM-FROM-FILE-TYPE-FLG indicates a path, the field will contain the offset value to the "FROM" path name area from the beginning of SRC-ELEMENT-MASTER-INFO-BLOCK. The path area is composed of a 2 byte binary path name length, followed by the name.

Field	Description
SRC-ELM-FROM-NAME-AREA-OFFSET	When SRC-ELM-FROM-FILE-TYPE-FLG indicates a path, the field will contain the offset value to the "FROM" file name area from the beginning of SRC-ELEMENT-MASTER-INFO-BLOCK. The name area is composed of a 2 byte binary file name followed by the name.
SRC-ELM-TO-PATH-AREA-OFFSET	When SRC-ELM-TO-FILE-TYPE-FLG indicates a path. This field will contain the offset value to the "TO" path name area from the beginning of the SRC-ELEMENT-MASTER-INFO-BLOCK. The path name area is composed of 2 byte binary path name length value, followed by the name.
SRC-ELM-TO-NAME-AREA-OFFSET	When SRC-ELM-TO-FILE-TYPE-FLG indicates a path. This field will contain the offset value to the "TO" file name area from the beginning of SRC-ELEMENT-MASTER-INFO-BLOCK. The file name area is composed of 2 bytes binary file name length value, followed by the name.
SRC-ELM-NAME-AREA-OFFSET	Contains the offset to the element name areas from the beginning of SRC-ELEMENT-MASTER-INFO-BLOCK. The element area is composed of a 2 byte binary element name length value, followed by the name. This area contains the true name of the name of the element which can be up to 255 character.
SRC-ELM-AREA-BUFFER	This buffer area contains the space where all areas are contained.

3.7.3 TGT-ELEMENT-MASTER-INFO-BLOCK Field Descriptions

This section contains a brief description of the fields in the TGT-ELEMENT-MASTER-INFO-BLOCK control block layout. None of these fields are modifiable be exits.

Field	Description
TGT-ELM-LENGTH	Size of the Element Master Information block (TGT-ELEMENT-MASTER-INFO-BLOCK), in bytes.
TGT-ELM-CURRENT-VERSION	Version number to identify the Element Master information block (TGT-ELEMENT-MASTER-INFO-BLOCK). This should always be 5.
TGT-ELM-BLOCK-ID	Identifies this as the Element Master Information block. The value is always "$ELM"

Field	Description
TGT-ELM-FLAGS	Not used.
TGT-ELM-ELEMENT-VERSION	Version number of the element specified in the action request. This field has the same value as the TGT-ENV-ELEMENT-VERSION field in the Environment block.
TGT-ELM-ELEMENT-LEVEL	The current (highest) level of the element. This field has the same value as the TGT-ENV-ELEMENT-LEVEL field in the Environment block.
TGT-ELM-NUMBER-INSERTS	Count of statements added to the source by the last action that invoked source management.
TGT-ELM-NUMBER-DELETES	Count of statements deleted from the element source by the the last action that invoked source management.
TGT-ELM-ELEMENT-NAME	Name of the MCF element name. Element names longer than 10, or names containing special characters or mixed case characters will have a generated MCF element name. The actual name will be placed in the buffer area. See also TGT-ELM-NAME-AREA-OFFSET.
Name of the MCF element specified in the current action request.	
TGT-ELM-CURRENT-CCID	Last-specified CCID for the element.
TGT-ELM-BASE-LEVEL-NUMBER	The base level of the element
TGT-ELM-BASE-TOTAL-STMTS	Count of source statements in the base level of the element in the current stage.
TGT-ELM-BASE-DATE	Base date for the element.
TGT-ELM-BASE-TIME	Base time for the element.
TGT-ELM-BASE-COMMENT	Base comment for the element.
TGT-ELM-BASE-USERID	Base user ID for the element.
TGT-ELM-GEN-USERID	Generate processor user ID for the element.
TGT-ELM-GEN-DATE	Generate processor date for the element.
TGT-ELM-GEN-TIME	Generate processor time for the element.
TGT-ELM-FROM-DSN	Name of the data set from which the element was last added, updated, or restored.
TGT-ELM-FROM-MEMBER	Name of the member from which the element was last added, updated, or restored. Applicable if the source data set was a library; blank otherwise.

Field	Description
TGT-ELM-PROCESSOR-FLAG	Flag used to describe the last processor run for the element (and related information). This field may reference a delete processor (in the case where a delete processor was the last processor run for the current action). Values are: ■ Value: X'00' – Meaning: No processor has been run yet for the element. ■ Value: X'80' – Meaning: The last processor failed ■ Value: X'40' – Meaning: The last processor was a generate processor ■ Value: X'20' – Meaning: The last processor was a move processor ■ Value: X'10' – Meaning: The last processor was a delete processor ■ Value: X'08' – Meaning: The element has been restored but not yet processed. Therefore, processor statistics are unchanged from the from the archived statistics for the element.
TGT-ELM-PROCESSOR-LAST-DATE	Processor date for element.
TGT-ELM-PROCESSOR-LAST-TIME	Processor time for element.
TGT-ELM-PROCESSOR-USERID	Processor user ID for element.
TGT-ELM-PROCESSOR-NAME	(Element) name of the processor last run for the element.
TGT-ELM-PROCESSOR-RC	Processor return code for the element.
TGT-ELM-PROCESSOR-C1-RC	Endevor return code for the element. Values are; ■ Value: X'10' – Meaning: Processor had a system abend ■ Value: X'20' – Meaning: Processor had a user abend
TGT-ELM-PROCESSOR-LAST-COMMENT	Generate processor comment for the element.
TGT-ELM-LEVEL-COMMENT	Current-level comment for the element.
TGT-ELM-LEVEL-DATE	Current-level date for the element.

Field	Description
TGT-ELM-LEVEL-TIME	Current-level time for the element
TGT-ELM-LEVEL-USERID	Current-level user ID for the element.
TGT-ELM-LAST-ACTION	Most recent action executed for the element. Once executed, each action is recorded as the last action -- except ARCHIVE, LIST, PRINT, DISPLAY.
TGT-ELM-LAST-LEVEL-TOTAL	Count of source statements for the element, as of the current level.
TGT-ELM-MOVE-DATE	Date the element was last moved.
TGT-ELM-MOVE-TIME	Time the element was last moved.
TGT-ELM-MOVE-USERID	User ID associated with the last move of the element.
TGT-ELM-RETRIEVE-DATE	Date the element was last retrieved.
TGT-ELM-RETRIEVE-TIME	Time the element was last retrieved.
TGT-ELM-RETRIEVE-USERID	User ID associated with the last retrieve of the element.
TGT-ELM-RETRIEVE-COMMENT	Comment associated with the last retrieve of the element.
TGT-ELM-RETRIEVE-TO-DSN	Name of the data set to which the element was last retrieved.
TGT-ELM-RETRIEVE-TO-MEMBER	Name of the member to which the element was last retrieved. Applicable if the target data set for the last retrieve request was a library; blank otherwise.
TGT-ELM-RET-CCID	CCID associated with the last retrieve of the element.
TGT-ELM-GEN-CCID	CCID associated with the last generate of the element.
TGT-ELM-MOD-ACTION	Last action that modified the element. For example. an ADD or UPDATE would modify an element. A RETRIEVE would not.
TGT-ELM-ACTION-CCID	CCID associated with the last action that modified the element.
TGT-ELEM-ACTION-COMM	User ID of the user who performed the last action that modified an element.
TGT-ELM-ACTION-DATE	Date of last action that modified the element.
TGT-ELM-ACTION-TIME	Time when the last modifying action was performed.
TGT-ELM-SIGNOUT-ID	ID of the user to whom the modified element is.
TGT-ELM-LOC-FLAG	Flag used to indicate whether or not there is information in the following fields which describe the FROM information in a MOVE or TRANSFER action. For example, the value of X'80' means the flag is set to ON.
TGT-ELM-FR-ACTION	The last action, either MOVE, TRANSFER, or RESTORE.

Field	Description
TGT-ELM-FR-ACT-USERID	The user ID of the person performing the last MOVE, TRANSFER, or RETRIEVE.
TGT-ELM-FR-ACT-DATE	The date of the last MOVE, TRANSFER, or RETRIEVE.
TGT-ELM-FR-ACT-TIME	The time of the last MOVE, TRANSFER, or RETRIEVE.
TGT-ELM-FR-SITE	The site of the last MOVE, TRANSFER, or RETRIEVE.
TGT-ELM-FR-ENV	The environment of the last MOVE, TRANSFER, or RETRIEVE.
TGT-ELM-FR-SYS	The system of the last MOVE, TRANSFER, or RETRIEVE.
TGT-ELM-FR-SUBSYS	The subsystem of the last MOVE, TRANSFER, or RETRIEVE.
TGT-ELM-FR-ELEMENT	The name of the MCF element in the FROM location.
TGT-ELM-FR-TYPE	The type of the elemement in the from location
TGT-ELM-FR-STG-NBR	The stage number of the element in the FROM location.
TGT-ELM-FR-VER	The version of the element in the FROM location.
TGT-ELM-FR-LEVEL	The level of the element in the FROM location.
TGT-ELM-UPDATE-MEMBER-NAME	Element name in the delta library.
TGT-ELM-BASE-MEMBER-NAME	Element name in the base library.
TGT-ELM-LOWER-LEVEL	Internal use only.
TGT-ELM-PCT-PREV-INSERT-DEL	Percentage of previous inserts deleted.
TGT-ELM-PCT-PREV-DELS-REINS	Percentage of previous deletes reinserted.
TGT-ELM-ESD-FP-NAME-IF-OBJ	Name of footprinted object.
TGT-ELM-DELTA-FORMAT	Element delta format. ■ Value: F – Label: TGT-ELM-FORWARD-DELTA – Meaning: Forward format. ■ Value: R – Label: TGT-ELM-REVERSE-DELTA – Meaning: Reverse format.
TGT-ELM-PACKED-INDICATOR	Element base is not compressed when the value is Y. Y -- TGT-ELM-BASE-NOT-PACKED.
TGT-ELM-LAST-PROC-VERSION	Version number of last processor.
TGT-ELM-LAST-PROC-LEVEL	Level number of last processor.

Field	Description
TGT-ELM-RECORD-FORMAT	ID of record format. For release 3.6 and above, all elements returned should have a value of x'36' in this field. ■ x'21' -- Release 2.1 format ID. ■ x'25' -- Release 2.5 format ID. ■ x'35' -- Release 3.5 format ID. ■ x'36' -- Release 3.6 format ID. ■ x'36' -- Current record format ID.
TGT-ELM-CONF-DELTA-MEM-ID	Component list delta member name.
TGT-ELM-CONF-DELTA-VERSION	Component list delta member version number.
TGT-ELM-CONF-BASE-TOTAL	Component list base total
TGT-ELM-CONF-LAST-LVL-TOTAL	Component list last level total.
TGT-ELM-CONF-LAST-LVL-NUMBER	Component list last level total.
TGT-ELM-INSERTS-LAST-LEVEL	Inserts at last level
TGT-ELM-DELETES-LAST-LEVEL	Deletes at last level
TGT-ELM-CONF-REGRES-INS-PCT	Component list regression insert percent.
TGT-ELM-CONF-REGRES-DEL-PCT	Component list regression deletes percent.
TGT-ELM-CONF-BASE-DATE	Component list base date.
TGT-ELM-CONF-BASE-TIME	Component list base time.
TGT-ELM-CONF-LAST-LVL-TIME	Component list last level time.
TGT-ELM-CONF-LAST-LVL-DATE	Component list last level date.
TGT-ELM-CONF-COMP-STAT-FLAG	Component list flag. ■ Value: X'80' – Meaning: Component list built using MONITOR=COMPONENT statement in processor. ■ Value: X'40' – Meaning: Component list built by copy or restore.
TGT-ELM-CONF-DELTA-FORMAT	Component list delta format. Y -- TGT-ELM-CONF-BASE-IN-DELTA Comp base in Delta library.
TGT-ELM-LAST-PROC-PACKAGE	ID of last package processed.
TGT-ELM-PACKAGE-TIMESTAMP1	Time stamp on last processed package.
TGT-ELM-PACKAGE-TIMESTAMP2	Time stamp on last processed package.

Field	Description
TGT-ELM-PACKAGE-WITH-OUTPUT	Package associated with element outputs.
TGT-ELM-OPACKAGE-TIMESTAMP1	Time stamp on package outputs.
TGT-ELM-OPACKAGE-TIMESTAMP2	Time stamp on package outputs.
TGT-ELM-PROCESSOR-GROUP	Processor group name.
TGT-ELM-FROM-FILE-TYPE-FLG	"FROM" data set flag type ■ Blank—No "FROM" data set type ■ P—Data set is a path. Path and file names are contained in data areas. See TGT-ELM-FROM-PATH-AREA-OFFSET and TGT-ELM-FROM-NAME-AREA-OFFSET. ■ D—Data set is a DSN. See fields TGT-ELM-FROM-DSN and TGT-ELM-FROM-MEMBER.
TGT-ELM-TO-FILE-TYPE-FLG	"TO" data set flag type. ■ Blank—No "TO" data set type ■ P—Data set is a path. Path and file names are contained in data areas. See TGT-ELM-TO-PATH-AREA-OFFSET and TGT-ELM-TO-NAME-AREA-OFFSET. ■ D—Data set is a DSN. See fields TGT-ELM-RETRIEVE-TO-DSN and TGT-ELM-RETRIEVE-TO-MEMBER.
TGT-ELM-FROM-PATH-AREA-OFFSET	When TGT-ELM-FROM-FILE-TYPE-FLG indicates a path, the offset value to the "FROM" path name area from the beginning of the TGT-ELM-FROM-INFO-BLOCK. The path area is composed of 2 byte binary path name length value, followed by the name.
TGT-ELM-FROM-NAME-AREA-OFFSET	When TGT-ELM-FROM-FILE-TYPE-FLG indicates a path, the offset value to the "FROM" path name area from the beginning of the TGT-ELMENT-MASTER-INFO-BLOCK. The name area is composed of 2 byte binary path name length value, followed by the name.
TGT-ELM-TO-PATH-AREA-OFFSET	When TGT-ELM-TO-FILE-TYPE-FLG indicates a path. This field will contain the offset value to the "TO" path name area from the beginning of the TGT-ELMENT-MASTER-INFO-BLOCK. The path name is composed of 2 byte binary path name length value, followed by the name.

Field	Description
TGT-ELM-TO-NAME-AREA-OFFSET	When TGT-ELM-TO-FILE-TYPE-FLG indicates a path file. This field will contain the offset value to the "TO" file name area from the beginning of the TGT-ELMENT-MASTER-INFO-BLOCK. The file name is composed of 2 byte binary path name length value, followed by the name.
TGT-ELM-NAME-AREA-OFFSET	Contains the offset to the element name areas from the beginning of TGT-ELEMENT-MASTER-INFO-BLOCK. The elment area is composed of a 2 byte binary element name length value, followed by the name. This area contains the true name of the element which can be up to 255 characters. The file name is composed of 2 byte binary path name length value, followed by the name.
TGT-ELM-AREA-BUFFER	This buffer area contains the space where all areas are contained.

3.8 Assembler File Block

Endevor passes this block to programs written for exits 2-4. It describes the external file used by the current action request: either the source file for an ADD, UPDATE, or RESTORE request, or the target file for an ARCHIVE or RETRIEVE request. It is pointed to by each Environment block that describes a location external to Endevor (ENVFCBA field).

3.8.1 $FILDS DSECT sample

```
FILLEN    DC    AL2(FILSIZE)       LENGTH OF THE BLOCK                 00160000
FILVERS   DC    AL2(FIL$VERS)      VERSION OF THE BLOCK                00170000
FIL$VERS  EQU   2                      VERSION 2                       00180001
FILID     DC    CL4'$FIL'          BLOCK ID                            00190000
FILDSN    DS    CL44' '            DATASET NAME                        00200000
FILDSMEM  DS    CL10' '            MEMBER NAME                         00210000
FILDDN    DS    CL8                DDNAME                              00211000
FILDSTY   DS    CL3' '             DATASET TYPE( DA,PSU,PO,IS,...)     00220000
*                                      FOR HFS (H@,H$,H/,HW)           00230000
          SPACE 1                                                      00231000
*                                                                      00232000
*   OFFSET (FROM BEGINNING OF FILDS) TO THE PATH AND FILE NAME AREAS.  00233000
*   THE FIRST 2 BYTES IN AREA = THE LENGTH OF THE NAME, THIS IS THEN   00234000
*   FOLLOWED BY THE NAME.                                              00235000
*                                                                      00236000
FILPAOFF  DS    H                  PATH AREA OFFSET, IF HFS            00240000
FILNAOFF  DS    H                  FILE NAME AREA OFFSET, IF HFS       00241000
          DC    CL21' '            ** RESERVED **                      00250000
FILBFAREA DS    XL1027             AREA BUFFER SPACE                   00260000
FILSIZE   EQU   *-$FILDS           LENGTH OF THE $FILDS BLOCK          00540000
         MEND                                                          00550000
```

3.8.2 $FILDS DSECT Field Descriptions

This section contains brief descriptions of the fields in the $FILDS DSECT. Unless otherwise noted, exit programs cannot modify these fields.

Field	Description
FILLEN	Size of the File block ($FILDS), in bytes.
FILVERS	Version number to identify the File block ($FILDS). This should always be 2. Use the FIL$VERS label to verify that the correct version of this DSECT was passed to your exit.
FILID	Identifies this as the File Control block. Value is always "$FIL".
FILDSN	Modifiable by exit 4 programs. Data set name of the external file used by the current action.
FILDSMEM	Modifiable by exit 4. Member name of the element for which the action applies. Applicable when the data set is a library; blank otherwise.

Field	Description
FILDDN	Modifiable by exit 4. DDname associated with the data set named above. If you change this field using exit 4, it is your responsibility to allocate the file (data set name) associated with the new DDname. In order for Endevor to recognize the change(s), you must return a code of 4 ($ECBDS field ECBRTCD). You do not have to change the FILDSN, FILDSMEM, and/or FILDSTY fields to be correct for the new DDname.
FILDSTY	Modifiable by exit 4. Type of data set named by the FILDSN field: DA, PSU, PO, IS, and so forth. The possible values are the same as those that apply for the DSORG JCL parameter.
FILPAOFF	When the data set type is HFS, it contains the offset from $FILDS to the path name area within the buffer space, FILBFAREA. The first 2 bytes of the path area contain the binary length value of path name. The name immediately follows.
FILNAOFF	When the data set type is HFS, it contains the offset from $FILDS to the file name area within the buffer space, see FILBFAREA The first 2 bytes of the area contain the binary length value of file name. The name immediately follows.
FILBFAREA	Buffer which contains the path and file areas.
FILSIZE	Size of the $FILDS DSECT.

3.9 COBOL File Block

Endevor passes this block to programs written for exits 2-4. It describes the external file used by the current action request: either the source file for an ADD, UPDATE, or RESTORE request, or the target file for an ARCHIVE or RETRIEVE request. It is pointed to by each Environment block that describes a location external to Endevor (field SRC-ENV-FILE-POINTER AND TGT-ENV-FILE-POINTER).

3.9.1 XXX-FILE-CONTROL-BLOCK Control block layout sample

```
01  SRC-FILE-CONTROL-BLOCK.                                      01840010
    05  SRC-FIL-LENGTH            PIC 9(4) COMP SYNC.            01850000
    05  SRC-FIL-CURRENT-VERSION   PIC 9(4) COMP SYNC.            01860000
    05  SRC-FIL-BLOCK-ID          PIC X(4).                      01870010
    05  SRC-FIL-DATASET-NAME      PIC X(44).                     01880010
    05  SRC-FIL-DATASET-MEMBER    PIC X(10).                     01890010
    05  SRC-FIL-DDNAME            PIC X(8).                      01900010
    05  SRC-FIL-DATASET-TYPE      PIC X(3).                      01910010
    05  SRC-FIL-PATH-AREA-OFFSET  PIC S9(4) COMP SYNC.
    05  SRC-FIL-NAME-AREA-OFFSET  PIC S9(4) COMP SYNC.
    05  FILLER                    PIC X(21).                     01920006
    05  SRC-FIL-AREA-BUFFER       PIC X(1027).

01 TGT-FILE-CONTROL-BLOCK

    05  TGT-FIL-LENGTH          PIC 9(4) COMP SYNC.              02860000
    05  TGT-FIL-CURRENT-VERSION PIC 9(4) COMP SYNC.              02870000
    05  TGT-FIL-BLOCK-ID        PIC X(4).                        02880010
    05  TGT-FIL-DATASET-NAME    PIC X(44).                       02890010
    05  TGT-FIL-DATASET-MEMBER  PIC X(10).                       02900010
    05  TGT-FIL-DDNAME          PIC X(8).                        02910010
    05  TGT-FIL-DATASET-TYPE    PIC X(3).                        02920010
    05  TGT-FIL-PATH-AREA-OFFSET  PIC S9(4) COMP SYNC.
    05  TGT-FIL-NAME-AREA-OFFSET  PIC S9(4) COMP SYNC.
    05  FILLER                    PIC X(21).                     01920006
    05  TGT-FIL-AREA-BUFFER       PIC X(1027).
```

3.9.2 SRC-FILE-CONTROL-BLOCK Field Descriptions

This section contains brief descriptions of the fields in the SRC-FILE-CONTROL-BLOCK. Unless otherwise noted, exit programs cannot modify these fields.

Field	Description
SRC-FIL-LENGTH	Size of the File block (SRC-FILE-CONTROL-BLOCK), in bytes.

Field	Description
SRC-FIL-CURRENT-VERSION	Version number to identify the File block (SRC-FILE-CONTROL-BLOCK). This should always be 2. Use the label to verify that the correct version of this DSECT was passed to your exit.
SRC-FIL-BLOCK-ID	Identifies this as the File Control block. Value is always "$FIL".
SRC-FIL-DATASET-NAME	Modifiable by exit 4 programs. Data set name of the external file used by the current action.
SRC-FIL-DATASET-MEMBER	Modifiable by exit 4. Member name of the element for which the action applies. Applicable when the data set is a library; blank otherwise.
SRC-FIL-DDNAME	Modifiable by exit 4. DDname associated with the data set named above. If you change this field using exit 4, it is your responsibility to allocate the file (data set name) associated with the new DDname. In order for Endevor to recognize the change(s), you must pass a return code of 4 (EXIT-CONTROL-BLOCK field ECB-RETURN-CODE). You do not have to change the SRC-FIL-DATASET-NAME, SRC-FIL-DATASET-MEMBER, or SRC-FIL-DATASET-TYPE to be correct for the new DDname.
SRC-FIL-DATASET-TYPE	Modifiable by exit 4. Type of data set named by the SRC-FIL-DATASET-NAME field: DA, PSU, PO, IS, and so forth. The possible values are the same as those that apply for the DSORG JCL parameter.
SRC-FIL-PATH-AREA-OFFSET	When the data set type is HFS, it contains the offset from SRC-FILE-CONTROL-BLOCK to the path name area within the buffer space, SRC-FIL-AREA-BUFFER. The first 2 bytes of the path area contain the binary length value of path name. The name immediately follows.
SRC-FIL-NAME-AREA-OFFSET	When the data set type is HFS, it contains the offset from SRC-FILE-CONTROL-BLOCK to the file name area within the buffer space, see SRC-FIL-AREA-BUFFER. The first 2 bytes of the area contain the binary length value of file name and the name immediately follows.
SRC-FIL-AREA-BUFFER	Buffer which contains the path and file areas.

3.9.3 TGT-FILE-CONTROL-BLOCK Field Descriptions

This section contains brief descriptions of the fields in the TGT-FILE-CONTROL-BLOCK and unless otherwise noted, exit programs cannot modify these fields.

Field	Description
TGT-FIL-LENGTH	Size of the file block (TGT-FILE-CONTROL-BLOCK), in bytes.
TGT-FIL-CURRENT-VERSION	Version number to identify the File block (TGT-FILE-CONTROL-BLOCK). This should always be 2. Use the label to verify that the correct version of this DSECT was passed to your exit.
TGT-FIL-BLOCK-ID	Identifies this as the Exit Control block. Value is always "$FIL".
TGT-FIL-DATASET-NAME	Modifiable by exit 4 programs. Data set name of the external file used by the current action.
TGT-FIL-DATASET-MEMBER	Modifiable by exit 4. Member name of the element for which the action applies. Applicable when the data set is a library; blank otherwise.
TGT-FIL-DDNAME	Modifiable by exit 4. DDNAME associated with the data set named above. If you change the field using exit 4, it is your responsibility to allocate the file (data set name) associated with the new DDname. In order for Endevor to recognize the change(s), you must return a code of 4 (EXIT-CONTROL-BLOCK field ECB-RETURN-CODE). You do not have to change the TGT-FIL-DATASET-MEMBER, or the TGT-FIL-DATASET-TYPE to be correct for the new DDname.
TGT-FIL-DATASET-TYPE	Modifiable by exit 4. Type of data set named by the TGT-FIL-DATASET-NAME field: DA, PSU, PO, IS, and so forth. The possible values are the same as those that apply for the DSORG JCL parameter.
TGT-FIL-PATH-AREA-OFFSET	When the data set type is HFS, it contains the offset from TGT-FILE-CONTROL-BLOCK to the path name area within the buffer space, TGT-FIL-AREA-BUFFER. The first 2 bytes of the path area contain the binary length value of path name. The name immediately follows.
TGT-FIL-NAME-AREA-OFFSET	When the data set type is HFS, it contains the offset from TGT-FILE-CONTROL-BLOCK to the file name area within the buffer space, see TGT-FIL-AREA-BUFFER. The first 2 bytes of the area contain the binary length value of file name and the name immediately follows.
TGT-FIL-AREA-BUFFER	Buffer which contains the path and file areas.

3.10 SMF Action Record

3.10.1 Overview

Endevor passes this block to programs written for exit 3, when SMF recording is in effect. It is used to log/record information using SMF, and/or to inhibit the writing of the SMF Action Request record, if desired. The address list passed to the exit is pointed to by Register 1: 8(R1).

For the assembler DSECT layout and field descriptions, see "SMF Recording," in the *Administration Guide*.

Chapter 4. Package Exits

This chapter describes the exits that are used with package processing. The sections include:

- Overview
- Writing Exit Programs for Package Actions
- Package Exit Points
- Enabling Exit Points
- Requests for Data
- Description of Package Exit Actions

4.1 Overview

4.1.1 Package Processing

Package processing consists of several actions that are used throughout the life cycle of the package. An example of a package action is CREATE or CAST. The package exit facility provides an exit point before and after most package actions. Exceptions are noted in the text. You must include exit programs for package actions in the user exit table just like the other Endevor exits.

Package exit points differ from other user exit points in several important ways:

- Standard Endevor exits have a separate number per exit point. Package exit points all share one number--seven (7).
- There is a separate exit point reserved for package exit setup. After building the user exit table during initialization, Endevor calls all programs identified as exit 7 programs, indicating that Setup is in progress. The Setup call allows users to enable just those package exit points that their exit 7 programs require. During the session, Endevor only invokes exit processing at those package exit points enabled by the setup call to each exit 7 program.
- Package exit programs can request data from Endevor after they are called. This information is returned to the exit program in a control block. There can be multiple requests for a given exit point.
- Package exits written in high-level languages do *not* need to follow the conventions for user exits that are discussed in the section High-Level Language Support.

4.2 Writing Exit Programs for Package Actions

Exit programs for package actions must:

- Determine which package function has invoked exit processing. This information is provided in the main exit control block ($PECBDS in assembler) and (PACKAGE-EXIT-BLOCK in COBOL). See the section 4.3, “Package Exit Points” on page 4-4 that follows for a list of the functions and subfunctions supported by the package exit facility.
- Include a routine called Setup. Endevor reads the control block returned by the exit 7 program during the setup call to determine which package exit function(s) the exit enables. For more information on Setup, see 4.4, “Enabling Exit Points” on page 4-5

Note: Your program must recognize the Setup call. Endevor calls all exit modules within the setup function. If your program does not change the necessary control block fields from **N** to **Y**, Endevor does not call it again.

- Include request flag and return code logic if you wish to obtain certain information from Endevor after the exit is called. For more information, see 4.5.3, “Assembler Request Flags and Return Codes” on page 4-13 and 4.5.4, “COBOL Request Flags and Return Codes” on page 4-14.
- Not assume that you are executing in an ISPF environment. The Endevor Batch Package Facility does not provide ISPF information to the exits. Any exit that relies on ISPF specific fields will have to be modified to execute in a batch environment. For information on executing packages in batch and the Endevor Batch Package Facility see the *SCL Reference Guide*.

  Note: Applicable only for COBOL:

  Do not link the utility EPC1UEXT into high-level language exit load modules as you would for a user exit.

Otherwise, when writing exit programs for package functions follow the guidelines provided for standard exit programs in the section Writing and Storing Exits.

4.3 Package Exit Points

4.3.1 With Package Actions and Functions

Endevor supports exit programs before and after the following package actions and functions:

- Archive.
- Backin.
- Backout.
- Cast (before-, after-, -mid-).
- Commit.
- Create, and its related functions Build, Copy, Edit, Import, and Define.
- Delete.
- Display (before- only), and its related functions Approver, Backout, SCL, Element Summary, Packages, Reports.
- Execute.
- Export.
- GENPKGID.
- List, and its related functions Build, Cast, Review, Execute, Ship, Backout, Commit, Utilities, and Display.
- Modify, and its related functions Build, Copy, Edit, Import.
- Reset.
- Review, and its related functions Approve, Deny.
- Ship, and its related functions Transmit, Confirm.

For more information about writing exit programs for each of these functions and subfunctions, see 4.6, “Description of Package Exit Actions” on page 4-18

4.4 Enabling Exit Points

4.4.1 Assembler Exit Points

You enable package exit points by setting flags in the Package Exit Control block ($PECBDS) during the Setup call. An exit program receives control once during the Setup call if its name was specified as an "@C1UEXIT EXIT#=7" entry in the user exit table (see 1.2, “Identifying Exits to Endevor” on page 1-16). The Setup call occurs only once per session during Endevor initialization. Setup information is stored for the life of the Endevor session. During the Setup call the exit program must:

- Indicate whether the exit program can operate under TSO and/or as a batch program (without TSO services). The default for TSO is **Y**(Yes). You can set the default for batch with the @C1UEXIT macro. For more information, see Chapter 1, Chapter 1, “Using Exits with Endevor” on page 1-1
- Specify which of the exit points the exit program is designed to handle.

Endevor extracts the Package Exit Control block ($PECBDS) values from all defined exit 7 programs at initialization, using them to generate a table of enabled package exits. Endevor uses this table during the session to determine whether to invoke an exit program at a particular point.

When Endevor passes the $PECBDS control block during the Setup call, it loads the following values:

- PECBPKID CL16 'INITIAL'
- PECBFNNM CL8 'SETUP'
- PECBFNCD DC H'0000'

The user exit must indicate to Endevor the operating environments in which the user exit is designed to execute by setting the proper $PECBDS field to **Y** or **N.**

4.4.1.1 $PECBDS Fields

To indicate the ability to operate in the following environment(s):	Set the following field(s) to Y or N:
TSO (including ISPF).	PECBTSOX
Batch (with no TSO or ISPF services available)	PECBBATX

The user exit must also indicate to Endevor which exit points it can handle by setting the proper $PECBDS fields to **Y** (enable):

To enable this exit point	Set this field to Y
Before-archive	PECBARBE
After-archive	PECBARAF
Before-backin	PECBBIBE
After-backin	PECBBIAF
Before-backout	PECBBOBE
After-backout	PECBBOAF
Before-cast	PECBCABE
Mid-cast	PECBCAMD
After-cast	PECBCAAF
Before-collect	PECBCSBE
After-collect	PECBCSAF
Before-commit	PECBCOBE
After-commit	PECBCOAF
Before-create/build	PECBCBBE
After-create/build	PECBCBAF
Before-create/copy	PECBCCBE
After-create/copy	PECBCCAF
Before-create/edit	PECBCEBE
After-create/edit	PECBCEAF
Before-create/import	PECBCIBE
After-create/import	PECBCIAF
Before-delete	PECBDEBE
After-delete	PECBDEAF
Before-display approver	PECBDABE
Before-display backout	PECBDABE
Before-display SCL	PECBDSBE
Before-display element summary	PECBDXBE
Before-display packages	PECBDPBE
Before-display reports	PECBDRBE
Before-execute	PECBEXBE
After-execute	PECBEXAF

To enable this exit point	Set this field to Y
Before-export	PECBEPBE
After-export	PECBEPAF
Before-GENPKGID	PECBGPBE
After-GENPKGID	PECBGPAF
Before-list	PECBLTBE
After-list	PECBLTAF
Before-modify/build	PECBMBBE
After-modify/build	PECBMBAF
Before-modify/copy	PECBMCBE
After-modify/copy	PECBMCAF
Before-modify/edit	PECBMEBE
After-modify/edit	PECBMEAF
Before-modify/import	PECBMIBE
After-modify/import	PECBMIAF
Before-reset	PECBRSBE
After-reset	PECBRSAF
Before-review/approve	PECBRABE
After-review/approve	PECBRAAF
Before-review/deny	PECBRDBE
After-review/deny	PECBRDAF
Before-ship/transmit	PECBSXBE
Before-ship/confirm	PECBSCBE
After-ship/transmit	PECBSXAF
After-ship/confirm	PECBSCAF

For further details on each field, see Chapter 6, “Package Exit Control Blocks” on page 6-1.

4.4.1.2 Sample SETUP Routine

The setup routine below enables four exit points.

```
SETUP    DS    0H
*
         MVI   PECBTSOX,C'Y'     ENABLE     ENABLED FOR TSO EXECUTION
         MVI   PECBBATX,C'Y'     ENABLE     ENABLED FOR BATCH EXECUTION
         MVI   PECBBOAF,C'Y'     ENABLE     AFTER BACKOUT EXIT
         MVI   PECBCAAF,C'Y'     ENABLE     AFTER CAST   EXIT
         MVI   PECBEXAF,C'Y'     ENABLE     AFTER EXECUTE EXIT
         MVI   PECBRSBE,C'Y'     ENABLE     BEFORE RESET EXIT
*
         BR    R14
```

4.4.2 COBOL Exit Points

You enable package exit points by setting flags in the Package Exit Control block (PACKAGE-EXIT-BLOCK) during the Setup call. An exit program receives control once during the Setup call if its name was specified as an "@C1UEXIT EXIT#=7" entry in the user exit table (see 1.2, “Identifying Exits to Endevor” on page 1-16). The Setup call occurs only once per session during Endevor initialization. Setup information is stored for the life of the Endevor session. During the Setup call the exit program must:

- Indicate whether the exit program can operate under TSO and/or as a batch program (without TSO services). The default for TSO is **Y**(Yes). You can set the default for batch with the @C1UEXIT macro. For more information, see Chapter 1, Chapter 1, “Using Exits with Endevor” on page 1-1
- Specify which of the exit points the exit program is designed to handle.

Endevor extracts the Package Exit Control block (PACKAGE-EXIT-BLOCK) values from all defined exit 7 programs at initialization, using them to generate a table of enabled package exits. Endevor uses this table during the session to determine whether to invoke an exit program at a particular point.

When Endevor passes the PACKAGE-EXIT-BLOCK control block during the Setup call, it loads the following values:

- PECB-PACKAGE-ID PIC X(16).
- PECB-FUNCTIONAL-LITERAL PIC X(8).
- PECB-FUNCTIONAL-CODE PIC 9(4) COMP SYNC.

The user exit must indicate to Endevor the operating environments in which the user exit is designed to execute by setting the proper PACKAGE-EXIT-BLOCK field to Y or N.

4.4.2.1 PACKAGE-EXIT-BLOCK Fields

To indicate the ability to operate in the following environment(s):	Set the following field(s) to Y or N:
TSO (including ISPF).	PECB-TSO-EXECUTE
Batch (with no TSO or ISPF services available)	PECB-BATCH-EXECUTE

The user exit must also indicate to Endevor which exit points it can handle by setting the proper PACKAGE-EXIT-BLOCK fields to Y (enable):

To enable this exit point	Set this field to Y
Before-archive	PECB-BEFORE-ARCHIVE
After-archive	PECB-AFTER-ARCHIVE
Before-backin	PECB-BEFORE-BACKIN
After-backin	PECB-AFTER-BACKIN
Before-backout	PECB-BEFORE-BACKOUT
After-backout	PECB-AFTER-BACKOUT
Before-cast	PECB-BEFORE-CAST
Mid-cast	PECB-MID-CAST
After-cast	PECB-AFTER-CAST
Before-collect	PECB-BEFORE-COLLECT
After-collect	PECB-AFTER-COLLECT
Before-commit	PECB-BEFORE-COMMIT
After-commit	PECB-AFTER-COMMIT
Before-create/build	PECB-BEFOR-CREATE-BLD
After-create/build	PECB-AFTER-CREATE-BLD
Before-create/copy	PECB-BEFORE-CREATE-COPY
After-create/copy	PECB-AFTER-CREATE-COPY
Before-create/edit	PECB-BEFORE-CREATE-EDIT
After-create/edit	PECB-AFTER-CREATE-EDIT
Before-create/import	PECB-BEFORE-CREATE-IMPT
After-create/import	PECB-AFTER-CREATE-IMPT
Before-delete	PECB-BEFORE-DELETE

To enable this exit point	Set this field to Y
After-delete	PECB-AFTER-DELETE
Before-display approver	PECB-BEFORE-DSPLY-APPR
Before-display backout	PECB-BEFORE-DSPLY-BKOUT
Before-display SCL	PECB-BEFORE-DSPLY-SCL
Before-display element summary	PECB-BEFORE-DSPLY-ELMSM
Before-display packages	PECB-BEFORE-DSPLY-PKG
Before-display reports	PECB-BEFORE-DSPLY-RPT
Before-execute	PECB-BEFORE-EXEC
After-execute	PECB-AFTER-EXEC
Before-export	PECB-BEFORE-EXPORT
After-export	PECB-AFTER-EXPORT
Before-GENPKGID	PECB-BEFORE-GENPID
After-GENPKGID	PECB-AFTER-GENPID
Before-list	PECB-BEFORE-LIST
After-list	PECB-AFTER-LIST
Before-modify/build	PECB-BEFORE-MOD-BLD
After-modify/build	PECB-AFTER-MOD-BLD
Before-modify/copy	PECB-BEFORE-MOD-CPY
After-modify/copy	PECB-AFTER-MOD-CPY
Before-modify/edit	PECB-BEFORE-MOD-EDIT
After-modify/edit	PECB-AFTER-MOD-EDIT
Before-modify/import	PECB-BEFORE-MOD-IMPT
After-modify/import	PECB-AFTER-MOD-IMPT
Before-reset	PECB-BEFORE-RESET
After-reset	PECB-AFTER-RESET
Before-review/approve	PECB-BEFORE-REVAPP
After-review/approve	PECB-AFTER-REV-APP
Before-review/deny	PECB-BEFORE-REV-DENY
After-review/deny	PECB-AFTER-REV-DENY
Before-ship/transmit	PECB-BEFORE-SHIP-XMIT
Before-ship/confirm	PECB-BEFORE-SHIP-XMIT

To enable this exit point	Set this field to Y
After-ship/transmit	PECB-AFTER-SHIP-XMIT
Before-ship/confirm	PECB-BEFORE-SHIP-CON
After-ship/confirm	PECB-AFTER-SHIP-CON

For further details on each field, see Chapter 6, “Package Exit Control Blocks” on page 6-1

4.4.2.2 Sample SETUP Routine

The setup routine below enables four exit points.

```
PROCEDURE DIVISION USING
        PACKAGE-EXIT-BLOCK
        PACKAGE-REQUEST-BLOCK
        PACKAGE-EXIT-HEADER-BLOCK
        PACKAGE-EXIT-FILE-BLOCK
        PACKAGE-EXIT-ACTION-BLOCK
        PACKAGE-EXIT-APPROVER-MAP
        PACKAGE-EXIT-BACKOUT-BLOCK
        PACKAGE-EXIT-SHIPMENT-BLOCK
        PACKAGE-EXIT-SCL-BLOCK.
    IF  SETUP-EXIT-OPTIONS
        MOVE 'Y'   TO PECB-AFTER-BACKOUT
        MOVE 'Y'   TO PECB-AFTER-CAST
        MOVE 'Y'   TO PECB-AFTER-EXEC
        MOVE 'Y'   TO PECB-BEFORE-RESET
        MOVE ZEROS TO RETURN-CODE
        GO TO 800-SET-RETURN.
```

4.5 Requests for Data

4.5.1 Record-Level Interface

Throughout the life cycle of a package there are groups of records that are created, updated, and deleted as a function of package processing. They may or may not be in existence at certain points of the process. Also, they are variable in number and existence per package. Due to the variable aspects of such data, the package exit facility was designed with a record-level interface that allows the user exit to drive its own resource requirements.

Note: This process is very different from the other Endevor exits in that it allows for multiple calls to a single exit module.

The following types of variable data are available to user exits:

- Element ID records
- Approver group records
- Backout records
- SCL records
- Shipment records
- Collection records

The user controls which and how much of a particular type of resource is passed to a program, thereby controlling how much storage and processing the user exit requires. The request process is managed through a series of request for data flags and request return codes.

4.5.2 How Requests for Data Work

The user exit fills in a request for a certain type of data. Assuming that the type of data is available at that exit point, the exit driver builds a control block, clears the USER REQUEST field, sets the request for data return code with the appropriate value, and returns either to Endevor (if an error occurs) or to the user exit. This interactive process continues per request type until one of the following conditions occurs:

- The user exit returns to Endevor without resetting the request for data flag.
- Endevor encounters an EOF on the request.
- Endevor encounters a NOT FOUND condition on the request.
- An error occurs.

The package exit facility processes these requests for data serially per exit module called. So if you have one exit 7 module, the exit program has one chance to process each of the available data types.

The following sections detail return codes and request for data flags, availability considerations, and the hierarchical relationship between standard exit return codes, updating Endevor fields, message processing, and REQUEST FOR DATA fields.

4.5.3 Assembler Request Flags and Return Codes

There are six fields in the $PECBDS associated with request for data processing: five request flags (one per type) and one return code field. The five request flags are used by the user exit to tell Endevor that there is a request for a record to be sent. These fields are defined below (note the default value is 'N').

Field	Description
PECBACTR DC C'N'	Request for element ID record.
PECBAPPR DC C'N'	Request for approver group record.
PECBBODR DC C'N'	Request for backout record.
PECBSCLR DC C'N'	Request for SCL record.
PECBSHPR DC C'N'	Request for shipment record.
PECBCOLR DC C'N'	Request for collection record.

After processing the request for data, Endevor sets a return code for the user exit to interrogate.

Field	Description
PECB-REQ-REQUEST-RETURNCODE	Request for data return code. This field is modified by CA-Endevor to inform the user exit of the request for data results. ■ Equates: PECB-SUCCESSFUL-RECORD-SENT – Value: 1 ■ Equates: PECB-END-OF-FILE-FOR-REC-TYPE – Value : 2 ■ Equates: PECB-NO-RECORDS-FOUND – Value: 3 ■ Equates: PECB-SEVERE-ERROR – Value: 12

When an EOF or NFND condition occurs, CA-Endevor returns this information to the user exit, but does not allow further requests against that data type.

4.5.4 COBOL Request Flags and Return Codes

There are six fields in the PACKAGE-EXIT-BLOCK associated with request for data processing:

five request flags (one per type) and one return code field. The five request flags are used by the user exit to tell Endevor that there is a request for a record to be sent. These fields are defined below (note the default value is 'N').

Field	Description
PECB-REQ-ACTION-ID-REC	Request for element ID record.
PECB-REQ-APPROVER-REC	Request for approver group record.
PECB-REQ-BACKOUT-REC	Request for backout record.
PECB-REQ-SCL-RECORDS	Request for SCL record.
PECB-REQ-SHIP-RECORDS	Request for shipment record.

After processing the request for data, Endevor sets a return code for the user exit to interrogate.

Field	Description
PECB-REQ-REQUEST-RETURNCODE	Request for data return code. This field is modified by CA-Endevor to inform the user exit of the request for data results. ■ Equates: PECB-SUCCESSFUL-RECORD-SENT – Value: 1 ■ Equates: PECB-END-OF-FILE-FOR-REC-TYPE – Value : 2 ■ Equates: PECB-NO-RECORDS-FOUND – Value: 3 ■ Equates: PECB-SEVERE-ERROR – Value: 12 When an EOF or NFND condition occurs, CA-Endevor returns this information to the user exit, but does not allow further requests against that data type.

4.5.5 Availability of Data Records

Five types of records may be available through the request for data facility. Not all records are available at all times. The following section, 4.6, “Description of Package Exit Actions” on page 4-18, outlines which request blocks are available per exit point. Use this as a reference while coding exit programs.

When an exit program requests a type of data that is not valid at the exit point, Endevor issues an error message and the function abends. There is a distinction between an invalid data request and a not found condition:

- An invalid request is a request for a record type that never exists at the exit point.
- A not found condition occurs when the data type may exist, but the particular package doesn't use that type of record.

4.5.5.1 Assembler - Availability of Data Records

To assist exit programs in the request for data process, Endevor sets three flags in $PECBDS to inform exit programs which types of data exist. These field names are PECBACTE, PECBAPPE, and PECBBODE. These flags are initialized on the first call.

The timing of the request for data is also regulated. The five types of record requests are processed sequentially. Each valid request type is processed once per exit module. The record types are processed in the following order:

1. PECBACTR (element ID records)
2. PECBAPPR (approver records)
3. PECBBODR (backout records)
4. PECBSCLR (SCL records)
5. PECBSHPR (shipment records)
6. PECBCOLR (collection records)

Once a valid request type has been processed or bypassed (per user exit) the flag is not checked again. If the exit program does not set any request for data flags the first time it returns to Endevor, Endevor does not reinvoke the exit program.

Note: Endevor sets the PECBRQRC field to 2 for EOF conditions, and to 3 for Not Found (NFND) conditions. It returns once to the exit program with this information. Make sure that your exit programs detect and act on an EOF or NFND condition, because, if the exit attempts to request more data after the EOF or NFND has been set, an error occurs.

4.5.5.2 COBOL - Availability of Data Records

To assist exit programs in the request for data process, Endevor sets three flags in PACKAGE-EXIT-BLOCKS to inform exit programs which types of data exist. These field names are PECB-ACT-REC-EXIST-FLAG, PECB-APP-REC-EXIST-FLAG, and PECB-BAC-REC-EXIST-FLAG. These flags are initialized on the first call.

The timing of the request for data is also regulated. The five types of record requests are processed sequentially. Each valid request type is processed once per exit module. The record types are processed in the following order:

1. PECB-REQ-ACTION-ID-REC (element ID records)
2. PECB-REQ-APPROVER-REC (approver records)
3. PECB-REQ-BACKOUT-REC (backout records)
4. PECB-REQ-SCL-RECORDS (SCL records)
5. PECB-SHIP-RECORDS (shipment records)
6. PECB-REQ-COLL-RECORDS (collection records)

Once a valid request type has been processed or bypassed (per user exit) the flag is not checked again. If the exit program does not set any request for data flags the first time it returns to Endevor, Endevor does not reinvoke the exit program.

Note: Endevor sets the PECB-REQUEST-RETURN-CODE field to 2 for EOF conditions, and to 3 for Not Found (NFND) conditions. It returns once to the exit program with this information. Make sure that your exit programs detect and act on an EOF or NFND condition, because, if the exit attempts to request more data after the EOF or NFND has been set, an error occurs.

4.6 Description of Package Exit Actions

There is a package exit point before and after each package action or function. The user exit is responsible for determining action, function, and type (before or after).

In Assembler, the control block fields PECBFNNM, PECBSFNM, and PECBBANM contain action, function, and type literals respectively. The control block fields PECBFNCD, PECBSFCD, and PECBBACD contain numeric codes for action, function, and type.

In COBOL, the control block fields PECB-FUNCTION-CODE, PECB-SUBFUNC-LITERAL, and PECB-BEF-AFTER-LITERAL contain action, function, and type literals respectively. The control block fields PECB-FUNCTION-CODE, PECB-SUBFUNCTION-CODE and PECB-BEFORE-AFTER-CODE contain numeric codes for action, function and type.

The following section describes each package exit action in terms of:

- Possible uses for the exits.
- Control blocks passed, and the fields in each control block that exits can modify.
- Data available to exit programs.

All exit points must be explicitly enabled by the Setup exit. In general, use before-exit calls to determine if a package action should be cancelled or allowed to continue, and after-exit calls for notification and logging purposes.

Note: After-exits can only modify set messages and return codes up to a maximum of 4. If an exit program sets a return code higher than 4, the exit driver will reduce the return code to 4.

4.6.1 Package Exit Return Codes

All exit programs return the same return codes to the PECBRTCD field in assembler and PECB-NDVR-EXIT-RC for COBOL in the Package Exit control block.

Return Code	Meaning
0	The exit didn't change any action-related data. Print messages, call the next exit program (if any), and proceed with package action.
4	The exit changed one or more control block fields. Print messages, check the request update flags in the $PECBDS for assembler and PACKAGE-EXIT-BLOCK in COBOL and updates the appropriate Endevor fields, calls the next exit program (if any), and proceeds with the package action.
8	Print messages and cancel the package action. This return code can be passed only by before-exits. The after-action exit will still be invoked.

With the request for data processing, when Endevor receives a control block from an user exit, it interrogates the PECBRTCD field in assembler and PECB-NDVR-EXIT-RC in COBOL, for cancel, update, and/or message requests. However, instead of calling the next exit in the table, Endevor checks to see if a request for data is currently in progress; if so it continues. If there are no request for data flags the first time the user exit returns to Endevor, Endevor treats the exit as a single call exit.

4.6.1.1 Archive

Endevor performs the ARCHIVE package action after a package is committed (if backout is enabled) or after a package is executed (if backout is not enabled).

4.6.1.2 Possible Uses

Before-archive programs can restrict who, by user ID, can perform an ARCHIVE. The before-archive package action can also allow or disallow the DELETE package action. After-archive programs can work with logging and user notification.

4.6.1.3 Assembler Control Blocks Passed

Endevor passes the following control blocks to ARCHIVE exit programs.

Control block	Modifiable fields
Package Exit Control	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None
Package Header Block	None
Package File Block	■ PFILDDN ■ PFILMEM

4.6.1.4 COBOL Control Blocks Passed

Endevor passes the following control blocks to ARCHIVE exit programs.

Control block	Modifiable fields
Package Exit Control	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	None
Package Header Block	None
Package File Block	■ PFIL-DD-NAME ■ PFIL-MEMBER-NAME

4.6.1.5 Data Resources

The following resources are available on request from the user exit:

Request block	Availability
Package Action	Before/After
Package Approver	Before/After
Package SCL	Before/After
Package SHIP	Before/After
Product COLLECT	Before/After

4.6.2 Backin

Endevor performs the BACKIN package action when a package that has been previously backed out is being reinstalled.

4.6.2.1 Possible Uses

Before-backin programs can be used to:

- Restrict who, by user ID, can perform the BACKIN action.
- Check an external process to see if backin criteria have been met.

After-backin programs can be used for:

- Logging
- Approver notification

4.6.2.2 Assembler Control Blocks Passed

Endevor passes the following control blocks to the BACKIN exit action:

Control block	Fields modifiable by exit
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None
Package Header Block	None

4.6.2.3 COBOL Control Blocks Passed

Endevor passes the following control blocks to the BACKIN exit action:

Control block	Fields modifiable by exit
Package Exit Control Block	■ PECB-NDVR-EXIT ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	None
Package Header Block	None

4.6.2.4 Data Resources

Backin exit programs can request the following data:

Request block	Available to these exits
Package Action	Before/After
Package Approver	Before/After
Package Backout	Before/After
Package SCL	Before/After
Package Ship	Before/After
Product Collect	Before/After

4.6.3 Backout

The BACKOUT package action returns all the outputs affected by the execution of a package to their prior state.

Possible Uses: Before-backout programs can be used to:

- Restrict who, by user ID, can perform the BACKOUT action.
- Check an external process to see if backout criteria have been met.

After-backout programs can be used for:

- Logging
- Approver notification
- Distribution

4.6.3.1 Assembler Control Blocks Passed

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None
Package Header Block	None

4.6.3.2 COBOL Control Blocks Passed

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	None
Package Header Block	None

4.6.3.3 Data Resources

Backout exit programs can request the following data:

Request block	Availability
Package Action	Before/After
Package Approver	Before/After
Package Backout	Before/After
Package SCL	Before/After
Package Ship	Before/After
Product Collect	Before/After

4.6.4 Cast

Endevor performs the CAST package action to freeze the elements included in the package and gather approvers associated with the inventory and, optionally, to validate package components.

Note: The CAST action has a MID exit point. Endevor executes it after the CAST is internally complete, just before returning to the caller. The user can cancel the cast at this point, and Endevor will perform a cleanup.

4.6.4.1 Possible Uses

Before-cast programs can be used to:

- Restrict who, by user ID, can perform the CAST action.
- Check an external process to see if cast criteria have been met.
- Supply approver groups from an external source (the Information/Management Interface, for example).

Mid-cast programs can be used to cancel or allow a cast based on:

- Inventory structure
- Approvers

After-cast programs can be used for:

- Logging
- Approver notification

4.6.4.2 Assembler Control Blocks Passed

Endevor passes the following control blocks to CAST exit programs.

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	■ PREQBOEN ■ PREQEWSD ■ PREQEWST ■ PREQEWED ■ PREQEWET
Package Header Block	None
Package Approver Request Block (modifiable as before function only)	■ PAPPGPNM ■ PAPPEVNM ■ PAPPQUOR ■ PAPPAPP# ■ PAPPAPID ■ PAPPAREQ

Note: If PECBRTCD is set to Return Code 8 during a mid-cast exit, the after cast exit will still be executed.

4.6.4.3 COBOL Control Blocks Passed

Endevor passes the following control blocks to CAST exit programs.

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD

Control block	Modifiable fields
Package Request Block	■ PREQ-BACKOUT-ENABLED ■ PREQ-EXEC-START-DATE ■ PREQ-EXEC-START-TIME ■ PREQ-EXEC-END-DATE ■ PREQ-EXEC-END-DATE
Package Header Block	None
Package Approver Request Block (modifiable as before function only)	■ PAPP-GROUP-NAME ■ PAPP-ENVIRONMENT ■ PAPP-QUORUM-COUNT ■ PAPP-APPROVER-NUMBER ■ PAPP-APPROVAL-ID ■ PAPP-APPR-REQUIRED

Note: If PECB-NDVR-EXIT-RC is set to Return Code 8 during a mid-cast exit, the after cast exit will still be executed.

4.6.4.4 Data Resources

CAST exit programs can request the following data:

Request block	Availability
Package Action	Mid/After
Package Approver	Mid/After
Package SCL	Before/Mid/After

Note: The before-cast exit function is unique in that the user exit may pass back an approver group record from an external source, to be used by the cast package during the gathering of approvers.

In Assembler the exit program initiates this task by:

- Setting the PECBUAPP flag to **Y**.
- Filling in the empty $PAPPREQ block that was passed as part of before-cast exit processing.

In COBOL the exit program initiates this task by:

- Setting the PECB-USENDING-APP-GRPSP flag to **Y**.

- Filling in the empty PACKAGE-EXIT-APPROVER-MAP block that was passed as part of before-cast exit processing.

When Endevor encounters the update request, it

- Validates and stores the data from the request block.
- Clears the request block and flag.
- Returns to the user exit to see if there are more approver groups to be added.

4.6.5 Commit

Endevor performs the COMMIT package action after a package has been accepted as correct. Once a package is committed, it cannot be backed out.

4.6.5.1 Possible Uses in Assembler

Before-commit programs can be used to restrict who, by user ID, can perform the COMMIT action.

After-commit programs can be used for logging and approver notification.

Control block	Modifiable fields
Package Exit Control Block	▪ PECBRTCD ▪ PECBMGCD ▪ PECBMSG ▪ PECBUEXT
Package Request Block	None
Package Header Block	None

4.6.5.2 Possible Uses in COBOL

After-commit programs can be used for logging and approver notification.

Control block	Modifiable fields
Package Exit Control Block	▪ PECB-NDVR-EXIT-RC ▪ PECB-MESSAGE-ID ▪ PECB-MESSAGE ▪ PECB-UEXIT-HOLD-FIELD
Package Request Block	None

Control block	Modifiable fields
Package Header Block	None

4.6.5.3 Data Resources

The following resources are available on request from the user exit:

Request block	Availability
Package Action	Before/After
Package Approver	Before/After
Package Backout	Before
Package SCL	Before/After
Package Ship	Before/After
Product Collect	Before/After

4.6.6 Create

Endevor performs the CREATE package function to create a package either from scratch, from another package, or from an external source. There are four functions with this exit action; BUILD, IMPORT, EDIT, and COPY.

4.6.6.1 Possible Uses

Before-create programs can be used to:

- Restrict who, by user ID, can perform the CREATE action.
- Check some external process to see if criteria for create has been met.
- Perform security checking on the external data set.

After-create programs can be used for:

- Logging
- Approver notification

4.6.6.2 Assembler Control Blocks Passed

Endevor passes the following control blocks to CREATE exit programs.

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	■ PREQBOEN ■ PREQEWSD ■ PREQEWST ■ PREQEWED ■ PREQEWET ■ PREQCOMM ■ PREQDEL ■ PREQAPD ■ PREQPSHR ■ PREQIPKG (COPY)
File Information Block	■ PFILDDN (IMPORT) ■ PFILMEM(IMPORT)
Package Header Block	None

4.6.6.3 COBOL Control Blocks Passed

Endevor passes the following control blocks to CREATE exit programs.

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD

Control block	Modifiable fields
Package Request Block	■ PREQ-BACKOUT-ENABLED ■ PREQ-EXEC-START-DATE ■ PREQ-EXEC-START-TIME ■ PREQ-EXEC-END-DATE ■ PREQ-EXEC-END-TIME ■ PREQ-PACKAGE-COMMENT ■ PREQ-DELETE-ENABLED ■ PREQ-APPEND-PACKAGE ■ PREQ-SHARE-ENABLED ■ PREQ-SOURCE-PACKAGE (COPY)
File Information Block	■ PFIL-DD-NAME (IMPORT) ■ PFIL-MEMBER-NAME (IMPORT)
Package Header Block	None

4.6.6.4 Data Resources

The following resources are available on request from the user exit:

Request block	Availability
Package SCL	Before/After

4.6.7 Delete

The DELETE package action is performed to delete a package.

4.6.7.1 Possible Uses

Before-delete programs can be used to restrict who, by user ID, can perform the DELETE action.

After-delete programs can be used for logging and approver notification.

4.6.7.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the DELETE exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None

4.6.7.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the DELETE exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	None

4.6.7.4 Data Resources

The following resources are available on request from the user exit:

Request block	Availability
Package Action	Before
Package Approver	Before
Package Backout	Before
Package Header	Before
Package SCL	Before
Package Ship	Before

Request block	Availability
Product Collect	Before

4.6.8 Display

The DISPLAY package action is used to display packages or parts of packages. There are five functions with the display package exit function: APPROVERS, ACTION SUMMARY, REPORTS, BACKOUT, and SCL.

4.6.8.1 Possible Uses

Before-display programs can be used to restrict or tailor who, by user ID, can display certain parts of the packages.

4.6.8.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the DISPLAY exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None
Package Header Block	None

4.6.8.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the DISPLAY exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	None
Package Header Block	None

4.6.8.4 Data Resources

The following resources may be available on request from the user exit:

Request block	Availability
Package Action	Before
Package Approver	Before
Package Backout	Before
Package SCL	Before/After
Package Ship	Before
Product Collect	Before

4.6.9 Execute

Endevor uses the EXECUTE package action to execute qualified packages.

4.6.9.1 Possible Uses

Before-execute programs can be used to:

- Restrict who, by user ID, can execute packages.
- Check external requirements.
- Perform additional SCL validation.

After-execute programs can be used for:

- Logging
- Approver notification
- Notification of external processes in case of an abend in the execute process.

4.6.9.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the EXECUTE exit action:

Control block	Modifiable fields
Package Exit Control Block	▪ PECBRTCD ▪ PECBMGCD ▪ PECBMSG ▪ PECBUEXT

Control block	Modifiable fields
Package Request Block	■ PREQEWSD ■ PREQEWST ■ PREQEWED ■ PREQEWET
Package Header Block	None

4.6.9.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the EXECUTE exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	■ PREQ-EXEC-START-DATE ■ PREQ-EXEC-START-TIME ■ PREQ-EXEC-END-DATE ■ PREQ-EXEC-END-TIME
Package Header Block	None

4.6.9.4 Data Resources

The following resources are available on request from the user exit:

Request block	Availability
Package Action	Before/After
Package Approver	Before/After
Package Backout	After
Package SCL	Before/After

4.6.10 Export

Endevor uses the EXPORT package action to create an external copy of the package SCL.

4.6.10.1 Possible Uses

Before-export programs can be used to:

- Restrict who, by user ID, can export packages.
- Check external processes.
- Perform data set checking.

After-export programs can be used for:

- Logging
- Approver notification

4.6.10.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the EXPORT exit function:

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None
Package Header Block	None
Package File Block	■ PFILDDN ■ PFILMEM

4.6.10.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the EXECUTE exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	■ PREQ-EXEC-START-DATE ■ PREQ-EXEC-START-TIME ■ PREQ-EXEC-END-DATE ■ PREQ-EXEC-END-TIME
Package Header Block	None

4.6.10.4 Data Resources

Export exit programs can request the following data:

Request block	Availability
Package Action	Before/After
Package Approver	Before/After
Package Backout	Before/After
Package SCL	Before/After
Package Ship	Before/After
Product Collect	Before/After

4.6.11 GENPKGID

Endevor uses the GENPKGID exit action for package ID generation.

Endevor invokes this exit function when you create a package (option 2 on the Package Options Menu) and leave the PACKAGE ID field blank. When you press ENTER:

- If the GENPKGID exit point has been enabled, Endevor displays a Create/Modify Package panel with a package ID assigned.
- Otherwise, Endevor displays a Package Selection list.

The after-exit point for GENPKGID is after the completion or cancellation of the CREATE process. Endevor executes based on whether or not the package ID was generated, and the status of the after-exit flag from the before-GENPKGID call.

4.6.11.1 Possible Uses

Before-GENPKGID programs can be used to:

- Provide Endevor with a package ID from an external source.
- Check external requirements.

After-GENPKGID programs can be used for:

- Logging
- Package ID cancellation

4.6.11.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the GENPKGID exit action:

Control block	Modifiable fields
Package Exit Control Block	▪ PECBRTCD ▪ PECBMGCD ▪ PECBMSG ▪ PECBUEXT ▪ PECBPKID
Package Request Block	PREQCOMM

4.6.11.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the GENPKGID exit action:

Control block	Modifiable fields
Package Exit Control Block	▪ PECB-NDVR-EXIT-RC ▪ PECB-MESSAGE-ID ▪ PECB-MESSAGE ▪ PECB-UEXIT-HOLD ▪ PECB-PACKAGE-ID
Package Request Block	PREQ-PACKAGE-COMMENT

4.6.11.4 Data Resources

No data blocks can be requested by a user exit.

4.6.12 List

Endevor can invoke LIST exit programs whenever it builds a package selection list. Use them to further tailor the criteria that determines which packages appear on, and can be selected from, that list.

To effectively use LIST package exits, it is important to understand how Endevor builds package selection lists. Endevor determines the content of package selection lists using a two-step process. It:

1. Selects all packages with a status required for the requested action. For example, if you specify the CAST option, Endevor first selects all packages with a status of in-edit.
2. Uses the current user ID to further refine this list, based on whether or not the package has been cast.
   - If a package has not been cast, it is included on the list if the user created it, or if it is sharable.
   - If a package has been cast, it is included on the list if the user is an approver for the package.

Endevor performs these two steps before displaying a selection list, to determine which packages appear on the list, and after a selection is made from the list, to determine if the selection is valid.

Before-list and after-list exit programs are invoked for all packages that pass Step 1. If a package does not pass Step 1, exit programs are not invoked for that package.

Note: If a user allows a pre-list package exit to override the default Endevor processing and allows a package to be put on a list, the user then becomes responsible for the actions that take place. Overriding an exit that is automatically issued by Endevor requires the user to further ensure invalid actions do not occur.

4.6.12.1 Possible Uses

Before-list programs can override the second selection list check and control which package IDs are included on a selection list.

After-list programs can override the second selection list check and control which package IDs can be selected from a selection list.

4.6.12.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the LIST exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBSFNM ■ PECBUEXT
Package Request Block	None
Package Header Block	None

4.6.12.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the LIST exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-SUBFUNC-LITERAL ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	None
Package Header Block	None

4.6.12.4 Data Resources

The following resources may be available on request from the user exit:

Request Block	Availability
Package Action	Before
Package Approver	Before

Request Block	Availability
Package Backout	Before/After
Package SCL	Before/After
Package Ship	Before

4.6.13 Modify

Endevor performs the MODIFY package action against an existing package. There are four functions with this exit action: BUILD, IMPORT, EDIT, and COPY.

4.6.13.1 Possible Uses

Before-modify programs can be used to:

- Restrict who, by user ID, can perform the MODIFY action.
- Check some external process to see if criteria for modify has been met.
- Perform security checking on the external data set.

After-modify programs can be used for:

- Logging
- Approver notification

4.6.13.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the MODIFY exit action:

Control block	Modifiable fields
Package Exit Control Block	▪ PECBRTCD ▪ PECBMGCD ▪ PECBMSG ▪ PECBUEXT

Control block	Modifiable fields
Package Request Block	■ PREQBOEN ■ PREQEWSD ■ PREQEWST ■ PREQEWED ■ PREQEWET ■ PREQCOMM ■ PREQDEL ■ PREQAPD ■ PREQPSHR ■ PREQIPKG (COPY)
File Information Block	■ PFILDDN (IMPORT) ■ PFILMEM (IMPORT)
Package Header Block	None

4.6.13.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the MODIFY exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD

Control block	Modifiable fields
Package Request Block	■ PREQ-BACKOUT-ENABLED ■ PREQ-EXEC-START-DATE ■ PREQ-EXEC-START-TIME ■ PREQ-EXEC-END-DATE ■ PREQ-EXEC-END-TIME ■ PREQ-PACKAGE-COMMENT ■ PREQ-DELETE-ENABLED ■ PREQ-APPEND-PACKAGE ■ PREQ-SHARE-ENABLED ■ PREQ-SOURCE-PACKAGE (COPY)
File Information Block	■ PFIL-DD-NAME (IMPORT) ■ PFIL-MEMBER-NAME (IMPORT)
Package Header Block	None

4.6.13.4 Data Resources

The following resources may be available on request from the user exit:

Request block	Availability
Package SCL	Before/After

4.6.14 Reset

The RESET package action is performed to put a package back into a status that it can be modified, recast, and re-executed.

4.6.14.1 Possible Uses

Before-reset programs can be used to:

- Restrict who, by user ID, can perform the RESET action.
- Check some external process to see if criteria for reset has been met.
- Approver notification.

After-reset programs can be used for logging.

4.6.14.2 Assembler Control Blocks Passed

Endevor passes the following control blocks to RESET exit programs:

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None
Package Header Block	None

4.6.14.3 COBOL Control Blocks Passed

Endevor passes the following control blocks to RESET exit programs:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD ■ PECB-UEXT
Package Request Block	None
Package Header Block	None

4.6.14.4 Data Resources

The following resources may be available on request from the user exit:

Request block	Availability
Package Action	Before
Package Approver	Before
Package Backout	Before
Package SCL	Before/After
Package Ship	Before

4.6.15 Review

Endevor performs the REVIEW package action to either approve or deny a package.

4.6.15.1 Possible Uses

Before-review programs can be used to:

- Restrict who, by user ID, can perform the REVIEW action.
- Check some external process to see if criteria for review has been met.
- Restrict reapproving a package.

After-review programs can be used for:

- Logging
- Approver notification

4.6.15.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the REVIEW exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None
Package Header Block	None

4.6.15.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the REVIEW exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	None

Control block	Modifiable fields
Package Header Block	None

4.6.15.4 Data Resources

The following resources are available on request from the user exit:

Request block	Availability
Package Action	Before/After
Package Approver	Before/After
Package SCL	Before/After

4.6.16 Ship

Endevor uses the SHIP action to ship packages to predefined destinations and to confirm transmission of package per destination. The SHIP action has two functions: XMIT and CONFIRM. SHIP uses the:

- XMIT function during the job that builds the transmit deck and performs the transmission.
- CONFIRM function in a standalone utility that checks for verification from target destinations.

4.6.16.1 Possible Uses

Before-ship programs can be used to:

- Restrict who, by user ID, can perform the SHIP action.
- Check an external process to see ship criteria have been met.

After-ship programs can be used for:

- Operational notification.
- Invoking an external process in case of transmission failure.

4.6.16.2 Assembler Control Blocks Passed

Endevor passes the following control blocks with the SHIP exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECBRTCD ■ PECBMGCD ■ PECBMSG ■ PECBUEXT
Package Request Block	None
Package Header Block	None

4.6.16.3 COBOL Control Blocks Passed

Endevor passes the following control blocks with the SHIP exit action:

Control block	Modifiable fields
Package Exit Control Block	■ PECB-NDVR-EXIT-RC ■ PECB-MESSAGE-ID ■ PECB-MESSAGE ■ PECB-UEXIT-HOLD-FIELD
Package Request Block	None
Package Header Block	None

4.6.16.4 Data Resources

The following resources may be available on request from the user exit:

Request block	Availability
Package Action	Before/After
Package Approver	Before/After
Package Backout	Before/After
Package SCL	Before/After
Package Ship	After

Chapter 5. Package Shipment Exit

Endevor Package Shipment provides an exit point to allow user modifications to be made to a JOB statement which heads up a JCL stream to be executed at a remote site. The job will copy package members to remote production data sets. The exit (contained in a user-written program named C1BMXJCG) will be invoked if it is found to be present in the STEPLIB or JOBLIB library concatenation or in the link list.

Note: The exit program can be located in the STEPLIB data sets, the JOBLIB data sets, or the link list.

The exit program is called prior to writing the remote JOB statement to the AHJOB JCL data set and prior to writing it to the optional CHJOB JCL data set. These data sets are built to contain the JCL needed to copy package members into the appropriate remote production libraries. The default JOB statement is obtained from the Destination Definition of the remote site. (See "Establishing Destinations" in the *User Guide*.) The four lines of the JOB statement, along with a package-id and user-id, are passed to C1BMXJCG. The JOB statement can be modified in any way desired. Upon return, **all four lines** are written to the JCL data set being processed.

Note: The default JOB statement is not edited for syntax when it is made part of the Destination Definition; nor is there a syntax check after the JOB statement is modified by C1BMXJCG.

5.1 Assembler Calling Conventions

Standard IBM linkage conventions are used in the call to C1BMXJCG.

Upon entry to C1BMXJCG:

- Register 13 points to an 18 fullword savearea into which C1BMXJCG **must save** the caller's registers. If C1BMXJCG is designed to CALL other programs, then it must provide itself with its own savearea and should cross chain the two together.
- Register 14 contains the caller's return address.
- Register 15 contains the entry point address of C1BMXJCG.
- Register 1 contains an address pointing to three contiguous fullwords:
  - Parm 1: Points to a 16 character area containing the package id.
  - Parm 2: Points to an 8 character area containing the id of the user executing this shipment.
  - Parm 3: Points to four contiguous 80 character areas. Each contains a single line of the default JOB statement set up in the Destination Definition.

To return to the calling program:

- Register 13 **must** be restored to the value it contained at the time the exit program was entered. Registers 14 and 0 through 12 **must then** be restored to their original values (as shown in the RETURN statement below).
- Register 15 can be set to a return code. (See Other Design Considerations below.)
- Register 14 contains the return address.

5.2 Assembler Example

The following is a simple example of an Assembler C1BMXJCG program, **which does not call any routines that require a savearea.**

Given: The default Destination Definition JOB statements look like this:

```
//JOBNAME  (ACCT),'PACKAGE SHIPMENT',CLASS=A,MSGCLASS=X,
// NOTIFY=USRNAME
/*ROUTE XEQ REMOTE01
//*
```

Required: The exit will change the JOBNAME and USRNAME to the id of the user executing this shipment. Register equates have been omitted.

```
*********************************************************************
* PARM DESCRIPTIONS
*********************************************************************
PARM1     DSECT
PKGID     DS  CL16                 PACKAGE ID
*
PARM2     DSECT
USRID     DS  CL8                  USER ID
*
PARM3 DSECT
JOBSTMNT  DS (0*4)CL80
JOBLINEI  DS CL80                  1ST OF 4
JOBLINE2  DS CL80                  2ND OF 4
JOBLINE3  DS CL80                  3RD OF 4
JOBLINE4  DS CL80                  4TH OF 4
*********************************************************************
* ENTRY LOGIC
*********************************************************************
C1BMXJCG  CSECT
          SAVE  (14,12),,C1BMXJCG  SAVE REGISTERS IN HIGH SAVEAREA
          LR  R12,R15              POINT AT THIS PROGRAM
          USING  C1BMXJCG,R12
          LM  R3,R5,0(Rl)          ADDRESS THE PARMS
          USING  PKGID,R3
          USING  USRID,R4
          USING  JOBSTMNT,R5
          LA  R5,0(,R5)            ELIMINATE THE "LAST PARM" BIT
*********************************************************************
* MOVE THE USERID INTO THE FIRST JOB STATEMENT LINE (AT JOBNAME)
*********************************************************************
         MVC  JOBLINE1+2(8),USRID
*********************************************************************
* MOVE THE USERID INTO THE SECOND JOB STATEMENT LINE (AT USRNAME)
*********************************************************************
         MVC  JOBLINE2+11(8),USRID
*********************************************************************
* EXIT LOGIC
*********************************************************************
         LM  R14,R12,12(R13)       RESTORE REGS AND RETURN
         LA  R15,0                 SET RETURN CODE
         BSM  O,R14 XA             RETURN TO CALLER
         END
```

5.3 COBOL Calling Conventions Example

Given: The default Destination Definition JOB statements look like this:

```
//JOBNAME (ACCT),IPACKAGE SHIPMENT',CLASS=A,MSGCLASS=X,
// NOTIFY=USRNAME
/*ROUTE XEQ REMOTE01
//*
```

Required: The exit will change the JOBNAME and USRNAME to the id of the user executing this shipment.

- The LINKAGE SECTION must contain an 01 level data description for each of the three parms passed by the caller.

```
LINKAGE SECTION.
 01  LS-PACKAGE-ID               PIC X(16).
 01  LS-USER-ID                  PIC X(8).
 01  LS-JOB-STATEMENT.
    05  LS-JOB-LINE-1.
        10  LS-JL1-SLASH-SLASH   PIC XX.
        10  LS-JL1-JOBNAME       PIC X(8).
        10  FILLER               PIC X(70).
    05  LS-JOB-LINE-2.
        10  FILLER               PIC X(11).
        10  LS-JL2-USER-ID       PIC X(8).
        10  FILLER               PIC X(61).
    05  LS-JOB-LINE-3            PIC X(80).
    05  LS-JOB-LINE-4            PIC X(80).
```

- The PROCEDURE DIVISION statement must "use" the three parms in the same order that they are passed.

```
PROCEDURE DIVISION USING LS-PACKAGE-ID LS-USER-ID LS-JOB-STATEMENT.
```

- The JOB statement is now addressable through the LINKAGE SECTION field names and can be modified.

```
MOVE LS-USER-ID TO LS-JL1-JOBNAME.
MOVE LS-USER-ID TO LS-JL2-USER-ID.
```

- The RETURN-CODE special register can be used to pass a value back to the caller. (See Other Design Considerations below.)

5.4 Other Design Considerations

- C1BMXJCG will be loaded only once during the shipment utility run. The exit need not be re-entrant, but must be serially re-useable.
- If C1BMXJCG loads below the 16M line, it will be entered in AMODE 24; otherwise it will be entered in AMODE 31. The savearea and parms are located below the 16M line.
- (Applicable to Assembler only.) The savearea is cleared to binary zeros (low values) prior to branching to C1BMXJCG. Therefore, data cannot be stored in the savearea between calls.
- The return code passed back in register 15 by Assembler programs and in the RETURN-CODE special register of COBOL programs will be reported as part of message "ClBMX541 C1BMXJCG WAS ENTERED FOR DESTINATION <dest-id> AND RETURNED WITH CODE <return-code>." The return code is displayed as eight hexadecimal digits and is used for **reporting purposes only**. The message is written to the ClBMXLOG (sysout) data set.

5.5 Sample

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID.                C1BMXJCG.
 AUTHOR.                    DAVE BERRY.
 DATE-WRITTEN.              01-23-95.
 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
 DATA DIVISION.
 FILE SECTION.
*****************************************************************
*   FROM WHAT I CAN SURMISE HAVING THIS PROGRAM/LOAD MODULE IN  *
*   STEPLIB, JOBLIB OR LINK PACK AREAS INVOKES AN UNDOCUMENTED  *
*   EXIT.  THIS PACKAGE SHIPMENT EXIT ALLOWS USER MODIFICATION  *
*   TO BE MADE TO THE JOB CARD THAT EXECUTES AT THE REMOTE      *
*   SITES (THE IEBCOPY JOB IN THE PRODUCTION DATACENTERS).      *
*                                                               *
*   THE EXIT IS CALLED PRIOR TO WRITING THE REMOTE JOB CARD     *
*   TO THE "AHJOB" JCL DATASET AND PRIOR TO WRITING IT TO THE   *
*   OPTIONAL "CHJOB" JCL DATASET.  THESE DATASETS ARE BUILT TO  *
*   CONTAIN THE JCL NEEDED TO COPY PACKAGE MEMBERS INTO THE     *
*   DESTINATION DEFINITION OF THE REMOTE SITE.                  *
*                                                               *
*   BASICALLY THIS PROGRAM CHANGES THE JOB CARD FROM WHAT THE   *
*   DESTINATION PANEL JOB CARD SAYS TO WHAT THE EQUIVALENT OF   *
*   WHAT THE PACKAGE EXECUTION JOB CARD SHOULD HAVE BEEN.  IT   *
*   ALSO ADDS ,USER=EV0001 TO THE JOB STATEMENTS. - DJB         *
*                                                               *
*****************************************************************
*
 WORKING-STORAGE SECTION.
*
 01  WS-PACKAGE-ID.
     05  WS-PACKAGE-SYS-ID       PIC X(02).
     05  WS-PACKAGE-NUMBER.
         10  WS-PACKAGE-SERIES   PIC X(01).
         10  WS-PACKAGE-TRAILER  PIC X(03).
     05  FILLER                  PIC X(10).
*
 01  WS-JOB-NAME.
     05  WS-PROD-JOBNAME.
         10  WS-PROD-JOB-EV      PIC X(02).
         10  WS-PROD-SYS-ID      PIC X(02).
         10  WS-PROD-TRAILER     PIC X(03).
         10  WS-PROD-POUND       PIC X(01).
     05  WS-NON-PROD-JOBNAME REDEFINES WS-PROD-JOBNAME.
         10  WS-JOB-USER-ID      PIC X(06).
         10  WS-JOB-POUND        PIC X(01).
         10  WS-JOB-SPACE        PIC X(01).
*
 01  WS-USER-EQ.
     05  FILLER                  PIC X(12) VALUE ',USER=EV0001'.
*
 LINKAGE SECTION.
*
 01  LS-PACKAGE-ID               PIC X(16).
```

```
*
 01  LS-USER-ID                  PIC X(08).
*
 01  LS-JOB-STATEMENT.
     05  LS-JOB-LINE-1.
         10  LS-JL1-SLASH-SLASH  PIC X(02).
         10  LS-JL1-JOBNAME      PIC X(08).
         10  FILLER              PIC X(70).
     05  LS-JOB-LINE-2.
         10  FILLER              PIC X(26).
         10  LS-JL2-USER-EQ      PIC X(12).
         10  FILLER              PIC X(42).
*
 PROCEDURE DIVISION USING
     LS-PACKAGE-ID
     LS-USER-ID
     LS-JOB-STATEMENT.

 1000-MAIN-PROCESS.
*******************************************************************
* PUT DIFFERENT "USER=" ON JOB STATEMENT SO JOB HAS AUTHORIZATION
*  TO COPY INTO REMOTE DATASETS.
******************************************************************

     MOVE WS-USER-EQ           TO LS-JL2-USER-EQ.
     MOVE SPACES               TO WS-JOBNAME.
     MOVE LS-PACKAGE-ID        TO WS-PACKAGE-ID.
     IF WS-PACKAGE-NUMBER > '3999'
       PERFORM 2000-PROD-PKG THRU 2000-EXIT
     ELSE
       PERFROM 3000-NON-PROD-PKG THRU 3000-EXIT.

 1000-EXIT.
     GOBACK.

 2000-PROD-PKG.
     MOVE 'EV'                 TO WS-PROD-JOB-EV.
     MOVE WS-PACKAGE-SYS-ID    TO WS-PROD-SYS-ID.
     MOVE WS-PACKAGE-TRAILER   TO WS-PROD-TRAILER.
     MOVE '#'                  TO WS-PROD-POUND.
     MOVE WS-PROD-JOBNAME      TO LS-JL1-JOBNAME.

 2000-EXIT.
     EXIT.

 3000-NON-PROD-PKG.
     MOVE LS-USER-ID           TO WS-JOB-USER-ID.
     MOVE SPACES               TO WS-JOB-SPACE.
     MOVE '#'                  TO WS-PROD-POUND.
     MOVE WS-NON-PROD-JOBNAME  TO LS-JL1-JOBNAME.

 3000-EXIT.
     EXIT.
```

Chapter 6. Package Exit Control Blocks

This chapter describes the package exit control blocks. The sections are:

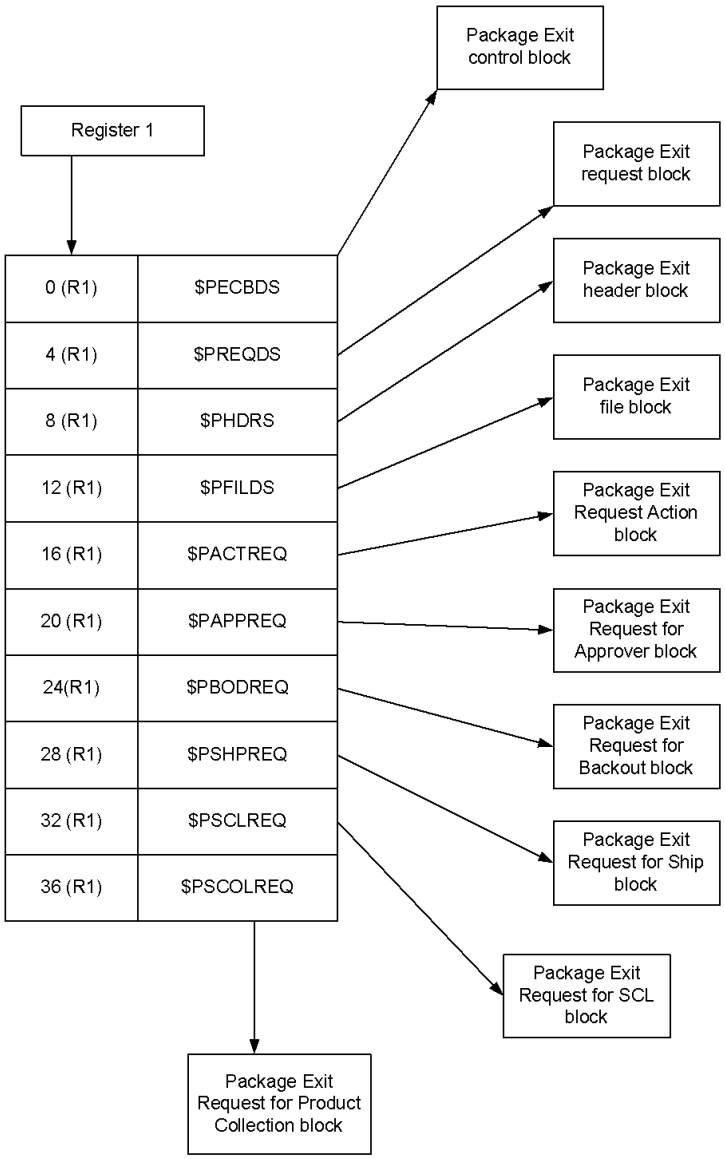
- Control Block Summaries
- Control Block Initialization and Availability
- $PECBDS DSECT: Package Exit Control Block
- PACKAGE-EXIT-BLOCK control block layout: Package Exit Control Block
- $PREQPDS DSECT: Package Request Control Block
- PACKAGE-REQUEST-BLOCK control block layout: Package Request Control Block
- $PHDRDS DSECT: Package Header Control Block
- PACKAGE-EXIT-HEADER-BLOCK control block layout: Package Header Control Block
- $PFILDS DSECT: Package File Control Block
- PACKAGE-EXIT-FILE-BLOCK control block layout: Package File Control Block
- $PACTREQ DSECT: Package Action Request Block
- PACKAGE-EXIT-ACTION-BLOCK control block layout: Package Action Request Block
- $PAPPREQ DSECT: Package Approver Request Block
- PACKAGE-EXIT-APPROVER-BLOCK control block layout: Package Approver Request Block
- $PBODREQ DSECT: Package Backout Request Block
- PACKAGE-EXIT-BACKOUT-BLOCK control block layout: Package Backout Request Block
- $PSHPREQ DSECT: Package Ship Request Block
- PACKAGE-EXIT-SHIPMENT-BLOCK control block layout: Package Ship Request Block
- $PSCLREQ DSECT: Package SCL Request Block
- PACKAGE-EXIT-SCL-BLOCK: Package SCL Request Block

- $PCOLREQ: Product Collection Request Block
- PACKAGE-EXIT-COLLECT-BLOCK: Product Collection Request Block
- Package Exit Example

6.1 Control Block Summaries

When a package exit is invoked, CA-Endevor formats a series of control blocks with information relevant to the exit action. Register 1 points to an address list, which in turn points to the control blocks.

The control blocks are always passed to exit programs in the order shown in the following diagram.



If the control block is not relevant to the package exit action or it doesn't exist, the address in the parameter list is zeros (=F'0'). The last six control blocks are filled in if the user requests approver, element ID, backout, or shipment records.

The last six control blocks are filled in if the user requests approver, element ID, backout or shipment records.

As a rule, all package exit actions can modify the return code fields, message fields, and the record request flags in the $PECBDS(for Assembler), or PACKAGE-EXIT-BLOCK(for COBOL). In addition to these fields, if a package action can modify a field on a panel, then the package exit action can modify the field at the before-exit point.

These control blocks are located in iprfx.iqual.SOURCE. The assembler control blocks are maintained as individual members. The assembler control blocks are maintained as individual members. The COBOL control blocks are all located in one member PKGXBLKS.

6.1.1 Control Blocks

Control block (DSECT name)	Contains
Package Exit Control Block ($PECBDS) (PACKAGE-EXIT-BLOCK)	Information related to the current package action, including the exit action, function, call type literals and codes, return code and message fields. This is the first and primary information block passed to the package exit.
Package Request Block ($PREQPDS) (PACKAGE-REQUEST-BLOCK)	Information specific to the current package action; special flags, comment, package options, and so forth.
Package Header Block ($PHDRDS) (PACKAGE-EXIT-HEADER-BLOCK)	General historic package data. It exists and is passed if the package exists, and includes data such as date and time stamps for execution, cast, approval, and execution history.
Package File Block ($PFILDS) (PACKAGE-EXIT-FILE-BLOCK)	Data set information for external data sets. This block is relevant to the package exit actions with external data sets.
Package Action Request Block ($PACTREQ) (PACKAGE-EXIT-ACTION-BLOCK)	A source or target ID record extracted from the package file on request from the user exit. After the record is extracted, data such as environment, system, subsystem and action is placed in this block and the parameter list. This block is available to all exit actions on request only.
Package Approver Request Block ($PAPPREQ) (PACKAGE-EXIT-APPROVER-MAP)	Approver data extracted from the package file on request from the user exit. Each block contains one group per call. This block is also used to gather approver groups from the user exit on the before-cast call, should the user choose to send such a request. This block is available to all exit actions on request only.

Control block (DSECT name)	Contains
Package Backout Request Block ($PBODREQ) (PACKAGE-EXIT-BACKOUT-BLOCKS)	Backout data extracted from the package file on request from the user exit. Each block contains one record per call. This block is available to all exit actions only on request.
Package Ship Request Block ($PSHPREQ) (PACKAGE-EXIT-SHIPMENT-BLOCKS)	Ship data extracted from the package file on request from the user exit. Each block contains one record per call. This block is available to certain exit actions on request only.
Package SCL Request Block ($PSCLREQ) (PACKAGE-EXIT-SCL-BLOCK)	SCL statements extracted from the package file on request from the user exit. Each block contains 1 to 10 SCL statements. This block is available to certain exit actions only on request.

6.2 Control Block Initialization and Availability

6.2.1 Summary

The following matrix summarizes initialized control blocks per action/exit point and request for data control block availability.

Matrix

Note: Initialized Control blocks have the designation "ICB" preceding their names; Request for Data Control blocks have the designation "RDCB" preceding their names.

- B=before, A=after, and M=mid
- EXIT = PACKAGE-EXIT-BLOCK
- REQUEST = PACKAGE-REQUEST-BLOCK
- HEADER = PACKAGE-EXIT-REQUEST-BLOCK
- FILE = PACKAGE-EXIT-FILE-BLOCK
- ACTION = PACKAGE-EXIT-ACTION-BLOCK
- APPROVER = PACKAGE-EXIT-APPROVER-MAP
- BACKOUT = PACKAGE-EXIT-BACKOUT-BLOCK
- SHIPMENT = PACKAGE-EXIT-SHIPMENT-BLOCK
- SCL = PACKAGE-EXIT-SCL-BLOCK
- COLLECT = PACKAGE-EXIT-COLLECT-BLOCK

Action	ICB $PECBDS	ICB $PREQPDS	ICB $PHDRDS	ICB $PFILDS	RDCB $PACTREQ	RDCB $PAPPREQ	RDCB $PBODREQ	RDCB $PSHPREQ	RDCB $PSCLREQ	RDCB $PCOLREQ
ARCHIVE	B/A	B/A	B/A	B	B/A	B/A		B/A	B/A	B/A
BACKIN	B/A	B/A	B/A		B/A	B/A	B/A	B/A	B/A	B/A
BACKOUT	B/A	B/A	B/A		B/A	B/A	B/A	B/A	B/A	B/A
CAST	B/A/M	B/A/M	B/A/M		M/A	M/A			B/A/M	
COLLECT	B/A	B/A	B/A		B/A	B/A	B/A		A	B/A
COMMIT	B/A	B/A	B/A		B/A	B/A	B	B/A	B/A	A
CREATE -BUILD -IMPORT -COPY -EDIT	 B/A B/A B/A B/A	 B/A B/A B/A B/A	 A A A A	 B					 A	
DELETE	B/A	B/A	B		B	B	B	B	B	B
DISPLAY	B	B	B	B	B	B	B	B	B/A	B
EXECUTE	B/A	B/A	B/A		B/A	B/A	A		B/A	
EXPORT	B/A	B/A	B/A	B	B/A	B/A	B/A	B/A	B/A	B/A
GENPKGID	B/A	B/A								
LIST	B/A	B/A	B/A		B/A	B/A	B/A	B/A	B/A	B
MODIFY -BUILD -IMPORT -COPY -EDIT	 B/A B/A B/A B/A	 B/A B/A B/A B/A	 A A A A	 B					 B/A	
RESET	B/A	B/A	B/A		B/A	B	B	B	B/A	B
REVIEW -APPROVE -DENY	 B/A B/A	 B/A B/A	 B/A B/A		 B/A B/A	 B/A B/A			 B/A	
SHIP -XMIT -CONFIRM	 B/A B/A	 B/A B/A	 B/A B/A		 B/A B/A	 B/A B/A	 B/A B/A	 A A	 B/A	

6.3 Package Exit Control Block

This is the first and primary information block passed to the exit. The information in this block relates to the current CA-Endevor configuration and processing, and includes the package ID, the package action, function, and exit type, an area for the exit to pass back a return code, and a message area.

6.3.1 Primary Information Block - for Assembler

Of particular note in this block is the *hold area*, PECBUEXT. When you are using anchor IDs, this field is used to store address values read by exit programs from the anchor ID table. Any value placed in this field remains there until the end of the CA-Endevor session, or until another user exit loads a new value in this field.

Note: PECBUEXIT is the same field as the ECBUEXIT field used for exits 1-6.

```
         MACRO
&NAME    $PECBDS &DSCT=YES
         AIF   ('&DSCT' NE 'YES').SKPDSCT
$PECBDS  DSECT
         AGO   .SKPEQU
.SKPDSCT ANOP
$PECBDS  DS    0F
.SKPEQU  ANOP
*********************************************************************
*                                                                   *
*         $PECBDS - BLOCK TO DESCRIBE THE PACKAGE EXIT INTERFACE    *
*                   EXIT CONTROL BLOCK                              *
*                                                                   *
*********************************************************************
PECBLEN   DC   AL2(PECBSIZE)       LENGTH OF THE BLOCK
PECBVERS  DC   AL2(PECB$VER)       CURRENT VERSION OF THE BLOCK
PECB$VER  EQU  3                      VERSION 3
PECBID    DC   CL4'PECB'          BLOCK ID
PECBPKID  DC   CL16' '             PACKAGE ID
*
*  THE FOLLOWING SIX FIELDS ARE FILLED IN BY ENDEVOR.
*  THEY CONTAIN THE FUNCTION CODE AND LITERAL NECESSARY FOR A USER
*  EXIT TO DETERMINE THE CALLING EXIT POINT.
*
*  CODES
*
PECBFNCD  DC   H'0'                PACKAGE FUNCTION CODE
*
*   EQUATES FOR THE PACKAGE FUNCTION CODE
*
*   EQU 0                       SETUP EXIT OPTIONS
*   EQU 1                       BACK-IN PACKAGE
*   EQU 2                       BACK-OUT PACKAGE
*   EQU 3                       CAST PACKAGE
*   EQU 4                       COMMIT PACKAGE
*   EQU 5                       CREATE PACKAGE
*   EQU 6                       DELETE PACKAGE
*   EQU 7                       DISPLAY PACKAGE
*   EQU 8                       EXECUTE PACKAGE
*   EQU 9                       EXPORT  PACKAGE
```

```
*   EQU 10                      GENERATE PACKAGE ID
*   EQU 11                      LIST GENERATION
*   EQU 12                      REPLACE SCL
*   EQU 13                      RESET PACKAGE
*   EQU 14                      REVIEW PACKAGE
*   EQU 15                      SHIP THE PACKAGE
*   EQU 16                      ARCHIVE THE PACKAGE
*   EQU 17                      COLLECT PRODUCT (DISTRIBULINK)
*********************************************************************
*
PECBSFCD  DC   H'0'                PACKAGE SUB FUNCTION CODE
*
*   EQUATES FOR THE PACKAGE SUB FUNCTION CODE
*
*
*   EQU 0                       APPROVE (REVIEW)
*   EQU 1                       DENY    (REVIEW)
*   EQU 2                       BUILD    (CREATE/MODIFY)
*   EQU 3                       COPY     (CREATE/MODIFY)
*   EQU 4                       EDIT     (CREATE/MODIFY)
*   EQU 5                       IMPORT   (CREATE/MODIFY)
*   EQU 6                       APPROVER (DISPLAY)
*   EQU 7                       BACKOUT  (DISPLAY)
*   EQU 8                       SCL      (DISPLAY)
*   EQU 9                       PACKAGE  (DISPLAY)
*   EQU 10                      ELMSUM   (DISPLAY)
*   EQU 11                      REPORTS  (DISPLAY)
*   EQU 12                      XMIT     (SHIP   )
*   EQU 13                      CONFIRM  (SHIP   )
*   EQU 14                      BUILD    (LIST   )
*   EQU 15                      CAST     (LIST   )
*   EQU 16                      REVIEW   (LIST   )
*   EQU 17                      EXECUTE  (LIST   )
*   EQU 18                      SHIP     (LIST   )
*   EQU 19                      BACK-OUT (LIST   )
*   EQU 20                      COMMIT   (LIST   )
*   EQU 21                      UTILITY  (LIST   )
*   EQU 22                      DISPLAY  (LIST   )
*   EQU 23                      COLLECT  (LIST   )
*   EQU 24                      STAGE    (COLLECT)
*   EQU 25                      CORRINFO (DISPLAY)
*   EQU 26                      ARCHIVE  (LIST)
*   EQU 27                      CONFIRM  (COLLECT)
*   EQU 28                      INSPECT  (LIST)
*   EQU 29                      BACK-OUT PACKAGE
*   EQU 30                      BACKIN   PACKAGE
*   EQU 31                      DELETE   PACKAGE
*   EQU 32                      EXPORT   PACKAGE
*   EQU 33                      RESET    PACKAGE
*   EQU 34                      CAST     PACKAGE
*   EQU 35                      COMMIT   PACKAGE
*   EQU 36                      EXECUTE  PACKAGE
*
*********************************************************************
PECBBACD  DC   H'0'                PACKAGE BEFORE/MID/AFTER CODE
*
*   EQUATES FOR THE BEFORE, MID, AND AFTER CODE
*
*   EQU 00                      BEFORE
```

```
*   EQU 01                      MIDDLE
*   EQU 02                      AFTER
*
*********************************************************************
*
*  LITERALS
*
PECBFNNM  DC   CL8' '              PACKAGE FUNCTION LITERAL
PECBSFNM  DC   CL8' '              SUBFUNCTION LITERAL
PECBBANM  DC   CL8' '              BEFORE/AFTER LITERAL
*
*  THE FOLLOWING FIELDS ARE USED EXCLUSIVELY DURING SETUP PROCESSING.
*  THE USER EXIT SHOULD MODIFY THESE FIELDS TO ENABLE EXIT POINTS.
*  THIS SETUP IS DONE ONCE PER ENDEVOR SESSION.
*
*  N_O_T_E:  RELATED RELEASE 3.7 FIELDS (PECBTSOX, PECBBATX, PECBARBE,
*            AND PECBARAF) APPEAR AT THE END OF THE DSECT.
*
PECBBIBE  DC   C'N'                BEFORE BACKIN EXIT
PECBBIAF  DC   C'N'                AFTER  BACKIN EXIT
*
PECBBOBE  DC   C'N'                BEFORE BACKOUT EXIT
PECBBOAF  DC   C'N'                AFTER  BACKOUT EXIT
*
PECBCABE  DC   C'N'                BEFORE CAST    EXIT
PECBCAMD  DC   C'N'                MID    CAST    EXIT
PECBCAAF  DC   C'N'                AFTER  CAST    EXIT
*
PECBCOBE  DC   C'N'                BEFORE COMMIT  EXIT
PECBCOAF  DC   C'N'                AFTER  COMMIT  EXIT
*
PECBCBBE  DC   C'N'                BEFORE CREATE / BUILD EXIT
PECBCBAF  DC   C'N'                AFTER  CREATE / BUILD EXIT
*
PECBCCBE  DC   C'N'                BEFORE CREATE / COPY EXIT
PECBCCAF  DC   C'N'                AFTER  CREATE / COPY EXIT
*
PECBCEBE  DC   C'N'                BEFORE CREATE / EDIT   EXIT
PECBCEAF  DC   C'N'                AFTER  CREATE / EDIT EXIT
*
PECBCIBE  DC   C'N'                BEFORE CREATE / IMPORT EXIT
PECBCIAF  DC   C'N'                AFTER  CREATE / IMPORT EXIT
*
PECBDEBE  DC   C'N'                BEFORE DELETE  EXIT
PECBDEAF  DC   C'N'                AFTER  DELETE  EXIT
*
PECBDABE  DC   C'N'                BEFORE DISPLAY APPROVER
PECBDBBE  DC   C'N'                BEFORE DISPLAY BACKOUT
*
PECBDSBE  DC   C'N'                BEFORE DISPLAY SCL
PECBDXBE  DC   C'N'                BEFORE DISPLAY ELMSUM
PECBDPBE  DC   C'N'                BEFORE DISPLAY PACKAGES
PECBDRBE  DC   C'N'                BEFORE DISPLAY REPORTS
*
PECBEXBE  DC   C'N'                BEFORE EXECUTE EXIT
PECBEXAF  DC   C'N'                AFTER  EXECUTE EXIT
*
PECBEPBE  DC   C'N'                BEFORE EXPORT  EXIT
PECBEPAF  DC   C'N'                AFTER  EXPORT  EXIT
```

```
*
PECBGPBE  DC   C'N'                BEFORE GENPKGID EXIT
PECBGPAF  DC   C'N'                AFTER  GENPKGID EXIT
*
PECBLTBE  DC   C'N'                BEFORE LIST    EXIT
PECBLTAF  DC   C'N'                AFTER  LIST    EXIT
*
PECBMBBE  DC   C'N'                BEFORE MODIFY / BUILD EXIT
PECBMBAF  DC   C'N'                AFTER  MODIFY / BUILD EXIT
*
PECBMCBE  DC   C'N'                BEFORE MODIFY / COPY EXIT
PECBMCAF  DC   C'N'                AFTER  MODIFY / COPY EXIT
*
PECBMEBE  DC   C'N'                BEFORE MODIFY / EDIT EXIT
PECBMEAF  DC   C'N'                AFTER  MODIFY / EDIT EXIT
*
PECBMIBE  DC   C'N'                BEFORE MODIFY / IMPORT EXIT
PECBMIAF  DC   C'N'                AFTER  MODIFY / IMPORT EXIT
*
PECBRSBE  DC   C'N'                BEFORE RESET   EXIT
PECBRSAF  DC   C'N'                AFTER  RESET   EXIT
*
PECBRABE  DC   C'N'                BEFORE REVIEW / APPROVE EXIT
PECBRAAF  DC   C'N'                AFTER  REVIEW / APPROVE EXIT
*
PECBRDBE  DC   C'N'                BEFORE REVIEW / DENY EXIT
PECBRDAF  DC   C'N'                AFTER  REVIEW / DENY EXIT
*
PECBSXBE  DC   C'N'                BEFORE SHIP / XMIT EXIT
PECBSXAF  DC   C'N'                AFTER  SHIP / XMIT EXIT
*
PECBSCBE  DC   C'N'                BEFORE SHIP / CONFIRM EXIT
PECBSCAF  DC   C'N'                AFTER  SHIP / CONFIRM EXIT
*
*  THE FOLLOWING FIELDS PERTAIN TO REQUEST FOR DATA PROCESSING.
*
***
***    THE FOLLOWING THREE FIELDS ARE INFORMATIONAL FIELDS FOR THE USER
***    EXIT TO VERIFY WHETHER OR NOT A REQUEST TYPE IS AVAILABLE.
***    THESE FLAGS ARE SET BY ENDEVOR EXIT PROCESSING PRIOR TO INVOKING
***    THE FIRST USER EXIT MODULE.
***
PECBACTE  DC   C'N'                ACTION RECORD EXISTENCE FLAG
PECBAPPE  DC   C'N'                APPROVER RECORD EXISTENCE FLAG
PECBBODE  DC   C'N'                BACKOUT RECORD EXISTENCE FLAG
***
***
***    THE FOLLOWING FIELDS ARE REQUEST FOR DATA FLAGS. TO REQUEST
***    A RECORD BE SENT, MOVE A C'Y' INTO THE APPROPRIATE FIELD.
***    CHECK USER DOC TO MAKE SURE THAT THE DATA REQUESTED EXISTS AT
***    THE POINT AT WHICH THE REQUEST IS BEING MADE.
***
PECBACTR  DC   C'N'                USER REQUESTS ACTION ID REC
PECBAPPR  DC   C'N'                USER REQUESTS APPROVER REC
PECBBODR  DC   C'N'                USER REQUESTS BACKOUT REC
PECBSHPR  DC   C'N'                USER REQUESTS SHIP RECORDS
PECBSCLR  DC   C'N'                USER REQUESTS SCL RECORDS
***
***    THE PECBRQRC FIELD IS USED BY ENDEVOR TO TELL USER EXIT WHAT
```

```
***    THE RESULT OF THE REQUEST FOR DATA ATTEMPT
***
PECBRQRC  DC   H'0'                USER REQUEST FOR DATA RETURN CODE
PECBRROK  EQU  1                   READ WAS SUCCESSFUL, RECORD SENT
PECBREOF  EQU  2                   END OF FILE FOR RECORD TYPE
PECBRNFD  EQU  3                   NO RECORDS FOUND FOR KEY
PECBRERR  EQU  12                  SEVERE ERROR
***
***
***    THE PECBNDXX FIELDS ARE USED BY ENDEVOR TO TELL USER EXIT
***    ABOUT HIGH RETURN CODES, ERROR TEXT, AND ABEND CODES.
***    THESE FIELDS ARE ONLY APPLICABLE FOR AFTER CALLS.
***
***
PECBNDRC  DC   H'0'                NDVR HIGH RETURN CODE
PECBNDAB  DC   CL6' '              NDVR ABEND CODE
PECBNDMG  DC   CL133' '            NDVR MSGTEXT
PECBNDML  DC   H'0'                NDVR MSGTEXT LENGTH
PECBRTCD  DC   F'0'                RETURN CODE FROM EXIT
PECB$OK   EQU  0                       RETURN 0 ALL O.K.
PECB$MOK  EQU  4                       RETURN 4 ALL O.K. WITH MODS
PECB$CAN  EQU  8                       RETURN 8 ABORT
PECBUSER  DC   CL8' '              TSO USER/BATCH JOBNAME
PECBMODE  DC   CL1' '              MODE FOR ONLINE(TSO) OR BATCH
PECB$TSO  EQU  C'T'                     TSO MODE
PECB$BAT  EQU  C'B'                     BATCH MODE
PECBMGCD  DC   CL4' '              MESSAGE ID FOR ERROR MESSAGE
*                                  MESSAGE CODE WILL BE C1XNNNNS
PECBMGLN  DC   H'0'                LENGTH OF ERROR MESSAGE
PECBMGSZ  EQU  132                     SIZE IS 132
PECBMSG   DC   CL(PECBMGSZ)' '     MESSAGE
PECBCALR  DC   CL1' '              CALLERS ORIGIN
PECB$NDV  EQU  C'E'                ENDEVOR
PECB$QE   EQU  C'Q'                QUICK-EDIT
PECBENPK  DC   CL1'N'              ENTERPRISE PKG FLAG (Y/N)
PECBUEXT  DC   F'0'                USER EXIT HOLD FIELD
PECBHIRC  DC   F'0'                HIGHEST RC SET BY PREVIOUS EXITS
***
***
***    THE FOLLOWING FIELDS ARE UPDATE FLAGS FOR THE USER CONTROL
***    BLOCKS.  THE USER EXIT MUST SET THESE FLAGS TO 'Y' TO INDICATE
***    TO ENDEVOR THAT THESE BLOCKS HAVE BEEN MODIFIED.
***
***
PECBUECB  DC   C'N'                MODS WHERE MADE TO THE PECBDS
PECBUREQ  DC   C'N'                MODS WHERE MADE TO THE PREQPDS
PECBUFIL  DC   C'N'                MODS WHERE MADE TO THE PFILDS
PECBUAPP  DC   C'N'                EXIT SENDING APPR GROUPS:$PAPPREQ
*
* ENDEVOR ACKNOWLEDGEMENT OF THE NUMBER OF $PAPPREQ BLOCKS
* RECEIVED FROM THE USER EXIT
*
PECB#APG  DC   H'0'                NDVR ACK: # OF $PAPPREQ BLKS REC'D
*
*  ARCHIVE PACKAGE EXIT ENABLE/DISABLE FLAGS
*
PECBARBE  DC   C'N'                BEFORE ARCHIVE
PECBARAF  DC   C'N'                AFTER  ARCHIVE
*
```

```
*
*  THE FOLLOWING FIELDS ARE USED EXCLUSIVELY DURING SETUP PROCESSING.
*  AT SETUP TIME, THE INITIAL VALUES ARE AS SHOWN UNLESS OVERRIDDEN
*  IN THE C1UEXITS TABLE (@C1UEXIT TYPE=START,XIT7BATX=Y).
*
PECBTSOX  DC    C'Y'               INVOKE EXIT IF RUNNING UNDER TSO
PECBBATX  DC    C'N'               INVOKE EXIT IF BATCH (NO TSO)
*
*  COLLECT PRODUCT EXIT ENABLE/DISABLE FLAGS (USED IN SETUP PROCESSING)
*
PECBCSBE  DC   C'N'                BEFORE COLLECT / STAGING EXIT
PECBCSAF  DC   C'N'                AFTER  COLLECT / STAGING EXIT
*
*  COLLECT PRODUCT REQUEST FOR DATA FLAG
*
PECBCOLR  DC   C'N'                USER REQUESTS COLLECTION RECORDS
PECBRSV1  DC   CL7' '              RESERVED FOR INTERNAL USE
PECBSIZE  EQU   *-$PECBDS          LENGTH OF $PECBDS BLOCK
         MEND
```

6.3.1.1 Field Descriptions

Brief descriptions of the fields in the $PECBDS DSECT follow.

Field	Description
PECBLEN	Size of the Exit control block ($PECBDS), in bytes. Not modifiable by any exits.
PECBVERS	Version number to identify the Exit control block ($PECBDS) written by the exit handler. This should always be 2. It is not modifiable by exits. Use the PECB$VER label to verify that the correct version of this DSECT was passed to your exit.
PECBID	Identifies this DSECT as the Exit control block. Value is always "PECB." Not modifiable by any exits.
PECBPKID	Current CA-Endevor package ID.

Note: The next six fields contain the action, function, and type codes. User exits use these values to determine which package exit has been invoked.

Field	Description
PECBFNCD	Package action code. Acceptable values are 0-17. **Value** — **Meaning** 0 — Setup exit options 1 — Back in package 2 — Back out package 3 — Cast package 4 — Commit package 5 — Create package 6 — Delete package 7 — Display package 8 — Execute package 9 — Export package 10 — Generate package ID 11 — Generate list 12 — Modify package 13 — Reset package 14 — Review package 15 — Ship package 16 — Archive package 17 — Collect Product

Field	Description
PECBSFCD	Package function code. Acceptable values are 0-36.

Value	Meaning	Related Action
0	Approve	Review
1	Deny	Review
2	Build	Create & Modify
3	Copy	Create & Modify
4	Edit	Create & Modify
5	Import	Create & Modify
6	Approver	Display
7	Backout	Display
8	SCL	Display
9	Package	Display
10	ELMSUM	Display
11	Reports	Display
12	XMIT	Ship
13	Confirm	Ship
14	Build	List
15	Cast	List
16	Review	List
17	Execute	List
18	Ship	List
19	Back out	List
20	Commit	List
21	Utility	List
22	Display	List
23	Collect	List
24	Stage	Collect
25	Corrinfo	Display
26	Archive	List
27	Confirm	Collect
28	Inspect	List
29	Back-out	Package
30	Backin	Package
31	Delete	Package
32	Export	Package
33	Reset	Package
34	Cast	Package
35	Commit	Package
36	Execute	Package

PECBBACD

Package before/mid-cast/after code.

- 00--Before
- 01--After
- 02--Mid-cast

Field	Description
PECBFNNM	Literal for requested package action. For example, CREATE.
PECBSFNM	Literal for requested package function. For example, IMPORT.
PECBBANM	Values are either BEFORE (for before-exits), MID (for mid-cast exits) or AFTER (for after-exits).

Note: The following fields are used during Setup. The Setup call allows an exit program to indicate whether it is designed to handle calls while operating under TSO and/or as a batch program (without TSO services) and also allows the exit program to enable those exit points which it was designed to process.

The user exit must indicate which operating environments it can handle. Acceptable values are **N** (not enabled for this operating environment) and **Y** (enabled).

Set this field to Y	To operate	The default is
PECBTSOX	Under TSO	Y
PECBBATX	Batch (no TSO)	N

The user exit must indicate which exit points are enabled. For an exit to get control, **both** the proper operating environment setup flag **and** the exit point setup flag must be **Y**. Acceptable values for exit point setup flags are **N** (not enabled: default) and **Y** (enabled).

Set this field to Y	To invoke this exit point
PECBARBE	Before-archive
PECBARAF	After-archive
PECBBIBE	Before-backin
PECBBIAF	After-backin
PECBBOBE	Before-backout
PECBBOAF	After-backout
PECBCABE	Before-cast
PECBCAMD	Mid-cast
PECBCAAF	After-cast
PECBCSBE	Before-collect
PECBCSAF	After-collect
PECBCOBE	Before-commit
PECBCOAF	After-commit

Set this field to Y	To invoke this exit point
PECBCBBE	Before-create/build
PECBCBAF	After-create/build
PECBCCBE	Before-create/copy
PECBCCAF	After-create/copy
PECBCEBE	Before-create/edit
PECBCEAF	After-create/edit
PECBCIBE	Before-create/import
PECBCIAF	After-create/import
PECBDEBE	Before-delete
PECBDEAF	After-delete
PECBDABE	Before-display approver
PECBDBBE	Before-display backout
PECBDSBE	Before Display SCL
PECBDXBE	Before Display element summary
PECBDPBE	Before Display packages
PECBDRBE	Before Display reports
PECBEXBE	Before Execute
PECBEXAF	After Execute
PECBEPBE	Before Export
PECBEPAF	After Export
PECBGPBE	Before GENPKGID
PECBGPAF	After GENPKGID
PECBLTBE	Before List
PECBLTAF	After List
PECBMBBE	Before Modify/Build
PECBMBAF	After Modify/Build
PECBMCBE	Before Modify/Copy
PECBMCAF	After Modify/Copy
PECBMEBE	Before Modify/Edit
PECBMEAF	After Modify/Edit
PECBMIBE	Before Modify/Import

Set this field to Y	To invoke this exit point
PECBMIAF	After Modify/Import
PECBRSBE	Before Reset
PECBRSAF	After Reset
PECBRABE	Before Review/Approve
PECBRAAF	After Review/Approve
PECBRDBE	Before Review/Deny
PECBRDAF	After Review/Deny
PECBSXBE	Before Ship transmit
PECBSCBE	Before Ship confirm
PECBSXAF	After Ship transmit
PECBSCAF	After Ship confirm

Note: The following fields pertain to the requests for data. CA-Endevor sets the five existence flags before calling the user exit. The user exit should check these fields before setting request for data flags:

Field	Description
PECBACTE	Action, element, data set records exist.
PECBAPPE	Approver group records exist.
PECBBODE	Backout records exist.

The next five fields are the request for data flags. CA-Endevor clears a flag each time the user exit returns to CA-Endevor. Therefore the user exit must reset the request flag each time it requests a record. An exit can request only one data block each time it returns to CA-Endevor.

Field	Description
PECBACTR	Request for action element, data set records.
PECBAPPR	Request for approver group record.
PECBBODR	Request for backout record.
PECBSHPR	Request for ship record.
PECBSCLR	Request for SCL record.
PECBCOLR	Request for collection record.

Field	Description
PECBRQRC	Request for data return code. This field is modified by CA-Endevor to inform the user exit of the request for data results. ■ 1--PECBRROK ■ 2--PECBREOF ■ 3--PECBNFD ■ 12--PECBRERR When an EOF or NFND condition occurs, CA-Endevor returns this information to the user exit, but does not allow further requests against that data type.
PECBNDRC	CA-Endevor high return code.
PECBNDAB	CA-Endevor abend code.
PECBNDMG	CA-Endevor message text.
PECBNDML	CA-Endevor message length.
PECBRTCD	Return code. Modifiable by all exit programs. ■ 0--PECB$OK. Successful. ■ 4--PECB$MOK. Successful, but the exit modified some control block data. ■ 8--PECB$CAN. Stop package processing. This return code can only be bypassed by before-exits.
PECBUSER	User ID associated with the current processing.
PECBMODE	Indicates whether the current action was requested in foreground or batch. Not modifiable by exits. ■ T--PECB$TSO. Action requested in foreground. ■ B--PECB$BAT. Action requested in batch.
PECBMGCD	Modifiable by all exits. The default is 0000. Used to construct the message code written to the CA-Endevor Execution Report for the current action. A message code precedes each message written to the report, and applies if the PECBMSG field contains a non-blank value. For messages produced by user exits, the error code is formatted as C1UNNNNs, where NNNN is defined by this field, and S is a severity code associated with PECBRTCD (W, C, E, or S, respectively for return codes 4, 8, 12, and 16).
PECBMGLN	Length of the error message.
PECBMGSZ	Error message size.

Field	Description
PECBMSG	Error message associated with the current activity. This is blank on entry and modifiable by all exit points. If there is a message in this field when the program returns from the user exit, the exit handler records the message in the CA-Endevor Execution Report. The message is preceded in that report with an error code (C1UNNNNs format, described further above for the PECBMGCD field).
PECBCALR	Exit is executing under Endevor or Quick-Edit. Value Label Meaning E PECB$NDV Exit is running under Endevor. Q PECB$QE Exit is running under Quick-Edit.
PECBUEXT	User field, used to save (and pass) information between user exits. Generally, this field contains an address that points to a block of user data. Modifiable by all exit points.
PECBHIRC	Highest return code set by previously executed system and user package exit for this action/function and before/after code.
PECBUECB	Update CA-Endevor fields related to fields in the PECBDS (value **Y/N**). When this flag is set, CA-Endevor copies the PECBDS fields to the correlating CA-Endevor control block fields.
PECBUREQ	Update CA-Endevor fields related to fields in the PREQPDS (value **Y/N**). When this flag is set, CA-Endevor copies the PREQPDS fields to the correlating CA-Endevor control block fields.
PECBUFIL	Update CA-Endevor fields related to fields in the PFILDS (value **Y/N**). When this flag is set, CA-Endevor copies the PFILDS fields to the correlating CA-Endevor control block fields.
PECBUAPP	Update Approver Group Records (value **Y/N**). Valid only for before-cast exit call.
PECB#APG	Number of approver groups sent.
PECBSIZE	Size of the $PECBDS DSECT. Not modifiable by exits.

6.3.2 Primary Information Block - for COBOL

Of particular note in this block is the hold area PECB-UEXIT-HOLD-FIELD. When you are using anchor IDs, this field is used to store address values read by exit programs from the anchor ID table. Any value placed in this field remains there until the end of the CA-Endevor session, or until another user exit loads a new value in this field.

Note: PECB-UEXIT-HOLD-FIELD is the same field as the ECB-UEXIT-HOLD-FIELD field field used for exits 1-6.

```
*****************************************************************
*    PACKAGE-EXIT-BLOCK - BLOCK TO DESCRIBE THE PACKAGE EXIT    *
*    INTERFACE EXIT CONTROL BLOCK                               *
*****************************************************************

 01  PACKAGE-EXIT-BLOCK.
     05  PECB-LENGTH             PIC 9(4) COMP SYNC.
     05  PECB-CURRENT-VERSION    PIC 9(4) COMP SYNC.
     05  PECB-BLOCK-ID           PIC X(4).
     05  PECB-PACKAGE-ID         PIC X(16).
*
*****************************************************************
* THE FOLLOWING SIX FIELDS ARE FILLED IN BY ENDEVOR.
* THEY CONTAIN THE FUNCTION CODE AND LITERAL NECESSARY FOR
* A USER EXIT TO DETERMINE THE CALLING EXIT POINT.
*
     05  PECB-FUNCTION-CODE      PIC 9(4) COMP SYNC.
*
* EQUATES FOR THE PACKAGE FUNCTION CODE
*
         88  SETUP-EXIT-OPTIONS  VALUE  0.
         88  BACK-IN-PACKAGE     VALUE  1.
         88  BACK-OUT-PACKAGE    VALUE  2.
         88  CAST-PACKAGE        VALUE  3.
         88  COMMIT-PACKAGE      VALUE  4.
         88  CREATE-PACKAGE      VALUE  5.
         88  DELETE-PACKAGE      VALUE  6.
         88  DISPLAY-PACKAGE     VALUE  7.
         88  EXECUTE-PACKAGE     VALUE  8.
         88  EXPORT-PACKAGE      VALUE  9.
         88  GENERATE-PACKAGE-ID VALUE 10.
         88  LIST-GENERATION     VALUE 11.
         88  REPLACE-SCL         VALUE 12.
         88  RESET-PACKAGE       VALUE 13.
         88  REVIEW-PACKAGE      VALUE 14.
         88  SHIP-THE-PACKAGE    VALUE 15.
         88  ARCHIVE-PACKAGE     VALUE 16.
         88  DISTRIB-COLLECT     VALUE 17.

     05  PECB-SUBFUNCTION-CODE   PIC 9(4) COMP SYNC.
*
* EQUATES FOR THE PACKAGE SUB-FUNCTION CODE
*
         88  PECB-APPROVE        VALUE  0.
         88  PECB-DENY           VALUE  1.
         88  PECB-BUILD          VALUE  2.
         88  PECB-COPY           VALUE  3.
         88  PECB-EDIT           VALUE  4.
         88  PECB-IMPORT         VALUE  5.
         88  PECB-DISP-APPRVER   VALUE  6.
         88  PECB-DISP-BACKOUT   VALUE  7.
         88  PECB-DISP-SCL       VALUE  8.
         88  PECB-DISP-PKG       VALUE  9.
         88  PECB-DISP-ELMSUM    VALUE 10.
         88  PECB-DISP-REPORTS   VALUE 11.
         88  PECB-SHIP-XMIT      VALUE 12.
```

```
            88  PECB-SHIP-CONFIRM   VALUE 13.
            88  PECB-LIST-BUILD     VALUE 14.
            88  PECB-LIST-CAST      VALUE 15.
            88  PECB-LIST-REVIEW    VALUE 16.
            88  PECB-LIST-EXECUTE   VALUE 17.
            88  PECB-LIST-SHIP      VALUE 18.
            88  PECB-LIST-BACKOUT   VALUE 19.
            88  PECB-LIST-COMMIT    VALUE 20.
            88  PECB-LIST-UTILITY   VALUE 21.
            88  PECB-LIST-DISPLAY   VALUE 22.
            88  PECB-LIST-COLLECT   VALUE 23.
            88  PECB-COLLECT-STAGE  VALUE 24.
            88  PECB-DISP-CORRINFO  VALUE 25.
            88  PECB-LIST-ARCHIVE   VALUE 26.
            88  PECB-COLL-CONFIRM   VALUE 27.
            88  PECB-LIST-INSPECT   VALUE 28.
            88  PECB-PKG-BACKOUT    VALUE 29.
            88  PECB-PKG-BACKIN     VALUE 30.
            88  PECB-PKG-DELETE     VALUE 31.
            88  PECB-PKG-EXPORT     VALUE 32.
            88  PECB-PKG-RESET      VALUE 33.
            88  PECB-PKG-CAST       VALUE 34.
            88  PECB-PKG-COMMIT     VALUE 35.
            88  PECB-PKG-EXECUTE    VALUE 36.

        05  PECB-BEFORE-AFTER-CODE  PIC  9(4) COMP SYNC.
*
* EQUATES FOR THE PACKAGE BEFORE-MID-AFTER CODE
*
            88  PECB-BEFORE         VALUE 0.
            88  PECB-AFTER          VALUE 1.
            88  PECB-MID            VALUE 2.

        05  PECB-FUNCTION-LITERAL   PIC X(8).
        05  PECB-SUBFUNC-LITERAL    PIC X(8).
        05  PECB-BEF-AFTER-LITERAL  PIC X(8).
*
*****************************************************************
* THE FOLLOWING FIELDS ARE USED EXCLUSIVELY DURING SETUP
* PROCESSING.  THE USER EXIT SHOULD MODIFY THESE FIELDS TO
* DISABLE EXIT POINTS.  THIS SETUP IS DONE ONCE PER ENDEVOR
* SESSION. TO DISABLE AN EXIT POINT, PUT AN 'N' IN THE
* APPROPRIATE FLAG. THESE FLAGS FOLLOW.
*
        05  PECB-BEFORE-BACKIN      PIC X(1).
        05  PECB-AFTER-BACKIN       PIC X(1).
        05  PECB-BEFORE-BACKOUT     PIC X(1).
        05  PECB-AFTER-BACKOUT      PIC X(1).
        05  PECB-BEFORE-CAST        PIC X(1).
        05  PECB-MID-CAST           PIC X(1).
        05  PECB-AFTER-CAST         PIC X(1).
        05  PECB-BEFORE-COMMIT      PIC X(1).
        05  PECB-AFTER-COMMIT       PIC X(1).
        05  PECB-BEFORE-CREATE-BLD  PIC X(1).
        05  PECB-AFTER-CREATE-BLD   PIC X(1).
        05  PECB-BEFORE-CREATE-COPY PIC X(1).
        05  PECB-AFTER-CREATE-COPY  PIC X(1).
        05  PECB-BEFORE-CREATE-EDIT PIC X(1).
        05  PECB-AFTER-CREATE-EDIT  PIC X(1).
```

```
     05  PECB-BEFORE-CREATE-IMPT PIC X(1).
     05  PECB-AFTER-CREATE-IMPT  PIC X(1).
     05  PECB-BEFORE-DELETE      PIC X(1).
     05  PECB-AFTER-DELETE       PIC X(1).
     05  PECB-BEFORE-DSPLY-APPR  PIC X(1).
     05  PECB-BEFORE-DSPLY-BKOUT PIC X(1).
     05  PECB-BEFORE-DSPLY-SCL   PIC X(1).
     05  PECB-BEFORE-DSPLY-ELMSM PIC X(1).
     05  PECB-BEFORE-DSPLY-PKG   PIC X(1).
     05  PECB-BEFORE-DSPLY-RPT   PIC X(1).
     05  PECB-BEFORE-EXEC        PIC X(1).
     05  PECB-AFTER-EXEC         PIC X(1).
     05  PECB-BEFORE-EXPORT      PIC X(1).
     05  PECB-AFTER-EXPORT       PIC X(1).
     05  PECB-BEFORE-GENPID      PIC X(1).
     05  PECB-AFTER-GENPID       PIC X(1).
     05  PECB-BEFORE-LIST        PIC X(1).
     05  PECB-AFTER-LIST         PIC X(1).
     05  PECB-BEFORE-MOD-BLD     PIC X(1).
     05  PECB-AFTER-MOD-BLD      PIC X(1).
     05  PECB-BEFORE-MOD-CPY     PIC X(1).
     05  PECB-AFTER-MOD-CPY      PIC X(1).
     05  PECB-BEFORE-MOD-EDIT    PIC X(1).
     05  PECB-AFTER-MOD-EDIT     PIC X(1).
     05  PECB-BEFORE-MOD-IMPT    PIC X(1).
     05  PECB-AFTER-MOD-IMPT     PIC X(1).
     05  PECB-BEFORE-RESET       PIC X(1).
     05  PECB-AFTER-RESET        PIC X(1).
     05  PECB-BEFORE-REV-APPR    PIC X(1).
     05  PECB-AFTER-REV-APPR     PIC X(1).
     05  PECB-BEFORE-REV-DENY    PIC X(1).
     05  PECB-AFTER-REV-DENY     PIC X(1).
     05  PECB-BEFORE-SHIP-XMIT   PIC X(1).
     05  PECB-AFTER-SHIP-XMIT    PIC X(1).
     05  PECB-BEFORE-SHIP-CON    PIC X(1).
     05  PECB-AFTER-SHIP-CON     PIC X(1).
     05  PECB-BEFORE-COLLECT     PIC X(1).
     05  PECB-AFTER-COLLECT      PIC X(1).
*
*****************************************************************
* THE FOLLOWING FIELDS PERTAIN TO REQUEST FOR DATA PROCESSING.
*
***   THE FOLLOWING THREE FIELDS ARE INFORMATIONAL FIELDS FOR
***   THE USER EXIT TO VERIFY WHETHER OR NOT A REQUEST TYPE IS
***   AVAILABLE.  THESE FLAGS ARE SET BY ENDEVOR EXIT
***   PROCESSING PRIOR TO INVOKING THE FIRST USER EXIT MODULE.
*
     05  PECB-ACT-REC-EXIST-FLAG PIC X(1).
     05  PECB-APP-REC-EXIST-FLAG PIC X(1).
     05  PECB-BAC-REC-EXIST-FLAG PIC X(1).
*
*****************************************************************
***   THE FOLLOWING FIVE FIELDS ARE REQUESTS FOR DATA FLAGS.
***   TO REQUEST A RECORD TO BE SENT, MOVE A 'Y' INTO THE
***   APPROPIATE FIELD.  CHECK USER DOC TO MAKE SURE THAT THE
***   DATA REQUESTED EXISTS AT THE POINT THE REQUEST IS BEING
***   MADE.
*
     05  PECB-REQ-ACTION-ID-REC  PIC X(1).
```

```
     05  PECB-REQ-APPROVER-REC   PIC X(1).
     05  PECB-REQ-BACKOUT-REC    PIC X(1).
     05  PECB-REQ-SHIP-RECORDS   PIC X(1).
     05  PECB-REQ-SCL-RECORDS    PIC X(1).
     05  PECB-REQ-COLL-RECORDS    PIC X(1).
*
* THE PECB-REQUEST-RETURNCODE IS USED BY ENDEVOR TO TELL THE
* USER EXIT OF THE RESULT OF THE REQUEST FOR DATA ATTEMPT
*
     05  PECB-REQUEST-RETURNCODE PIC 9(4) COMP SYNC.

         88  PECB-SUCCESSFUL-RECORD-SENT  VALUE  1.
         88  PECB-END-OF-FILE-FOR-REC-TYP VALUE  2.
         88  PECB-NO-RECORDS-FOUND        VALUE  3.
         88  PECB-SEVERE-ERROR            VALUE 12.
*
* THE PECB-NDVR-XXXX FIELDS ARE USED BY ENDEVOR TO TELL THE
* USER EXIT ABOUT HIGH RETURN CODES, ERROR TEXT, AND ABEND
* CODES.  THESE FIELDS ARE ONLY APPLICABLE FOR AFTER CALLS.
*
     05  PECB-NDVR-HIGH-RC       PIC S9(4) COMP SYNC.
     05  PECB-NDVR-ABEND-CODE    PIC X(6).
     05  PECB-NDVR-MESSAGE-TEXT  PIC X(133).
     05  PECB-NDVR-MESSTEXT-LEN  PIC S9(4) COMP SYNC.
     05  PECB-NDVR-EXIT-RC       PIC  9(8) COMP SYNC.

         88  PECB-ALL-IS-OK           VALUE 0.
         88  PECB-ALL-IS-OK-WITH-MODS VALUE 4.
         88  PECB-ABORT-RC            VALUE 8.

     05  PECB-USER-BATCH-JOBNAME PIC X(8).
     05  PECB-MODE               PIC X(1).

         88  PECB-TSO-MODE       VALUE 'T'.
         88  PECB-BATCH-MODE     VALUE 'B'.

     05  PECB-MESSAGE-ID         PIC X(4).
     05  PECB-ERROR-MESS-LENGTH  PIC S9(4) COMP SYNC.
     05  PECB-MESSAGE            PIC X(132).
     05  PECB-CALLER-ORIGIN      PIC X(1).
         88  CALLER-ENDEVOR      VALUE 'E'.
         88  CALLER-QUICKEDIT    VALUE 'Q'.
     05  FILLER                  PIC X(1).
     05  PECB-UEXIT-HOLD-FIELD   PIC S9(8) COMP SYNC.
     05  PECB-HIGH-PREV-RC       PIC S9(8) COMP SYNC.
*
* THE FOLLOWING TWO FIELDS ARE UPDATE FLAGS FOR THE USER
* CONTROL BLOCKS.  THESE FLAGS REPLACE THE RETURN CODE OF
* ($MOK) ($) IN THE OLD EXITS.
*
     05  PECB-MODS-MADE-TO-PECB  PIC X(1).
     05  PECB-MODS-MADE-TO-PREQ  PIC X(1).
     05  PECB-MODS-MADE-TO-PFIL  PIC X(1).
     05  PECB-USENDING-APP-GRPS  PIC X(1).
     05  PECB-NBR-APPR-GRPS-SENT PIC S9(4) COMP.

*
*  ARCHIVE PACKAGE EXIT ENABLE/DISABLE FLAGS
*
```

```
*
     05  PECB-BEFORE-ARCHIVE     PIC X(1).
     05  PECB-AFTER-ARCHIVE      PIC X(1).
*
*  THE FOLLOWING FIELDS ARE USED EXCLUSIVELY DURING SETUP
*  PROCESSING AT SETUP TIME, THE INITIAL VALUES ARE AS SHOWN
*  UNLESS OVERRIDDEN IN THE C1UEXITS TABLE
*  (@C1UEXIT TYPE=START,XIT7BATX=Y).
*
     05  PECB-TSO-EXECUTE        PIC X(1).
     05  PECB-BATCH-EXECUTE      PIC X(1).
*
*          RESERVED FOR INTERNAL USE
*
     05  PECB-RSV1               PIC X(4).

     EJECT
```

6.3.2.1 Field Descriptions

Brief descriptions of the fields in the PACKAGE-EXIT-BLOCK control block layout follow.

Field	Description
PECB-LENGTH	Size of the Exit control block (PACKAGE-EXIT-BLOCK), in bytes. Not modifiable by any exits.
PECB-CURRENT-VERSION	Version number to identify the Exit control block (PACKAGE-EXIT-BLOCK) written by the exit handler. This should always be 2. It is not modifiable by exits.
PECB-BLOCK-ID	Identifies this DSECT as the Exit control block. Value is always "PECB." Not modifiable by any exits.
PECB-PACKAGE-ID	Current CA-Endevor package ID.

Note: The next six fields contain the action, function, and type codes. User exits use these values to determine which package exit has been invoked.

Field	Description
PECB-FUNCTION-CODE	Package action code. Acceptable values are 0-17.

Value	Equates	Meaning
0	SETUP-EXIT-OPTION	Setup exit options
1	BACK-IN-PACKAGE	Backin package
2	BACK-OUT-PACKAGE	Backout package
3	CAST-PACKAGE	Cast package
4	COMMIT-PACKAGE	Commit package
5	CREATE-PACKAGE	Create package
6	DELETE-PACKAGE	Delete package
7	DISPLAY-PACKAGE	Display package
8	EXECUTE-PACKAGE	Execute package
9	EXPORT-PACKAGE	Export package
10	GENERATE-PACKAGE-ID	Generate package ID
11	LIST-GENERATION	Generate list
12	REPLACE-SCL	Modify package
13	RESET-PACKAGE	Reset package
14	REVIEW-PACKAGE	Review package
15	SHIP-THE-PACKAGE	Ship package
16	ARCHIVE-PACKAGE	Archive package
17	DISTRIB-COLLECT	Collect product

Field	Description
PECB-SUBFUNCTION-CODE	Package function code.

Equates	Value	Meaning	Related Action
0	PECB-APPROVE	Approve	Review
1	PECB-DENY	Deny	Review
2	PECB-BUILD	Build	Create & Modify
3	PECB-CREATE	Copy	Create & Modify
4	PECB-EDIT	Edit	Create & Modify
5	PECB-IMPORT	Import	Create & Modify
6	PECB-DISP-APPRVER	Approver	Display
7	PECB-DISP-BACKOUT	Backout	Display
8	PECB-DISP-SCL	SCL	Display
9	PECB-DISP-PKG	Package	Display
10	PECB-DISP-ELMSUM	ELMSUM	Display
11	PECB-DISP-REPORTS	Reports	Display
12	PECB-SHIP-XMIT	XMIT	Ship
13	PECB-SHIP-CONFIRM	Confirm	Ship
14	PECB-LIST-BUILD	Build	List
15	PECB-LIST-CAST	Cast	List
16	PECB-LIST-REVIEW	Review	List
17	PECB-LIST-EXECUTE	Execute	List
18	PECB-LIST-SHIP	Ship	List
19	PECB-LIST-BACKOUT	Back out	List
20	PECB-LIST-COMMIT	Commit	List
21	PECB-LIST-UTILITY.	Utility	List
22	PECB-LIST-DISPLAY	Display	List
23	PECB-LIST-COLLECT	Collect	List
24	PECB-COLLECT-STAGE	Stage	Collect
25	PECB-DISP-CORRINFO	Corrinfo	Display
26	PECB-LIST-ARCHIVE	Archive	List
27	PECB-COLL-CONFIRM	Confirm	Collect
28	PECB-LIST-INSPECT	Inspect	List
29	PECB-PKG-BACKOUT	Back-out	Package
30	PECB-PKG-BACKIN	Backin	Package
31	PECB-PKG-DELETE	Delete	Package
32	PECB-PKG-EXPORT	Export	Package
33	PECB-PKG-RESET	Reset	Package
34	PECB-PKG-CAST	Cast	Package
35	PECB-PKG-COMMIT	Commit	Package
36	PECB-PKG-EXECUTE	Execute	Package

PECB-FUNCTION-LITERAL	Literal for requested package action. For example, CREATE.
PECB-SUBFUNC-LITERAL	Literal for requested package function. For example, IMPORT.
PECB-BEF-AFTER-LITERAL	Values are either BEFORE (for before-exits), MID (for mid-cast exits) or AFTER (for after-exits).

Note: The following fields are used during Setup. The Setup call allows an exit program to indicate whether it is designed to handle calls while operating under TSO and/or as a batch program (without TSO services) and also allows the exit program to enable those exit points which it was designed to process.

The user exit must indicate which operating environments it can handle. Acceptable values are N. (not enabled for this operating environment) and Y. (enabled).

Set this field to Y	To operate	The default is
PECB-TSO-EXECUTE	Under TSO	Y
PECB-BATCH-EXECUTE	Batch (no TSO)	N

The user exit must indicate which exit points are enabled. For an exit to get control, **both** the proper operating environment setup flag **and** the exit point setup flag must be **Y**. Acceptable values for exit point setup flags are **N** (not enabled: default) and **Y** (enabled).

Set this field to Y	To invoke this exit point
PECB-BEFOR-ARCHIVE	Before-archive
PECB-AFTER-ARCHIVE	After-archive
PECB-BEFORE-BACKIN	Before-backin
PECB-AFTER-BACKIN	After-backin
PECB-BEFORE-BACKOUT	Before-backout
PECB-AFTER-BACKOUT	After-backout
PECB-BEFORE-CAST	Before-cast
PECB-MID-CAST	Mid-cast
PECB-AFTER-CAST	After-cast
PECB-BEFORE-COLLECT	Before-collect
PECB-AFTER-COLLECT	After-collect
PECB-BEFORE-COMMIT	Before-commit
PECB-AFTER-COMMIT	After-commit
PECB-BEFORE-CREATE-BLD	Before-create/build
PECB-AFTER-CREATE-BLD	After-create/build
PECB-BEFORE-CREATE-COPY	Before-create/copy
PECB-AFTER-CREATE-COPY	After-create/copy
PECB-BEFORE-CREATE-EDIT	Before-create/edit

Set this field to Y	To invoke this exit point
PECB-AFTER-CREATE-EDIT	After-create/edit
PECB-BEFORE-CREATE-IMPT	Before-create/import
PECB-AFTER-CREATE-IMPT	After-create/import
PECB-BEFORE-DELETE	Before-delete
PECB-AFTER-DELETE	After-delete
PECB-BEFORE-DSPLY-APPR	Before-display approver
PECB-BEFORE-DSPLY-BKOUT	Before-display backout
PECB-BEFORE-DSPLY-SCL	Before Display SCL
PECB-BEFORE-DSPLY-ELMSM	Before Display element summary
PECB-BEFORE-DSPLY-PKG	Before Display packages
PECB-BEFORE-DSPLY-RPT	Before Display reports
PECB-BEFORE-EXEC	Before Execute
PECB-AFTER-EXEC	After Execute
PECB-BEFORE-EXPORT	Before Export
PECB-AFTER-EXPORT	After Export
PECB-BEFORE-GENPID	Before GENPKGID
PECB-AFTER-GENPID	After GENPKGID
PECB-BEFORE-LIST	Before List
PECB-AFTER-LIST	After List
PECB-BEFORE-MOD-BLD	Before Modify/Build
PECB-AFTER-MOD-BLD	After Modify/Build
PECB-BEFORE-MOD-CPY	Before Modify/Copy
PECB-AFTER-MOD-CPY	After Modify/Copy
PECB-BEFORE-MOD-EDIT	Before Modify/Edit
PECB-AFTER-MOD-EDIT	After Modify/Edit
PECB-BEFORE-MOD-IMPT	Before Modify/Import
PECB-AFTER-MOD-IMPT	After Modify/Import
PECB-BEFOR-RESET	Before Reset
PECB-AFTER-RESET	After Reset
PECB-BEFORE-REV-APPR	Before Review/Approve
PECB-AFTER-REV-APPR	After Review/Approve

Set this field to Y	To invoke this exit point
PECB-BEFORE-REV-DENY	Before Review/Deny
PECB-AFTER-REV-DENY	After Review/Deny
PECB-BEFORE-SHIP-XMIT	Before Ship transmit
PECB-AFTER-SHIP-XMIT	Invoke Ship confirm
PECB-BEFORE-SHIP-CON	After Ship transmit
PECB-AFTER-SHIP-CON	After Ship confirm

Note: The following fields pertain to the requests for data. CA-Endevor sets the five existence flags before calling the user exit. The user exit should check these fields before setting request for data flags:

Field	Description
PECB-ACT-REC-EXIST-FLAG	Action, element, data set records exist.
PECB-APP-REC-EXIST-FLAG	Approver group records exist.
PECB-BAC-REC-EXIST-FLAG	Backout records exist.

The next six fields are the request for data flags. CA-Endevor clears a flag each time the user exit returns to CA-Endevor. Therefore the user exit must reset the request flag each time it requests a record. An exit can request only one data block each time it returns to CA-Endevor.

Field	Description
PECB-REQ-ACTION-ID-REC	Request for action element, data set records.
PECB-REQ-APPROVER-REC	Request for approver group record.
PECB-REQ-BACKOUT-REC	Request for backout record.
PECB-REQ-SHIP-RECORDS	Request for ship record.
PECB-REQ-SCL-RECORDS	Request for SCL record.
PECB-REQ-COLL-RECORDS	Request for collection record.

Field	Description
PECB-REQ-REQUEST-RETURNCODE	Request for data return code. This field is modified by CA-Endevor to inform the user exit of the request for data results. ■ Equates: PECB-SUCCESSFUL-RECORD-SENT – Value: 1 ■ Equates: PECB-END-OF-FILE-FOR-REC-TYPE – Value: 2 ■ Equates: PECB-NO-RECORDS-FOUND – Value: 3 ■ Equates: PECB-SEVERE-ERROR – Value: 12 When an EOF or NFND condition occurs, CA-Endevor returns this information to the user exit, but does not allow further requests against that data type.
PECB-NDVR-HIGH-RC	CA-Endevor high return code.
PECB-NDVR-ABEND-CODE	CA-Endevor abend code.
PECB-NDVR-MESSAGE-TEXT	CA-Endevor message text.
PECB-NDVR-MESSTEXT-LEN	CA-Endevor message length.
PECB-NDVR-EXIT-RC	Return code. Modifiable by all exit programs. ■ Equates: PECB-ALL-IS-OK – Value: 0 – Meaning: Successful. ■ Equates: PECB-ALL-IS-OK-WITH-MODS – Value: 4 – Meaning: Successful, but the exit modified some control block data. ■ PECB-ABORT-RC – Value: 8 – Meaning: Stop package processing. This return code can only be bypassed by before-exits.
PECB-USER-BATCH-JOBNAME	User ID associated with the current processing.

Field	Description
PECB-MODE	Indicates whether the current action was requested in foreground or batch. Not modifiable by exits. ■ Equates: PECB-TSO-MODE – Value: T – Meaning: Action requested in foreground. ■ Equates: PECB-BATCH-MODE – Value: B – Meaning: Action requested in batch.
PECB-MESSAGE-ID	Modifiable by all exits. The default is 0000. Used to construct the message code written to the CA-Endevor Execution Report for the current action. A message code precedes each message written to the report, and applies if the PECBMSG field contains a non-blank value. For messages produced by user exits, the error code is formatted as C1UNNNNs, where NNNN is defined by this field, and S is a severity code associated with PECBRTCD (W, C, E, or S, respectively for return codes 4, 8, 12, and 16).
PECB-ERROR-MESS-LENGTH	Length of the error message.
PECB-MESSAGE	Error message associated with the current activity. This is blank on entry and modifiable by all exit points. If there is a message in this field when the program returns from the user exit, the exit handler records the message in the CA-Endevor Execution Report. The message is preceded in that report with an error code (C1UNNNNs format, described further above for the PECB-MESSAGE-ID field).
PECB-CALLER-ORIGIN	Exit is executing under Endevor or Quick-Edit. ■ Equates: CALLER-ENDEVOR – Value: E – Meaning: Exit is running under Endevor ■ Equates: CALLER-QUICK-EDIT – Value: T – Meaning: Exit is running under QUICK-EDIT
PECB-UEXIT-HOLD-FIELD	User field, used to save (and pass) information between user exits. Generally, this field contains an address that points to a block of user data. Modifiable by all exit points.
PECB-HIGH-PREV-RC	Highest return code set by previously executed system and user package exit for this action/function and before/after code.

Field	Description
PECB-MODS-MADE-TO-PECB	Update CA-Endevor fields related to fields in the PECBDS (value Y/N). When this flag is set, CA-Endevor copies the PACKAGE-EXIT-BLOCK fields to the correlating CA-Endevor control block fields.
PECB-MODS-MADE-TO-PREQ	Update CA-Endevor fields related to fields in the PREQPDS (value Y/N). When this flag is set, CA-Endevor copies the PACKAGE-REQUEST-BLOCK fields to the correlating CA-Endevor control block fields.
PECB-MODS-MADE-TO-PFIL	Update CA-Endevor fields related to fields in the PACKAGE-EXIT-FILE-BLOCK (value Y/N). When this flag is set, CA-Endevor copies the PACKAGE-EXIT-FILE-BLOCK fields to the correlating CA-Endevor control block fields.
PECB-USENDING-APP-GRPS	Update Approver Group Records (value Y/N). Valid only for before-cast exit call.
PECB-NBR-APPR-GRPS-SENT	Number of approver groups sent.

6.4 Package Request Control Block

6.4.1 Second Information Block

This is the second information block passed to the exit, and is pointed to by Register 1: 4(R1). The information in this block relates to the package options associated with the current package ID.

6.4.1.1 $PREQPDS DSECT

```
         MACRO                                                          00010000
&NAME    $PREQPDS &DSCT=YES                                             00020000
         AIF   ('&DSCT' NE 'YES').SKPDSCT                               00030000
$PREQPDS DSECT                                                          00040000
         AGO   .SKPEQU                                                  00050000
.SKPDSCT ANOP                                                           00060000
$PREQPDS DS    0F                                                       00070000
.SKPEQU  ANOP                                                           00071000
*********************************************************************** 00080000
*                                                                     * 00090000
*         $PREQPDS - BLOCK FOR REQUEST PARMS ASSOCIATED WITH A        * 00100000
*                    PACKAGE          7/15/90                          * 00110001
*                                                                     * 00130000
*********************************************************************** 00140000
PREQLEN   DC    AL2(PREQSIZE)      LENGTH OF THE BLOCK                  00160000
PREQVERS  DC    AL2(PREQ$VER)      CURRENT VERSION OF THE BLOCK         00170000
PREQ$VER  EQU   1                      VERSION OF THE BLOCK             00180000
PREQID    DC    CL4'PREQ'          BLOCK ID                             00190000
PREQCOMM  DC    CL50' '            PACKAGE COMMENT                      00192000
PREQFLAG  DC    F'0'               SPECIAL FLAGS                        00200000
PREQBOEN DC     CL1' '             BACKOUT ENABLED FOR PACKAGE          00210001
PREQ$BYS  EQU    C'Y'                   YES                             00211003
PREQ$BNO  EQU   C'N'                    NO                              00212003
PREQDEL   DC    CL1' '             DELETE  ENABLED FOR PACKAGE          00210001
PREQ$DYS  EQU   C'Y'                    YES                             00211003
PREQ$DNO  EQU   C'N'                    NO                              00212003
PREQPSHR  DC    CL1' '             SHARE   ENABLED FOR PACKAGE          00210001
PREQ$SYS  EQU   C'Y'                    YES                             00211003
PREQ$SNO  EQU   C'N'                    NO                              00212003
PREQPAPD  DC    CL1' '             APPEND PACKAGE ?                     00210001
PREQ$PYS  EQU   C'Y'                    YES                             00211003
PREQ$PNO  EQU   C'N'                    NO                              00212003
PREQEWSD  DC    CL7' '             EXECUTION WINDOW START DATE
PREQEWST  DC    CL5' '             EXECUTION WINDOW START TIME
PREQEWED  DC    CL7' '             EXECUTION WINDOW END DATE
PREQEWET  DC    CL5' '             EXECUTION WINDOW END TIME
PREQIPKG  DC    CL16' '            COPY SUBFUNC - SOURCE PACKAGE ID
PREQDEST  DC    CL8' '             SHIP - DESTINATION (BEF/AFT/CONF)
PREQSTYP  DC    CL3' '             SHIP - SHIPMENT TYPE (BEF/AFT)
PREQS$PKG EQU   C'P'                      PKG = PACKAGE SHIPMENT
PREQS$BKO EQU   C'B'                      BKO = BACKOUT SHIPMENT
PREQSCMP  DC    CL3' '             SHIP - COMPLEMENTS (BEF/AFT)
```

```
PREQS$CO  EQU   C'Y'                       YES = COMPLEMENTS REQUESTED
PREQS$NC  EQU   C'N'                       NO  = NO COMPLEMENT FILES
PREQSCNF  DC    CL4' '             SHIP - CONFIRMATION TYPE (CNF)
PREQ$HXMT EQU   C'HXMT'                    HOST - TRANSMITTED
PREQ$RCPY EQU   C'RCPY'                 REMOTE - COPIED TO PRODUCTION
PREQSRES  DC    CL2' '             SHIP - CONFIRMATION RESULT (CNF)
PREQ$EXEC EQU   C'EX'                      EXECUTED (RC NOT AVAILABLE)
PREQ$ABND EQU   C'AB'                      ABENDED (RC NOT AVAILABLE)
PREQ$RCGT EQU   C'GT'                      RC GREATER THAN PREQSRCV
PREQ$RCLT EQU   C'LT'                      *  LESS THAN
PREQ$RCEQ EQU   C'EQ'                      *  EQUAL
PREQ$RCGE EQU   C'GE'                      *  GREATER THAN OR EQUAL
PREQ$RCLE EQU   C'LE'                      *  LESS THAN OR EQUAL
PREQSRCV  DC    CL4' '             SHIP - CONF RETURN CODE VALUE (CNF)
PREQ$NONE EQU   C'*'                       NO RC IF PREQ$EXEC/PREQ$ABND
PREQAPGP  DC    CL16' '                    OVERRIDE APPROVER GROUP
PREQAPID  DC    CL8' '                     OVERRIDE APPROVER ID
*
PREQDARC  DC    CL1' '             DELETE PACKAGE WITH ARCHIVE
PREQ$AYS  EQU   C'Y'                    YES                            00211003
PREQ$ANO  EQU   C'N'                    NO                             00212003
          DC    CL3' '             *** RESERVED ***
PREQCONN  DC    CL8' '             COLLECT - CONNECTION (BEF/AFT)
PREQCTYP  DC    CL3' '             COLLECT - COLLECTN TYPE (BEF/AFT)
PREQC$PKG EQU   C'P'                         PKG = PACKAGE COLLECT
PREQC$BKO EQU   C'B'                         BKO = BACKOUT COLLECT
PREQCRCV  DC    CL4' '             COLLECT - SDSBAPI RETURN CODE (AFT)
PREQPROD  DC    CL20' '            COLLECT - PRODUCT NAME (BEF/AFT)
          DC    F'0'               *** RESERVED ***
PREQSIZE  EQU   *-$PREQPDS         LENGTH OF $PREQPDS BLOCK             00540000
         MEND                                                          00550000
```

6.4.1.2 Field Descriptions

Brief descriptions of the fields in the $PREQPDS DSECT follow.

Field	Description
PREQLEN	Size of the Exit control block ($PREQPDS), in bytes. Not modifiable by any exits.
PREQVERS	Version number to identify the Exit control block ($PREQPDS) written by the exit handler. This should always be 1. It is not modifiable by exits. Use the PREQ$VERS label to verify that the correct version of this DSECT was passed to your exit.
PREQID	Code that identifies this as the Exit control block, for use in rapid identification when you are scanning your modules visually. Value is always "PREQ." Not modifiable by any exit point.
PREQFLAG	Not used.
PREQCOMM	Comments associated with the current package ID. Modifiable by create and modify package exits.

Field	Description
PREQBOEN	Backout enabled flag for package. Value are: ■ Y -- PREQ$BYS. Backout is enabled. ■ N -- PREQ$BNO. Backout is not enabled.
PREQDEL	Delete enabled flag for package. Value are: ■ Y -- PREQ$DYS. Delete is enabled. ■ N -- PREQ$DNO. Delete is not enabled.
PREQPSHR	Share enabled flag for package. Value are: ■ Y -- PREQ$SYS. Share is enabled. ■ N -- PREQ$SNO. Share is not enabled.
PREQPAPD	Append SCL enabled flag for package. ■ Y -- PREQ$PYS. Append SCL is enabled. ■ N -- PREQ$PNO. Append SCL is not enabled
PREQEWSD	Execution window start date. Format *ddmmmyy*.
PREQEWST	Execution window start time. Format *hh:mm*.
PREQEWED	Execution window end date. Format *ddmmmyy*.
PREQEWET	Execution window end time. Format *hh:mm*.
PREQIPKG	Source package ID. COPY function only.
PREQDEST	Destination ID. SHIP action only.
PREQSTYP	Shipment type. Possible values are: ■ PKG--PREQS$PKG. Package shipment. ■ BKO--PREQS$BKO. Backout shipment.
PREQSCMP	Indicates whether complementary data sets are to be shipped. ■ **YES--** PREQS$CO. Complementary data sets requested. ■ **NO--** PREQS$NC. Complementary data sets not requested.
PREQSCNF	Type of confirmation. ■ **HXMT--** PREQ$HXMT. Transmitted from host to remote site. ■ **RCPY--** PREQ$RCPY. Copied from staging to production data sets at remote site.

Field	Description
PREQSRES	Shipment results. ■ **EX--** PREQ$EXEC. Shipment executed (return code not available). ■ **AB--** PREQ$ABND. Shipment abended (return code not available). ■ **GT--** PREQ$RCGT. Shipment executed with return code greater than PREQSRCV value. ■ **LT--** PREQ$RCLT. Shipment executed with return code less than PREQSRCV value. ■ **EQ--** PREQ$RCEQ. Shipment executed with return code equal to PREQSRCV value. ■ **GE--** PREQ$RCGE. Shipment executed with return code greater than or equal to PREQSRCV value. ■ **LE--** PREQ$RCLE. Shipment executed with return code less than or equal to PREQSRCV value.
PREQSRCV	Threshold return code value for shipment confirmation. *--PREQ$NONE. No return code if values returned in the PREQ$EXEC or PREQ$ABND fields.
PREQAPGP	Override approver group. (Approver groups can only be overridden from the before-cast exit point.)
PREQAPID	Override approver ID. (Approver groups can only be overridden from the before-cast exit point.)
PREQDARC	Delete package after successful archive. Modifiable by archive package exit. Values are: ■ Y -- PREQ$AYS. Delete is enabled ■ N -- PREQ$ANO. Delete is not enabled
PREQCONN	DistribuLink connection ID.
PREQSIZE	Size of the $PREQPDS DSECT. Not modifiable by any exit.
PREQCTYP	Collection type. Possible values are: ■ PKG--PKEQC$PKG. Package collection. ■ BKO--PREQC$BKO. Backout collection.
PREQCRCV	DistribuLink API (SDSBAPI) return code value.

6.4.1.3 PACKAGE-REQUEST-BLOCK control block layout

This is the second information block passed to the exit. The information in this block relates to the package options associated with the current package ID.

```
***************************************************************
*    PACKAGE-REQUEST-BLOCK - BLOCK FOR REQUEST PARMS          *
*    ASSOCIATED  WITH A PACKAGE.                              *
***************************************************************

 01  PACKAGE-REQUEST-BLOCK.
     05  PREQ-LENGTH             PIC 9(4)  COMP SYNC.
     05  PREQ-CURRENT-VERSION    PIC 9(4)  COMP SYNC.
     05  PREQ-BLOCK-ID           PIC X(4).
     05  PREQ-PACKAGE-COMMENT    PIC X(50).
     05  PREQ-SPECIAL-FLAGS      PIC S9(8) COMP SYNC.
     05  PREQ-BACKOUT-ENABLED    PIC X(1).
         88  PREQ-BACKOUT-YES    VALUE 'Y'.
         88  PREQ-BACKOUT-NO     VALUE 'N'.
     05  PREQ-DELETE-ENABLED     PIC X(1).
         88  PREQ-DELETE-YES     VALUE 'Y'.
         88  PREQ-DELETE-NO      VALUE 'N'.
     05  PREQ-SHARE-ENABLED      PIC X(1).
         88  PREQ-SHARED-YES     VALUE 'Y'.
         88  PREQ-SHARED-NO      VALUE 'N'.
     05  PREQ-APPEND-PACKAGE     PIC X(1).
         88  PREQ-APPEND-YES     VALUE 'Y'.
         88  PREQ-APPEND-NO      VALUE 'N'.
     05  PREQ-EXEC-START-DATE    PIC X(7).
     05  PREQ-EXEC-START-TIME    PIC X(5).
     05  PREQ-EXEC-END-DATE      PIC X(7).
     05  PREQ-EXEC-END-TIME      PIC X(5).
     05  PREQ-SOURCE-PACKAGE     PIC X(16).
     05  PREQ-SHIP-DESTINATION   PIC X(8).
     05  PREQ-SHIPMENT-TYPE.
         10  FILLER              PIC X(1).
         88  PREQ-PACKAGE-SHIP   VALUE 'P'.
         88  PREQ-BACKOUT-SHIP   VALUE 'B'.
         10  FILLER              PIC X(2).
     05  PREQ-SHIP-COMPLEMENTS.
         10  FILLER              PIC X(1).
         88  PREQ-COMPLEMENTS    VALUE 'Y'.
         88  PREQ-NO-COMPLEMENTS VALUE 'N'.
         10  FILLER              PIC X(2).
     05  PREQ-SHIP-COMPLE-TYPE   PIC X(4).
         88  PREQ-HOST-TRANSMIT  VALUE 'HXMT'.
         88  PREQ-REMOTE-COPIED  VALUE 'RCPY'.
     05  PREQ-SHIP-CONFIRM-RESUL PIC X(2).
         88  PREQ-SHIP-EXECUTED  VALUE 'EX'.
         88  PREQ-SHIP-ABENDED   VALUE 'AB'.
         88  PREQ-SHIP-GT-SRVC   VALUE 'GT'.
         88  PREQ-SHIP-LESS-THAN VALUE 'LT'.
         88  PREQ-SHIP-EQUAL     VALUE 'EQ'.
         88  PREQ-SHIP-MORE-EQUA VALUE 'GE'.
         88  PREQ-SHIP-LESS-EQUA VALUE 'LE'.
     05  PREQ-SHIP-CONF-RETURN-CODE.
         10  FILLER              PIC X(1).
         88  PREQ-SHIP-NO-RC     VALUE '*'.
```

```
    10  FILLER              PIC X(3).
05  PREQ-OVERRIDE-APPR-GRP  PIC X(16).
05  PREQ-OVERRIDE-APPR-ID   PIC X(8).
05  PREQ-DELETE-PKG-ARCH    PIC X(1).
    88  ARCHIVE-WITH-DELETE VALUE 'Y'.
    88  ARCHIVE-NO-DELETE   VALUE 'N'.
05  PREQ-NOTES              PIC X(1).
    88  PREQ-NOTES-YES      VALUE 'Y'.
    88  PREQ-NOTES-NO       VALUE 'N'.
05  FILLER                  PIC X(2).
05  PREQ-COLLECT-BEF-AFT    PIC X(8).
05  PREQ-COLLECT-TYPE       PIC X(1).

    88 PREQ-COLLECT-PKG     VALUE 'P'.
    88 PREQ-COLLECT-BACKOUT VALUE 'B'.

05  PREQ-COLLECT-SDSBAPI-RC PIC X(4).
05  PREQ-COLLECT-PRODUCT-NAME PIC X(20).
05  FILLER                  PIC X(4).

EJECT
```

6.4.1.4 Field Descriptions

Brief descriptions of the fields in the PACKAGE-EXIT-REQUEST-BLOCK Paragraph follow.

Field	Description
PREQ-LENGTH	Size of the Exit control block (PACKAGE-REQUEST-BLOCK), in bytes. Not modifiable by any exits.
PREQ-CURRENT-VERSION	Version number to identify the Exit control block (PACKAGE-REQUEST-BLOCK) written by the exit handler. This should always be 1. It is not modifiable by exits.
PREQ-BLOCK-ID	Code that identifies this as the Exit control block, for use in rapid identification when you are scanning your modules visually. Value is always "PREQ." Not modifiable by any exit point.
PREQ-SPECIAL-FLAGS	Not used.
PREQ-PACKAGE-COMMENT	Comments associated with the current package ID. Modifiable by create and modify package exits.

Field	Description
PREQ-BACKOUT-ENABLED	Backout enabled flag for package. Values are: ■ Equates: PREQ-BACKOUT-YES – Value: Y – Meaning: Backout is enabled. ■ Equates: PREQ-BACKOUT-NO – Value: N – Meaning: Backout is not enabled.
PREQ-DELETE-ENABLED	Delete enabled flag for package. Values are: ■ Equates: PREQ-DELETE-YES – Value: Y – Meaning: Delete is enabled. ■ Equates: PREQ-DELETE-NO – Value: N – Meaning: Delete is not enabled.
PREQ-SHARE-ENABLED	Share enabled flag for package. Values are: ■ Equates: PREQ-SHARED-YES – Value: Y – Meaning: Share is enabled. ■ Equates: PREQ-SHARED-NO – Value: N – Meaning: Share is not enabled
PREQ-APPEND-PACKAGE	Append SCL enabled flag for package. ■ Equates: PREQ-APPEND-YES – Value: Y – Meaning: Append SCL is enabled. ■ Equates: PREQ-APPEND-NO – Value: N – Meaning: Append SCL is not enabled.
PREQ-EXEC-START-DATE	Execution window start date. Format *ddmmmyy*.
PREQ-EXEC-START-TIME	Execution window start time. Format *hh:mm*.
PREQ-EXEC-END-DATE	Execution window end date. Format *ddmmmyy*.

Field	Description
PREQ-EXEC-END-TIME	Execution window end time. Format *hh:mm*.
PREQ-SOURCE-PACKAGE	Source package ID. COPY function only.
PREQ-SHIP-DESTINATION	Destination ID. SHIP action only.
PREQ-SHIPMENT-TYPE	Shipment type. Possible values are: ■ Equates: PREQ-PACKAGE-SHIP – Value: P – Meaning: Package Shipment ■ Equates: PREQ-BACKOUT-SHIP – Value: B – Meaning: Backout Shipment
PREQ-SHIP-COMPLEMENTS	Indicates whether complementary data sets are to be shipped. ■ Equates: PREQ-COMPLEMENTS – Value: Y – Meaning: Complementary data sets requested. ■ Equates: PREQ-NO-COMPLEMENTS – Value: N – Meaning: Complementary data sets not requested.
PREQ-SHIP-COMPLE-TYPE	Type of confirmation. ■ Equates: PREQ-HOST-TRANSMIT – Value: HXMT – Meaning: Transmitted from host to remote site. ■ Equates: PREQ-REMOTE-COPIED – Value: RCPY – Meaning: Copied from staging to production data sets at remote site.

Field	Description
PREQ-SHIP-CONFIRM-RESUL	Shipment results. ■ Equates: PREQ-SHIP-EXECUTED – Value: EX – Meaning: Shipment executed (return code not available). ■ Equates: PREQ-SHIP-ABENDED – Value: AB – Meaning: Shipment abended (return code not available). ■ Equates: PREQ-SHIP-GT-SRVC – Value: GT – Meaning: Shipment executed with return code greater than ■ Equates: PREQ-SHIP-LESS-THAN – Value: LT – Meaning: Shipment executed with return code less than PREQ-SHIP-CONF-RETURN-CODE value. ■ Equates: PREQ-SHIP-EQUAL – Value: EQ – Meaning: Shipment executed with return code equal to PREQ-SHIP-CONF-RETURN-CODE value. ■ Equates: PREQ-SHIP-MORE-EQUA – Value: GE – Meaning: Shipment executed with return code greater than or equal to PREQ-SHIP-CONF-RETURN-CODE value. ■ Equates: PREQ-SHIP-LESS-EQUA – Value: LE – Meaning: Shipment executed with return code less than or equal to PREQ-SHIP-CONF-RETURN-CODE value.

Field	Description
PREQ-SHIP-CONF-RETURN-CODE	Threshold return code value for shipment confirmation. ■ Equates: PREQ-SHIP-NO-RC – Value: * – Meaning: No return code if values returned in the PREQ-SHIP-EXECUTED or PREQ-SHIP-ABENDED fields.
PREQ-OVERRIDE-APPR-GRP	Override approver group. (Approver groups can only be overridden from the before-cast exit point.)
PREQ-OVERRIDE-APPR-ID	Override approver ID. (Approver groups can only be overridden from the before-cast exit point.)
PREQ-DELETE-PKG-ARCH	Delete package after successful archive. Modifiable by archive package exit. Values are: ■ Equates: ARCHIVE-WITH-DELETE – Value: Y – Meaning: Delete is enabled ■ Equates: ARCHIVE-NO-DELETE – Value: N – Meaning: Delete is not enabled

6.5 Package Header Control Block

6.5.1 Third Information Block

6.5.1.1 $PHDRDS DSECT

This is the third information block passed to the exit, and is pointed to by Register 1: 8(R1). The information in this block is general package data such as cast date/time, execution status, etc. None of the fields are modifiable by the package user exits.

```
$PHDRDS  DSECT

PHDRLEN      DS  AL2(PHDRSIZE)         LENGTH OF PACKAGE HEADER BLOCK
PHDRVERS     DS  AL2(PHDR$VER)         VERSION OF PACKAGE HEADER BLOCK
PHDR$VER     EQU 1                     VERSION 1 PACKAGE EXIT HDR BLK
PHDRID       DC  CL4'PHRD'             BLOCK ID
PHDRTYPE     DS  CL10                  PACKAGE TYPE (EMERGENCY/STD)
PHDRSTAT     DS  CL12                  PACKAGE STATUS
PHDRBOST     DS  CL10                  PACKAGE BACK-OUT  STATUS
PHDRCRD      DS  CL7                   PACKAGE CREATION DATE
PHDRCRT      DS  CL5                   PACKAGE CREATION TIME
PHDRXRC      DS  CL5                   EXEC RETURN CODE (READ-ONLY)
PHDRPUD      DS  CL7                   DATE OF LAST UPDT TO PKG SCL
PHDRPUT      DS  CL5                   TIME OF LAST UPDT TO PKG SCL
PHDRCD       DS  CL7                   CAST DATE
PHDRCT       DS  CL5                   CAST TIME
PHDRAD       DS  CL7                   FINAL APPROVAL/DENIAL DATE
PHDRAT       DS  CL5                   FINAL APPROVAL/DENIAL TIME
PHDRWSD      DS  CL7                   EXECUTION WINDOW START DATE
PHDRWST      DS  CL5                   EXECUTION WINDOW START TIME
PHDRWED      DS  CL7                   EXECUTION WINDOW END DATE
PHDRWET      DS  CL5                   EXECUTION WINDOW END TIME
PHDRXD       DS  CL7                   EXECUTION DATE
PHDRXT       DS  CL5                   EXECUTION TIME
PHDRCMD      DS  CL7                   COMMIT DATE
PHDRCMT      DS  CL5                   COMMIT TIME
PHDRBOD      DS  CL7                   PACKAGE BACK-OUT DATE
PHDRBOT      DS  CL5                   PACKAGE BACK-OUT TIME
PHDRBID      DS  CL7                   PACKAGE BACK-IN DATE
PHDRBIT      DS  CL5                   PACKAGE BACK-IN TIME
PHDREEXD     DS  CL7                   END EXECUTION DATE
PHDREEXT     DS  CL5                   END EXECUTION TIME
PHDREEXT     DS  CL5                   END EXECUTION TIME
PHDRCRUS     DS  CL8                   PACKAGE CREATE USERID
PHDRUPUS     DS  CL8                   PACKAGE UPDATE USERID
PHDRCAUS     DS  CL8                   PACKAGE CAST USERID
PHDREXUS     DS  CL8                   PACKAGE EXEC USERID
PHDRBOUS     DS  CL8                   PACKAGE BACKOUT USERID
PHDRBIUS     DS  CL8                   PACKAGE BACKIN USERID
PHDRCMUS     DS  CL8                   PACKAGE COMMIT USERID
*            DS  CL20                    ** RESERVED **
PHDRSIZE     EQU *-$PHDRDS
```

6.5.1.2 Field Descriptions

Brief descriptions of the fields in the $PHDRDS DSECT follow.

Field	Description
PHDRLEN	Size of the Exit control block ($PHDRDS), in bytes. Not modifiable by any exits.
PHDRVERS	Version number to identify the Exit control block ($PHDRDS) written by the exit handler. This should always be 1. Use the PHDR$VER label to verify that the correct version of this DSECT was passed to your exit.
PHDRID	Code that identifies this as the Package Header Exit control block, for use in rapid identification when you are scanning your modules visually. Value is always PHDR. Not modifiable by any exit.
PHDRTYPE	Package type (emergency/standard).
PHDRSTAT	Package status. This is a formatted literal field with the following possible values: ■ IN-EDIT ■ IN-APPROVAL ■ DENIED ■ APPROVED ■ IN-EXECUTION ■ EXEC-FAILED ■ EXECUTED ■ COMMITTED
PHDRBOST	Package backout status. This is a formatted character field with values of either blank or BACKED-OUT.
PHDRCRD	Package creation date. Format *ddmmmyy*.
PHDRCRT	Package creation time. Format *hh:mm*.
PHDRXRC	Package execution return code.
PHDRPUD	Date of last update to package SCL. Format *ddmmmyy*.
PHDRPUT	Time of last update to package SCL. Format *hh:mm*.
PHDRCD	Cast date. Format *ddmmmyy*.
PHDRCT	Cast time. Format *hh:mm*.
PHDRAD	Final approval/denial date. Format *ddmmmyy*.
PHDRAT	Final approval/denial time. Format *hh:mm*.

Field	Description
PHDRXD	Execution date. Format *ddmmmyy*.
PHDRXT	Execution time. Format *hh:mm*.
PHDRCMD	Commit date. Format *ddmmmyy*.
PHDRCMT	Commit time. Format *hh:mm*.
PHDRBOD	Backout date. Format *ddmmmyy*.
PHDRBOT	Backout time. Format *hh:mm*.
PHDRBID	Backin date. Format *ddmmmyy*.
PHDRBIT	Backin time. Format *hh:mm*.
PHDREEXD	End execution date. Format *ddmmmyy*.
PHDREEXT	End execution time. Format *hh:mm*.
PHDRSIZE	Size of the $PHDRDS DSECT.
PHDRCRUS	Package create user ID.
PHDRUPUS	Package modify user ID.
PHDRCAUS	Package cast user ID.
PHDREXUS	Package exec user ID.
PHDRBOUS	Package backout user ID.
PHDRBIUS	Package backin user ID.
PHDRCMUS	Package commit user ID.
PHDRWSD	Execution window start date.
PHDRWED	Execution window end date.
PHDRWST	Execution window start time.
PHDRWET	Execution window end time.

Note: The target fields in the $PACTREQ DSECT are not used.

6.5.1.3 PACKAGE-EXIT-HEADER-BLOCK control block layout

This is the third information block passed to the exit. The information in this block is general package data such as cast date/time, execution status, etc. None of the fields are modifiable by the package user exits.

```
*****************************************************************
*    PACKAGE-EXIT-HEADER-BLOCK - PACKAGE EXIT HEADER CONTROL    *
*    BLOCK.                                                     *
*****************************************************************
```

```
01  PACKAGE-EXIT-HEADER-BLOCK.
    05  PHDR-LENGTH             PIC 9(4) COMP SYNC.
    05  PHDR-CURRENT-VERSION    PIC 9(4) COMP SYNC.
    05  PHDR-BLOCK-ID           PIC X(4).
    05  PHDR-PACKAGE-TYPE       PIC X(10).
    05  PHDR-PACKAGE-STATUS     PIC X(12).
    05  PHDR-PKG-BACKOUT-STATUS PIC X(10).
    05  PHDR-PKG-CREATION-DATE  PIC X(7).
    05  PHDR-PKG-CREATION-TIME  PIC X(5).
    05  PHDR-PKG-EXEC-RC        PIC X(5).
    05  PHDR-PSCL-LAST-UPD-DATE PIC X(7).
    05  PHDR-PSCL-LAST-UPD-TIME PIC X(5).
    05  PHDR-PKG-CAST-DATE      PIC X(7).
    05  PHDR-PKG-CAST-TIME      PIC X(5).
    05  PHDR-PKG-FIN-APPR-DATE  PIC X(7).
    05  PHDR-PKG-FIN-APPR-TIME  PIC X(5).
    05  PHDR-PKG-EXEC-STRT-DATE PIC X(7).
    05  PHDR-PKG-EXEC-STRT-TIME PIC X(5).
    05  PHDR-PKG-EXEC-END-DATE  PIC X(7).
    05  PHDR-PKG-EXEC-END-TIME  PIC X(5).
    05  PHDR-PKG-EXECUTION-DATE PIC X(7).
    05  PHDR-PKG-EXECUTION-TIME PIC X(5).
    05  PHDR-PKG-COMMIT-DATE    PIC X(7).
    05  PHDR-PKG-COMMIT-TIME    PIC X(5).
    05  PHDR-PKG-BACKOUT-DATE   PIC X(7).
    05  PHDR-PKG-BACKOUT-TIME   PIC X(5).
    05  PHDR-PKG-BACKIN-DATE    PIC X(7).
    05  PHDR-PKG-BACKIN-TIME    PIC X(5).
    05  PHDR-PKG-END-EXEC-DATE  PIC X(7).
    05  PHDR-PKG-END-EXEC-TIME  PIC X(5).
    05  PHDR-PKG-CREATE-USER    PIC X(8).
    05  PHDR-PKG-UPDATE-USER    PIC X(8).
    05  PHDR-PKG-CAST-USER      PIC X(8).
    05  PHDR-PKG-EXEC-USER      PIC X(8).
    05  PHDR-PKG-BACKOUT-USER   PIC X(8).
    05  PHDR-PKG-BACKIN-USER    PIC X(8).
    05  PHDR-PKG-COMMIT-USER    PIC X(8).
    05  FILLER                  PIC X(20).

    EJECT
```

6.5.1.4 Field Descriptions

Brief descriptions of the fields in the PACKAGE-EXIT-HEADER-BLOCK Paragraph follow.

Field	Description
PHDR-LENGTH	Size of the Exit control block (PACKAGE-EXIT-HEADER-BLOCK), in bytes. Not modifiable by any exits.
PHDR-CURRENT-VERSIONS	Version number to identify the Exit control block (PACKAGE-EXIT-HEADER-BLOCK) written by the exit handler. This should always be 1.

Field	Description
PHDR-BLOCK-ID	Code that identifies this as the Package Header Exit control block, for use in rapid identification when you are scanning your modules visually. Value is always PHDR. Not modifiable by any exit.
PHDR-PACKAGE-TYPE	Package type (emergency/standard).
PHDR-PACKAGE-STATUS	Package status. This is a formatted literal field with the following possible values: ■ IN-EDIT ■ IN-APPROVAL ■ DENIED ■ APPROVED ■ IN-EXECUTION ■ EXEC-FAILED ■ EXECUTED ■ COMMITTED
PHDR-PKG-BACKOUT-STATUS	Package backout status. This is a formatted character field with values of either blank or BACKED-OUT.
PHDR-PKG-CREATION-DATE	Package creation date. Format *ddmmmyy*.
PHDR-PKG-CREATION-TIME	Package creation time. Format *hh:mm*.
PHDR-EXEC-RC	Package execution return code.
PHDR-PSCL-LAST-UPD-DATE	Date of last update to package SCL. Format *ddmmmyy*.
PHDR-PSCL-LAST-UPD-TIME	Time of last update to package SCL. Format *hh:mm*.
PHDR-PKG-CAST-DATE	Cast date. Format *ddmmmyy*.
PHDR-PKG-CAST-TIME	Cast time. Format *hh:mm*.
PHDR-PKG-FIN-APPR-DATE	Final approval/denial date. Format *ddmmmyy*.
PHDR-PKG-FIN-APPR-TIME	Final approval/denial time. Format *hh:mm*.
PHDR-PKG-EXEC-STRT-DATE	Execution date. Format *ddmmmyy*.
PHDR-PKG-EXEC-STRT-TIME	Execution time. Format *hh:mm*.
PHDR-PKG-COMMIT-DATE	Commit date. Format *ddmmmyy*.
PHDR-PKG-COMMIT-TIME	Commit time. Format *hh:mm*.
PHDR-PKG-BACKOUT-DATE	Backout date. Format *ddmmmyy*.
PHDR-PKG-BACKOUT-TIME	Backout time. Format *hh:mm*.
PHDR-PKG-BACKIN-DATE	Backin date. Format *ddmmmyy*.

Field	Description
PHDR-PKG-BACKIN-TIME	Backin time. Format *hh:mm*.
PHDR-PKG-END-EXEC-DATE	End execution date. Format *ddmmmyy*.
PHDR-PKG-END-EXEC-TIME	End execution time. Format *hh:mm*.
PHDR-PKG-CREATE-USER	Package create user ID.
PHDR-PKG-UPDATE-USER	Package modify user ID.
PHDR-PKG-CAST-USER	Package cast user ID.
PHDR-PKG-EXEC-USER	Package exec user ID.
PHDR-PKG-BACKOUT-USER	Package backout user ID.
PHDR-PKG-BACKIN-USER	Package backin user ID.
PHDR-PKG-COMMIT-USER	Package commit user ID.

6.6 Package File Control Block

6.6.1 Fourth Information Block

This is the fourth information block passed to the exit. The information in this block includes external data set and DDnames.

6.6.1.1 $PFILDS DSECT

```
$PFILDS  DSECT

PFILLEN   DC    AL2(PFILSIZE)     LENGTH OF THE BLOCK
PFILVERS  DC    AL2(PFIL$VER)     VERSION OF THE BLOCK
PFIL$VER  EQU   2                     VERSION 2
PFILID    DC    CL4'PFIL'          BLOCK ID
PFILDSN   DS    CL44' '            DATASET NAME
PFILMEM   DS    CL10' '            MEMBER NAME
PFILDDN   DS    CL8                DDNAME
PFILDSTY  DS    CL3' '             DATASET TYPE( DA,PSU,PO,IS,...)
                                   FOR HFS (H@, H4, H/,HW)
          DC    CL21' '            ** RESERVED **
SPACE 1

*  OFFSET (FROM THE BEGINNING OF $PFILDS) TO THE PATH AND FILE NAME AREAS.
*  THE FIRST 2 BYTES IN THE AREA = THE LENGTH OF THE NAME.  THIS IS THEN
*  FOLLOWED BY THE NAME.
*
PFILPAOFF DS    H                  PATH AREA OFFSET
PFILNAOFF DS    H                  NAME AREA OFFSET
PFILBFAREA DS   XL1027             AREA BUFFER SPACE
PFILSIZE  EQU   *-$PFILDS          LENGTH OF THE $FILDS BLOCK
         MEND
```

6.6.1.2 Field Descriptions

Brief descriptions of the fields in the $PFILDS DSECT follow.

Field	Description
PFILLEN	Size of the Exit control block ($PFILDS), in bytes. Not modifiable by any exits.
PFILVERS	Version number to identify the Exit control block ($PFILDS) written by the exit handler. This should always be 2. Use the PFIL$VER label to verify that the correct version of this was passed to your exit. Not modifiable by any exits.

Field	Description
PFILID	Code that identifies this as the File Exit control block, for use in rapid identification when you are scanning your modules visually. Value is always PFIL. Not modifiable by any exit.
PFILDSN	Data set name associated with the external file used by the current package.
PFILMEM	Member name associated with the element for which the action applies, within the above data set. Applicable when data set is a library, blank otherwise.
PFILDDN	DDname associated with the data set named above. If you change this field, the exit program must allocate the file (data set name) associated with the new DDname.
PFILDSTY	Type of data set named by the PFILDSN field; DA, PSU, PO, IS and so forth.
PFILPAOFF	If the data set type is HFS, this field will contain a binary offset value from $PFILDS to a path name area. The area is composed of 2 parts; a 2 byte binary path name length value, followed by a name. The space is taken out of the PFILBFAREA buffer.
PFILNAOFF	If the data set type is HFS, this field will contain a binary offset value from $PFILDS to a file name area. The area is composed of 2 parts; a 2 byte binary file name length value, followed by it's name. The space is taken out of the PFILBFAREA buffer.
PFILBFAREA	Contains space for $PFILDS areas.
PFILSIZE	Size of the $PFILDS DSECT.

6.6.1.3 PACKAGE-EXIT-FILE-BLOCK control block layout

```
*****************************************************************
*    PACKAGE-EXIT-FILE-BLOCK - FOR INFORMATION ASSOCIATED WITH  *
*    PACKAGE FUNCTIONS.                                         *
*****************************************************************

 01  PACKAGE-EXIT-FILE-BLOCK.
     05  PFIL-LENGTH            PIC 9(4) COMP SYNC.
     05  PFIL-CURRENT-VERSION   PIC 9(4) COMP SYNC.
     05  PFIL-BLOCK-ID          PIC X(4).
     05  PFIL-DATASET-NAME      PIC X(44).
     05  PFIL-MEMBER-NAME       PIC X(10).
     05  PFIL-DD-NAME           PIC X(8).
     05  PFIL-DATASET-TYPE      PIC X(3).
     05  FILLER                 PIC X(21).
*****************************************************************
* OFFSET (FROM BEGINNING OF PACKAGE-EXIT-FILE-BLOCK) TO THE     *
* PATH AND FILE NAME AREA.                                      *
*    THE FIRST 2 BYTES IN AN AREA = THE LENGTH OF THE NAME.     *
```

```
*    THE NAME VALUE THEN FOLLOWS                                *
*****************************************************************
     05  PFIL-PATH-AREA-OFFSET   PIC 9(4) COMP SYNC.
     05  PFIL-NAME-AREA-OFFSET   PIC 9(4) COMP SYNC.
     05  PFIL-AREA-BUFFER        PIC X(1027).

     EJECT
```

6.6.1.4 Field Descriptions

Brief descriptions of the fields in the PACKAGE-EXIT-FILE-BLOCK control block layout follow.

Field	Description
PFIL-LENGTH	Size of the Exit control block (PACKAGE-EXIT-FILE-BLOCK), in bytes. Not modifiable by any exits.
PFIL-CURRENT-VERSION	Version number to identify the Exit control block (PACKAGE-EXIT-FILE-BLOCK) written by the exit handler. This should always be 2. Not modifiable by any exits.
PFIL-BLOCK-ID	Code that identifies this as the File Exit control block, for use in rapid identification when you are scanning your modules visually. Value is always PFIL. Not modifiable by any exit.
PFIL-DATASET-NAME	Data set name associated with the external file used by the current package.
PFIL-MEMBER-NAME	Member name associated with the element for which the action applies, within the above data set. Applicable when data set is a library, blank otherwise.
PFIL-DD-NAME	DDname associated with the data set named above. If you change this field, the exit program must allocate the file (data set name) associated with the new DDname.
PFIL-DATASET-TYPE	Type of data set named by the PFILDSN field; DA, PSU, PO, IS and so forth.
PFIL-PATH-AREA-OFFSET	If the data set type is HFS, this field will contain a binary offset value from PACKAGE-EXIT-FILE-BLOCK to a path name area. The area is composed of 2 parts; a 2 byte binary path name length value, followed by a name. The space is taken out of the PFILBFAREA buffer.
PFIL-PATH-NAME-OFFSET	If the data set type is HFS, this field will contain a binary offset value from PACKAGE-EXIT-FILE-BLOCK to a file name area. The area is composed of 2 parts; a 2 byte binary file name length value, followed by it's name. The space is taken out of the PFIL-AREA-BUFFER buffer.
PFIL-AREA-BUFFER	Contains space for PACKAGE-EXIT-FILE-BLOCK areas.

6.7 Package Action Request Block

6.7.1 Fifth Information Block

6.7.1.1 $PACTREQ DSECT

This is the fifth information block passed to the exit, and is pointed to by Register 1: 16(R1). This block is built only on request from the user exit. The block contains target and source element data such as action, version and level numbers of the elements involved, etc. None of the fields are modifiable by user exits.

```
         MACRO                                                          00010000
         $PACTREQ &DSCT=YES                                             00020003
         AIF ('&DSCT' EQ 'YES').DSCT1                                   00280000
$PACTREQ     DS   0D                                                    00290000
         AGO .START1                                                    00300000

$PACTREQ     DSECT                                                      00320000

.START1  ANOP                                                           00330000

***                                                                     00670011

***  BLOCK INITIALIZATION DATA                                          00680011

***                                                                     00690011

PACTLEN      DC  AL2(PACTSIZE)               BLOCK SIZE

PACTVERS     DC  AL2(PACT$VER)              BLOCK VERSION

PACT$VER     EQU 1

PACTID       DC  CL4'PACT'                   BLOCK ID

***                                                                     00670011

***  BLOCK TOTAL DATA                                                   00680011

***                                                                     00690011

PACTSEQ#     DS  H                           SEQ NUM OF THIS ENTRY

***                                                                     01750004

***  FORMATTED ACTION FIELDS                                            01750000

***                                                                     01770004

PACTACTN     DS  CL8                   C1 ACTION                        01780012

PACTCCID     DS  CL12                  ACTION CCID                      01780012

PACTCOMM     DS  CL40                  ACTION COMMENT                   01780012
```

```
PACTNDRC     DS  CL5                  HIGH ENDEVOR/MVS RETURN CODE     01790012
PACTPRRC     DS  CL5                  HIGH PROCESSOR   RETURN CODE     01800012
PACTBEXD     DS  CL7                  BEGIN EXECUTION DATE             01800000
PACTBEXT     DS  CL5                  BEGIN EXECUTION TIME             01810000
PACTEEXD     DS  CL7                  END   EXECUTION DATE             01820000
PACTEEXT     DS  CL5                  END   EXECUTION TIME             01830000
***                                                                   00670011
***  SOURCE ELEMENT/MEMBER ID FIELDS                                  00680011
***                                                                   00690011
PACTSSIT    DS  CL1                 SITE ID                           00840011
PACTSENV    DS  CL8                 ENVIRONMENT                       00850011
PACTSSYS    DS  CL8                 SYSTEM                            00860011
PACTSSBS    DS  CL8                 SUBSYSTEM                         00870011
PACTSELM    DS  CL10                ELEMENT                           00880011
PACTSTYP    DS  CL8                 TYPE                              00890011
PACTSSTG    DS  CL8                 STAGE NAME                        00900011
PACTSSTI    DS  CL1                 STG ID (TEMPORARY)                00910012
         SPACE ,                                                      00980000
PACTSVL      DS  CL5                  VERSION.LEVEL NUMBER            01810011
PACTSDD      DS  CL7                  DELTA DATE (OR BASE IF LVL 0)   01820016
PACTSDT      DS  CL5                  DELTA TIME (OR BASE IF LVL 0)   01830016
PACTSGD      DS  CL7                  GENERATE DATE                   01840011
PACTSGT      DS  CL5                  GENERATE TIME                   01850011
PACTSPD      DS  CL7                  LAST PROCESS DATE               01860011
PACTSPT      DS  CL5                  LAST PROCESS TIME               01870011
         SPACE ,                                                      00980000
***                                                                   01000000
***  EXTERNAL DATA SET ID FIELDS                                      01010011
***                                                                   01020000
PACTSDSN     DS  CL44                 EXTERNAL DATA SET NAME          01030011
PACTSMBR     DS  CL10                 EXTERNAL MEMBER NAME            01040011
```

```
            SPACE ,                                                  01080023
PACTSPPI     DS  CL16                 PREV PKG ID ASSOCIATED W/SRC    01090023
PACTSCTS     DS  FL8                  CAST TIMESTAMP ASSOCIATED W/TGT 01530023
         SPACE 2                                                     01110011
*           ************************************                     01120011
*           **    PACKAGE TARGET ID FIELDS    **                     01130011
*           ************************************                     01140011
***                                                                  01160011
***  TARGET ELEMENT/MEMBER ID FIELDS                                 01170011
***                                                                  01180011
PACTTSIT    DS  CL1                  SITE ID                         01260011
PACTTENV    DS  CL8                  ENVIRONMENT                     01270011
PACTTSYS    DS  CL8                  SYSTEM                          01280011
PACTTSBS    DS  CL8                  SUBSYSTEM                       01290011
PACTTELM    DS  CL10                 ELEMENT                         01300011
PACTTTYP    DS  CL8                  TYPE                            01310011
PACTTSTG    DS  CL8                  STAGE NAME                      01320011
PACTTSTI    DS  CL1                  STG ID (TEMPORARY)              01330012
         SPACE ,                                                     01410011
PACTT41     DS  CL41                 RESERVED FOR FUTURE USE         01910011
***                                                                  01430011
***  EXTERNAL DATA SET ID FIELDS                                     01440011
***                                                                  01450011
PACTTDSN     DS  CL44                 EXTERNAL DATA SET NAME          01460011
PACTTMBR     DS  CL10                 EXTERNAL MEMBER NAME            01470011
PACTTPPI     DS  CL16                 PREV PKG ID ASSOCIATED W/TGT    01520023
PACTTCTS     DS  FL8                  CAST TIMESTAMP ASSOCIATED W/TGT 01530023
***                                                                  01430011
***  SOURCE AND TARGET AREA OFFSET FIELDS                            01430011
***                                                                  01430011
PACTSFLAG    DS  CL1                  SRC FILE TYPE INDICATOR
```

```
PACTSF_NA    EQU C' '                      - N/A  (NONE)
PACTSF_DSN   EQU C'D'                      - DSN, SEE PACTSDSN & PACTSMBR
PACTSF_PATH  EQU C'P'                      - PATH, INFO IN PATH AREA
*
PACTTFLAG    DS  CL1                   TGT FILE TYPE INDICATOR
PACTTF_NA    EQU C' '                      - N/A  (NONE)
PACTTF_DSN   EQU C'D'                      - DSN, SEE PACTSDSN & PACTSMBR
PACTTF_PATH  EQU C'P'                      - PATH, INFO IN PATH AREA
*
*    OFFSET ARE CALCULATED FROM THE BEGINNING OF $PACTREQ
*    OFFSET VALUE + $PACTREQ ADDRESS = AREA ADDRESS.
*
*    FIRST TWO BYTES W/IN THE AREA = THE HEX LENGTH OF THE DATA.
*    THE DATA FOLLOWS IMMEDIATELY AFTER THE LENGTH FIELD.
*
PACTSEAOFF   DS  H                     SRC ELEMENT AREA OFFSET
PACTSPAOFF   DS  H                     SRC PATH AREA OFFSET
PACTSNAOFF   DS  H                     SRC FILE NAME AREA OFFSET
         SPACE 1
PACTTEAOFF   DS  H                     TGT ELEMENT AREA OFFSET
PACTTPAOFF   DS  H                     TGT PATH AREA OFFSET
PACTTNAOFF   DS  H                     TGT FILE NAME AREA OFFSET
         SPACE 1
PACTBFAREA   DS  XL2600                AREA BUFFER SPACE
PACTSIZE     EQU  *-$PACTREQ           LENGTH OF BLOCK
         MEND                                                          02080008
```

6.7.1.2 Field Descriptions

Brief descriptions of the fields in the $PACTREQ DSECT follow.

Field	Description
PACTLEN	Size of the Exit control block ($PACTREQ), in bytes. Not modifiable by any exits.
PACTVERS	Version number to identify the Exit control block ($PACTREQ) written by the exit handler. This should always be 1. Use the PACT$VER label to verify that the correct version of this DSECT was passed to your exit. Not modifiable by any exits.
PACTID	Code that identifies this as the Package Action request block, for use in rapid identification when you are scanning your modules visually. Value is always PACT. Not modifiable by any exit.
PACTSEQ#	Sequential number set by Endevor to tell the user exit the number of records it has requested. Output only. Not modifiable.
Action fields	
PACTACTN	Endevor action literal.
PACTCCID	Endevor action CCID.
PACTCOMM	Endevor action comment.
PACTNDRC	High Endevor return code.
PACTPRRC	High processor return code.
PACTBEXD	Beginning execution date for this element. Format *ddmmmyy*.
PACTBEXT	Beginning execution time for this element. Format *hh:mm*.
PACTEEXD	Ending execution date for this element. Format *ddmmmyy*.
PACTEEXT	Ending execution time for this element. Format *hh:mm*.
Source element/member ID fields:	
PACTSSIT	Site ID of source element.
PACTSENV	Environment of source element.
PACTSSYS	System name of source element.
PACTSSBS	Subsystem of source element.
PACTSELM	Source element name.
PACTSTYP	Type name of source element.
PACTSSTG	Stage name of source element.
PACTSSTI	Stage ID of source element.
PACTSVL	Version and level number of the source element.
PACTSDD	Delta date of source element.
PACTSDT	Delta time of source element. Format *hh:mm*.

Field	Description
PACTSGD	Date the source element was last generated. Format *ddmmmyy*.
PACTSGT	Time the source element was last generated. Format *hh:mm*.
PACTSPD	Last process date of source element. Format *ddmmmyy*.
PACTSPT	Last process time of source element. Format *hh:mm*.
PACTSDSN	External data set name associated with source element.
PACTSMBR	External member name associated with source element.
PACTSPPI	Previous package ID associated with source element.
Target element/member ID fields:	
PACTTSIT	Site ID of target element.
PACTTENV	Environment of target element.
PACTTSYS	System name of target element.
PACTTSBS	Subsystem of target element.
PACTTELM	Target element name.
PACTTTYP	Type name of target element.
PACTTSTG	Stage name of target element.
PACTTSTI	Stage ID of source element.
PACTT41	Reserved for future use.
PACTTDSN	External data set name associated with target element.
PACTTMBR	External member name associated with target element.
PACTTPPI	Previous package ID associated with target element.
PACTTCTS	Cast time stamp associated with target element.
PACTSFLAG	Source file type indicator. ■ Blank—None ■ D—Data set fields: see fields PACTSDSN and PACTSMBR. ■ P—Path, see path and file name areas:PACTSPAOFF and PACTSNAOFF.
PACTTFLAG	Target file type indicator. ■ Blank—None ■ D—Data set fields: see fields PACTTDSN and PACTTMBR. ■ P—Path, see path and file name areas:PACTTPAOFF and PATTNAOFF.

Field	Description
PACTSEAOFF	Source element area binary offset from beginning of $PACTREQ. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACTSPAOFF	Source path area binary offset from beginning of $PACTREQ. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACTSNAOFF	Source file name area binary offset from beginning of $PACTREQ. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACTTEAOFF	Target element area binary offset from beginning of $PACTREQ. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACTTPAOFF	Target path area binary offset from beginning of $PACTREQ. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACTTNAOFF	Target file name area binary offset from beginning of $PACTREQ. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACTBFAREA	Space where all area is built
PACTSIZE	Size of the $PACTREQ.

6.7.1.3 PACKAGE-EXIT-ACTION-BLOCK control block layout

This is the fifth information block passed to the exit. This block is built only on request from the user exit. The block contains target and source element data such as action, version and level numbers of the elements involved, etc. None of the fields are modifiable by user exits.

```
*****************************************************************
*    PACKAGE-EXIT-ACTION-BLOCK                                  *
*****************************************************************

 01  PACKAGE-EXIT-ACTION-BLOCK.
     05  PACT-LENGTH             PIC 9(4)  COMP SYNC.
     05  PACT-CURRENT-VERSION    PIC 9(4)  COMP SYNC.
     05  PACT-BLOCK-ID           PIC X(4).

*
*    BLOCK TOTAL DATA
*

     05  PACT-SEQ-NBR-FOR-ENTRY  PIC S9(4) COMP SYNC.

*
*    FORMATTED ACTION FIELDS
```

```
*

     05  PACT-C1-ACTION          PIC X(8).
     05  PACT-ACTION-CCID        PIC X(12).
     05  PACT-COMMENT            PIC X(40).
     05  PACT-HIGH-NDVR-RC       PIC X(5).
     05  PACT-HIGH-PROCESSOR-RC  PIC X(5).
     05  PACT-BEGIN-EXEC-DATE    PIC X(7).
     05  PACT-BEGIN-EXEC-TIME    PIC X(5).
     05  PACT-END-EXEC-DATE      PIC X(7).
     05  PACT-END-EXEC-TIME      PIC X(5).

*****************************************
*    SOURCE ELEMENT/MEMBER ID FIELDS      *
*****************************************

     05  PACT-S-SITE-ID          PIC X(1).
     05  PACT-S-ENVIRONMENT      PIC X(8).
     05  PACT-S-SYSTEM           PIC X(8).
     05  PACT-S-SUBSYSTEM        PIC X(8).
     05  PACT-S-ELEMENT          PIC X(10).
     05  PACT-S-TYPE             PIC X(8).
     05  PACT-S-STAGE-NAME       PIC X(8).
     05  PACT-S-STAGE-ID         PIC X(1).

     05  PACT-S-VERSION-LEVEL    PIC X(5).
     05  PACT-S-DELTA-DATE       PIC X(7).
     05  PACT-S-DELTA-TIME       PIC X(5).
     05  PACT-S-GENERATE-DATE    PIC X(7).
     05  PACT-S-GENERATE-TIME    PIC X(5).
     05  PACT-S-LST-PROCESS-DATE PIC X(7).
     05  PACT-S-LST-PROCESS-TIME PIC X(5).

*
*    EXTERNAL DATASET ID FIELDS
*

     05  PACT-S-DATASET-NAME     PIC X(44).
     05  PACT-S-MEMBER-NAME      PIC X(10).

     05  PACT-S-PREV-PKG-ASSOC   PIC X(16).
     05  PACT-S-CAST-TIMESTAMP   PIC X(8).

*****************************************
*    TARGET ELEMENT/MEMBER ID FIELDS      *
*****************************************

     05  PACT-T-SITE-ID          PIC X(1).
     05  PACT-T-ENVIRONMENT      PIC X(8).
     05  PACT-T-SYSTEM           PIC X(8).
     05  PACT-T-SUBSYSTEM        PIC X(8).
     05  PACT-T-ELEMENT          PIC X(10).
     05  PACT-T-TYPE             PIC X(8).
     05  PACT-T-STAGE-NAME       PIC X(8).
     05  PACT-T-STAGE-ID         PIC X(1).

     05  FILLER                  PIC X(41).

*****************************************************************
```

```
*  THE FOLLOWING FIELDS WERE REMOVED AS THEY DO NOT EXIST IN   *
*  THE ASSEMBLER VERSION, NOR ARE THEY USED IN THE CODE        *
***************************************************************
*    05  PACT-T-VERSION-LEVEL    PIC X(5).                     *
*    05  PACT-T-DELTA-DATE       PIC X(7).                     *
*    05  PACT-T-DELTA-TIME       PIC X(5).                     *
*    05  PACT-T-GENERATE-DATE    PIC X(7).                     *
*    05  PACT-T-GENERATE-TIME    PIC X(5).                     *
*    05  PACT-T-LST-PROCESS-DATE PIC X(7).                     *
*    05  PACT-T-LST-PROCESS-TIME PIC X(5).                     *
***************************************************************
*
*    EXTERNAL DATASET ID FIELDS
*

     05  PACT-T-DATASET-NAME     PIC X(44).
     05  PACT-T-MEMBER-NAME      PIC X(10).

     05  PACT-T-PREV-PKG-ASSOC   PIC X(16).
     05  PACT-T-CAST-TIMESTAMP   PIC X(8).

***************************************************************
* SOURCE AND TARGET AREA OFFSET FIELDS                        *
***************************************************************
     05  PACT-S-FILE-TYPE        PIC X(1).
         88  PACT-SF-NONE        VALUE ' '.
         88  PACT-SF-DSN         VALUE 'D'.
         88  PACT-SF-PATH        VALUE 'P'.

     05  PACT-T-FILE-TYPE        PIC X(1).
         88  PACT-TF-NONE        VALUE ' '.
         88  PACT-TF-DSN         VALUE 'D'.
         88  PACT-TF-PATH        VALUE 'P'.

***************************************************************
* OFFSETS ARE CALCULATED FROM THE BEGINNING OF                *
* PACKAGE-EXIT-ACTION-BLOCK.                                  *
* OFFSET VALUE + PACKAGE-EXIT-ACTION-BLOCK ADDRESS = AREA ADDR.*
*                                                             *
* FIRST TWO BYTES W/IN AREA = THE BINARY LENGTH OF THE DATA.  *
* THE DATA FOLLOWS IMMEDIATELY AFTER THE LENGTH FIELD.        *
***************************************************************

     05  PACT-SRC-ELM-AREA-OFFSET   PIC 9(4) COMP SYNC.
     05  PACT-SRC-PATH-AREA-OFFSET  PIC 9(4) COMP SYNC.
     05  PACT-SRC-NAME-AREA-OFFSET  PIC 9(4) COMP SYNC.

     05  PACT-TGT-ELM-AREA-OFFSET   PIC 9(4) COMP SYNC.
     05  PACT-TGT-PATH-AREA-OFFSET  PIC 9(4) COMP SYNC.
     05  PACT-TGT-NAME-AREA-OFFSET  PIC 9(4) COMP SYNC.

     05  PACT-AREA-BUFFER           PIC X(2600).

     EJECT
```

6.7.1.4 Field Descriptions

Brief descriptions of the fields in the $PACTREQ DSECT follow.

Field	Description
PACT-LENGTH	Size of the Exit control block (PACKAGE-EXIT-ACTION-BLOCK), in bytes. Not modifiable by any exits.
PACT-CURRENT-VERSION	Version number to identify the Exit control block (PACKAGE-EXIT-ACTION-BLOCK) written by the exit handler. This should always be 1. Use the PACT$VER label to verify that the correct version of this DSECT was passed to your exit. Not modifiable by any exits.
PACT-BLOCK-ID	Code that identifies this as the Package Action request block, for use in rapid identification when you are scanning your modules visually. Value is always PACT. Not modifiable by any exit.
PACT-SEQ-NBR-FOR-ENTRY	Sequential number set by Endevor to tell the user exit the number of records it has requested. Output only. Not modifiable.
Action fields	
PACT-C1-ACTION	Endevor action literal.
PACT-ACTION-CCID	Endevor action CCID.
PACT-COMMENT	Endevor action comment.
PACT-HIGH-NDVR-RC	High Endevor return code.
PACT-HIGH-PROCESSOR-RC	High processor return code.
PACT-BEGIN-EXEC-DATE	Beginning execution date for this element. Format *ddmmmyy*.
PACT-BEGIN-EXEC-TIME	Beginning execution time for this element. Format *hh:mm*.
PACT-END-EXEC-DATE	Ending execution date for this element. Format *ddmmmyy*.
PACT-END-EXEC-TIME	Ending execution time for this element. Format *hh:mm*.
Source element/ member ID fields:	
PACT-S-SITE-ID	Site ID of source element.
PACT-S-ENVIRONMENT	Environment of source element.
PACT-S-SYSTEM	System name of source element.
PACT-S-SUBSYSTEM	Subsystem of source element.
PACT-S-ELEMENT	Source element name.
PACT-S-TYPE	Type name of source element.
PACT-S-STAGE-NAME	Stage name of source element.

Field	Description
PACT-S-STAGE-ID	Stage ID of source element.
PACT-S-VERSION-LEVEL	Version and level number of the source element.
PACT-S-DELTA-DATE	Delta date of source element.
PACT-S-DELTA-TIME	Delta time of source element. Format *hh:mm*.
PACT-S-GENERATE-DATE	Date the source element was last generated. Format *ddmmmyy*.
PACT-S-GENERATE-TIME	Time the source element was last generated. Format *hh:mm*.
PACT-S-LST-PROCESS-DATE	Last process date of source element. Format *ddmmmyy*.
PACT-S-LST-PROCESS-TIME	Last process time of source element. Format *hh:mm*.
PACT-S-DATASET-NAME	External data set name associated with source element.
PACT-S-MEMBER-NAME	External member name associated with source element.
PACT-S-PREV-PKG-ASSOC	Previous package ID associated with source element.
PACT-S-CAST-TIMESTAMP	Cast time stamp associated with source element.
Target element/ member ID fields:	
PACT-T-SITE-ID	Site ID of target element.
PACT-T-ENVIRONMENT	Environment of target element.
PACT-T-SYSTEM	System name of target element.
PACT-T-SUBSYSTEM	Subsystem of target element.
PACT-T-ELEMENT	Target element name.
PACT-T-TYPE	Type name of target element.
PACT-T-STAGE-NAME	Stage name of target element.
PACT-T-STAGE-ID	Stage ID of target element.
PACT-T-DATASET-NAME	External data set name associated with target element.
PACT-T-MEMBER-NAME	External member name associated with target element.
PACT-T-PREV-PKG-ASSOC	Previous package ID associated with target element.
PACT-S-FILE-TYPE	Source file type indicator. ■ Blank—None ■ D—Data set fields: see fields PACT-S-DATASET-NAME and PACT-S-MEMBER-NAME. ■ P—Path, see path and file name areas: PACT-SRC-PATH-AREA-OFFSET and PACT-SRC-NAME-AREA-OFFSET.

Field	Description
PACT-T-FILE-TYPE	Target file type indicator. ■ Blank—None ■ D—Data set fields: see fields PACT-T-DATASET-NAME and PACT-T-MEMBER-NAME. ■ P—Path, see path and file name areas: PACT-TGT-PATH-AREA-OFFSET AND PACT-TGT-NAME-AREA-OFFSET. PATTNAOFF.
PACT-SRC-ELM-AREA-OFFSET	Source element area binary offset from beginning of PACKAGE-EXIT-ACTION-BLOCK. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACT-SRC-PATH-AREA-OFFSET	Source path area binary offset from beginning of PACKAGE-EXIT-ACTION-BLOCK. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACT-SRC-NAME-AREA-OFFSET	Source file name area binary offset from beginning of PACKAGE-EXIT-ACTION-BLOCK. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACT-TGT-ELM-AREA-OFFSET	Target element area binary offset from beginning of PACKAGE-EXIT-ACTION-BLOCK. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACT-TGT-PATH-AREA-OFFSET	Target path area binary offset from beginning of PACKAGE-EXIT-ACTION-BLOCK. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACT-TGT-NAME-AREA-OFFSET	Target file name area binary offset from beginning of PACKAGE-EXIT-ACTION-BLOCK. Area composed of 2 parts; a binary name length value, immediately followed by the name.
PACT-AREA-BUFFER	Space where all areas are built.
PACT-T-CAST-TIME-STAMP	Cast time stamp associated with target element.

6.8 Package Approver Request Block

6.8.1 Sixth Information Block

This is the sixth information block passed to the exit. This block is used both as an input block (before-cast) and as a output block (request for data). In both cases the block is used only on the request of the user exit. The block contains internal approver group record information. There can be 1 group and up to 16 internal approver IDs per group. As an input block, certain fields are required; these will be marked with an "(RI)"--required input.

6.8.1.1 $PAPPREQ DSECT

```
         MACRO                                                          00010000
         $PAPPREQ &DSCT=YES                                             00020000
*           ************************************                        00030000
*           **     PACKAGE EXIT APPROVER MAP  **                        00040000
*           ************************************                        00050000
         AIF ('&DSCT' EQ 'YES').DSCT1                                   00290000
$PAPPREQ DS   0D                                                        00300000
         AGO .START1                                                 00310000

$PAPPREQ DSECT                                                          00330000

.START1  ANOP                                                           00330100

*           ************************************                        00330200

*           **     BLOCK INTIALIZATION FIELDS **                        00330300

*           ************************************                        00330400

PAPPLEN     DC  AL2(PAPPSIZE)                                           00331000

PAPPVERS    DC  AL2(PAPP$VER)                                           00332000

PAPP$VER   EQU  1                                                       00333000

PAPPID      DC  CL4'PAPP'                                               00334000

*           ************************************                        00335000

*           **     BLOCK TOTAL DATA           **                        00336000

*           ************************************                        00337000

PAPPSEQ#    DS  H                     SEQUENTIAL NUMBER OF THIS ENTRY   00339000

*           ************************************                        00370000

*           **     APPROVER RECORD FIELDS     **                        00380000

*           ************************************                        00390000
```

```
         SPACE 1                                                        00400000
PAPPGPNM     DS  CL16                  GROUP NAME                       00410000
PAPPEVNM     DS  CL8                   ENVIRONMENT NAME                 00410100
PAPPQUOR     DS  H                     REQUIRED APPROVER QUORUM COUNT   00411000
PAPPFLAG     DS  CL8                   OVERALL PACKAGE APPROVER FLAG    00420000
PAPPAPP#     DS  H                  NBR OF APPROVERS IN THE GROUP       00460000
**                                                                      00470000
** THE FOLLOWING FIELDS OCCUR PAPPAPP# TIMES.                           00480000
** THE RECURRING FIELDS ARE INDIVIDUALLY LABELED SO THAT THEY  DEFI0290 00490000
** WILL SHOW UP IN THE EXIT TRACE.                            DEFI0290 00491000
**                                                                      00492000
PAPPAPID     DS  CL8                   APPROVER ID                      00500000
PAPPAPDA     DS  CL7                   APPROVAL/DENIAL DATE             00510000
PAPPAPTI     DS  CL5                   APPROVAL/DENIAL TIME             00511000
PAPPAFLG     DS  CL8                   INDIVIDUAL APPROVER FLAG         00520000
PAPPAREQ     DS  CL8                   APPROVER'S APPROVAL REQUIRED     00560000
PAPPAPLN     EQU *-PAPPAPID           LNG OF INDIVIDUAL APPR SEGMENT    00590000
*                                                                       00590201
PAPPID02     DS  CL8                   APPROVER ID                      00591001
PAPPDT02     DS  CL7                   APPROVAL/DENIAL DATE             00592001
PAPPTI02     DS  CL5                   APPROVAL/DENIAL TIME             00593001
PAPPFL02     DS  CL8                   INDIVIDUAL APPROVER FLAG         00594001
PAPPRQ02     DS  CL8                   APPROVER'S APPROVAL REQUIRED     00595001
*                                                                       00596001
PAPPID03     DS  CL8                   APPROVER ID                      00597001
PAPPDT03     DS  CL7                   APPROVAL/DENIAL DATE             00598001
PAPPTI03     DS  CL5                   APPROVAL/DENIAL TIME             00599001
PAPPFL03     DS  CL8                   INDIVIDUAL APPROVER FLAG         00600001
PAPPRQ03     DS  CL8                   APPROVER'S APPROVAL REQUIRED     00601001
*                                                                       00602001
PAPPID04     DS  CL8                   APPROVER ID                      00603001
```

```
PAPPDT04     DS  CL7                  APPROVAL/DENIAL DATE            00604001
PAPPTI04     DS  CL5                  APPROVAL/DENIAL TIME            00605001
PAPPFL04     DS  CL8                  INDIVIDUAL APPROVER FLAG        00606001
PAPPRQ04     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00607001
*                                                                    00608001
PAPPID05     DS  CL8                  APPROVER ID                     00609001
PAPPDT05     DS  CL7                  APPROVAL/DENIAL DATE            00609101
PAPPTI05     DS  CL5                  APPROVAL/DENIAL TIME            00609201
PAPPFL05     DS  CL8                  INDIVIDUAL APPROVER FLAG        00609301
PAPPRQ05     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00609401
*                                                                    00609501
PAPPID06     DS  CL8                  APPROVER ID                     00609601
PAPPDT06     DS  CL7                  APPROVAL/DENIAL DATE            00609701
PAPPTI06     DS  CL5                  APPROVAL/DENIAL TIME            00609801
PAPPFL06     DS  CL8                  INDIVIDUAL APPROVER FLAG        00609901
PAPPRQ06     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00610001
*                                                                    00610101
PAPPID07     DS  CL8                  APPROVER ID                     00610201
PAPPDT07     DS  CL7                  APPROVAL/DENIAL DATE            00610301
PAPPTI07     DS  CL5                  APPROVAL/DENIAL TIME            00610401
PAPPFL07     DS  CL8                  INDIVIDUAL APPROVER FLAG        00610501
PAPPRQ07     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00610601
*                                                                    00610701
PAPPID08     DS  CL8                  APPROVER ID                     00610801
PAPPDT08     DS  CL7                  APPROVAL/DENIAL DATE            00610901
PAPPTI08     DS  CL5                  APPROVAL/DENIAL TIME            00611001
PAPPFL08     DS  CL8                  INDIVIDUAL APPROVER FLAG        00611101
PAPPRQ08     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00611201
*                                                                    00611301
PAPPID09     DS  CL8                  APPROVER ID                     00611401
PAPPDT09     DS  CL7                  APPROVAL/DENIAL DATE            00611501
```

```
PAPPTI09     DS  CL5                  APPROVAL/DENIAL TIME            00611601
PAPPFL09     DS  CL8                  INDIVIDUAL APPROVER FLAG        00611701
PAPPRQ09     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00611801
*                                                                    00611901
PAPPID10     DS  CL8                  APPROVER ID                     00612001
PAPPDT10     DS  CL7                  APPROVAL/DENIAL DATE            00612101
PAPPTI10     DS  CL5                  APPROVAL/DENIAL TIME            00612201
PAPPFL10     DS  CL8                  INDIVIDUAL APPROVER FLAG        00612301
PAPPRQ10     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00612401
*                                                                    00612501
PAPPID11     DS  CL8                  APPROVER ID                     00612601
PAPPDT11     DS  CL7                  APPROVAL/DENIAL DATE            00612701
PAPPTI11     DS  CL5                  APPROVAL/DENIAL TIME            00612801
PAPPFL11     DS  CL8                  INDIVIDUAL APPROVER FLAG        00612901
PAPPRQ11     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00613001
*                                                                    00613101
PAPPID12     DS  CL8                  APPROVER ID                     00613201
PAPPDT12     DS  CL7                  APPROVAL/DENIAL DATE            00613301
PAPPTI12     DS  CL5                  APPROVAL/DENIAL TIME            00613401
PAPPFL12     DS  CL8                  INDIVIDUAL APPROVER FLAG        00613501
PAPPRQ12     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00613601
*                                                                    00613701
PAPPID13     DS  CL8                  APPROVER ID                     00613801
PAPPDT13     DS  CL7                  APPROVAL/DENIAL DATE            00613901
PAPPTI13     DS  CL5                  APPROVAL/DENIAL TIME            00614001
PAPPFL13     DS  CL8                  INDIVIDUAL APPROVER FLAG        00614101
PAPPRQ13     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00614201
*                                                                    00614301
PAPPID14     DS  CL8                  APPROVER ID                     00614401
PAPPDT14     DS  CL7                  APPROVAL/DENIAL DATE            00614501
PAPPTI14     DS  CL5                  APPROVAL/DENIAL TIME            00614601
```

```
PAPPFL14     DS  CL8                  INDIVIDUAL APPROVER FLAG        00614701
PAPPRQ14     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00614801
*                                                                    00614901
PAPPID15     DS  CL8                  APPROVER ID                     00615001
PAPPDT15     DS  CL7                  APPROVAL/DENIAL DATE            00615101
PAPPTI15     DS  CL5                  APPROVAL/DENIAL TIME            00615201
PAPPFL15     DS  CL8                  INDIVIDUAL APPROVER FLAG        00615301
PAPPRQ15     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00615401
*                                                                    00615501
PAPPID16     DS  CL8                  APPROVER ID                     00615601
PAPPDT16     DS  CL7                  APPROVAL/DENIAL DATE            00615701
PAPPTI16     DS  CL5                  APPROVAL/DENIAL TIME            00615801
PAPPFL16     DS  CL8                  INDIVIDUAL APPROVER FLAG        00615901
PAPPRQ16     DS  CL8                  APPROVER'S APPROVAL REQUIRED    00616001
*                                                                    00616101
             DS  0D                                                  00616200
PAPPAP$L     EQU *-PAPPAPID            LENGTH OF APPROVER LIST        00617000
PAPPSIZE     EQU *-$PAPPREQ            PHYSICAL RECORD LENGTH         00620000
********************************************************************* 00640000
**            E N D   O F   $ P A P P T B L                       ** 00650000
********************************************************************* 00660000
         MEND                                                        00670000
```

6.8.1.2 Field Descriptions

Brief descriptions of the fields in the $PAPPREQ DSECT follow.

Field	Description
PAPPLEN	Size of the Exit control block ($PAPPREQ), in bytes. Not modifiable by any exits.
PAPPVERS	Version number to identify the Exit control block ($PAPPREQ) written by the exit handler. This should always be 1. Use the PAPP$VER label to verify that the correct version of this DSECT was passed to your exit. Not modifiable by any exits.

Field	Description
PAPPID	Code that identifies this as the Approver request block, for use in rapid identification when you are scanning your modules visually. Value is always PAPP. Not modifiable by any exit point.
PAPPSEQ#	Sequential number set by Endevor to tell the user exit the number of records it has requested. Output only. Not modifiable.
PAPPGPNM	Approver group name (RI).
PAPPEVNM	Environment associated with approver group (RI).
PAPPQUOR	Required approval quorum count (RI).
PAPPFLAG	Overall package approver flag.
PAPPAPP#	Number of approvers in the group (RI).
Fields that occur for each approver in group:	
PAPPAPID	Approver ID (RI).
PAPPAPDA	Approval/denial date. Format *ddmmmyy*.
PAPPAPTI	Approval/denial time. Format *hh:mm*.
PAPPAFLG	Individual approver's flag.
PAPPAREQ	Approver's approval required flag (RI).
PAPPAPLN	Length of individual approver segment.
PAPPAPID02-PAPPAPID16	Second through sixteenth occurrence of PAPPAPID.
PAPPDT02-PAPPDT16	Second through sixteenth occurrence of PAPPAPDA.
PAPPTI02-PAPPTI16	Second through sixteenth occurrence of PAPPAPTI.
PAPPFL02-PAPPFL16	Second through sixteenth occurrence of PAPPAFLG.
PAPPRQ02-PAPPRQ16	Second through sixteenth occurrence of PAPPAREQ.
PAPPAP$L	Length of approver list data. Not modifiable.
PAPPSIZE	Size of $PAPPREQ DSECT.

6.8.1.3 PACKAGE-EXIT-APPROVER-MAP control block layout

```
01  PACKAGE-EXIT-APPROVER-MAP.
    05  PAPP-LENGTH             PIC 9(4)  COMP SYNC.
    05  PAPP-CURRENT-VERSION    PIC 9(4)  COMP SYNC.
    05  PAPP-BLOCK-ID           PIC X(4).
    05  PAPP-SEQUENCE-NUMBER    PIC S9(4) COMP SYNC.
    05  PAPP-GROUP-NAME         PIC X(16).
    05  PAPP-ENVIRONMENT        PIC X(8).
    05  PAPP-QUORUM-COUNT       PIC S9(4) COMP SYNC.
```

```
05  PAPP-APPROVER-FLAG      PIC X(8).
05  PAPP-APPROVER-NUMBER    PIC S9(4) COMP SYNC.
05  PAPP-APPROVAL-DATA OCCURS 0 TO 16 TIMES
               DEPENDING ON PAPP-APPROVER-NUMBER.
    10  PAPP-APPROVAL-ID    PIC X(8).
    10  PAPP-APPR-DEN-DATE  PIC X(7).
    10  PAPP-APPR-DEN-TIME  PIC X(5).
    10  PAPP-APPROVAL-FLAG  PIC X(8).
    10  PAPP-APPR-REQUIRED  PIC X(8).
```

6.8.1.4 Field Descriptions

Brief descriptions of the fields in the PACKAGE-EXIT-APPROVER-MAP control block layout follow.

Field	Description
PAPP-LENGTH	Size of the Exit control block (PACKAGE-EXIT-APPROVER-MAP), in bytes. Not modifiable by any exits.
PAPP-CURRENT-VERSION	Version number to identify the Exit control block (PACKAGE-EXIT-APPROVER-MAP) written by the exit handler. This should always be 1. Not modifiable by any exits.
PAPP-BLOCK-ID	Code that identifies this as the Approver request block, for use in rapid identification when you are scanning your modules visually. Value is always PAPP. Not modifiable by any exit point.
PAPP-SEQUENCE-NUMBER	Sequential number set by Endevor to tell the user exit the number of records it has requested. Output only. Not modifiable.
PAPP-GROUP-NAME	Approver group name (RI).
PAPP-ENVIRONMENT	Environment associated with approver group (RI).
PAPP-QUORUM-COUNT	Required approval quorum count (RI).
PAPP-APPROVER-FLAG	Overall package approver flag.
PAPP-APPROVAL-DATA	Number of approvers in the group (RI).
Fields that occur for each approver in group:	
PAPP-APPROVAL-ID	Approver ID (RI).
PAPP-APP-DEN-DATE	Approval/denial date. Format *ddmmmyy*.
PAPP-APP-DEN-TIME	Approval/denial time. Format *hh:mm*.
PAPP-APPROVAL-FLAG	Individual approver's flag.
PAPP-APPR-REQUIRED	Approver's approval required flag (RI).

6.9 Package Backout Request Block

6.9.1 Seventh Information Block

This is the seventh information block passed to the exit. This block contains backout data set information. This block is only built on request from the user exit. None of the fields are modifiable by exit programs.

6.9.1.1 $PBODREQ DSECT

```
$PBODREQ DSECT
*          ***********************************************
*          *  BLOCK INITIALIZATION DATA                  *
*          ***********************************************
PBODLEN    DC    AL2(PBODSIZE)         LENGTH OF BOD TABLE ENTRY
PBODVERS   DC    AL2(PBOD$VER)        VERSION OF BOD TABLE BLK
PBOD$VER   EQU   1                      RELEASE 1
PBODID     DC    CL4'PBOD'             ID OF CONTROL BLOCK
*          ***********************************************
*          *  BLOCK TOTAL DATA                           *
*          ***********************************************
PBODSEQ#     DS  H                     SEQUENTIAL # OF THIS ENTRY
*          ***********************************************
*          *  BACKOUT DATA FIELDS                        *
*          ***********************************************
PBODFLG1     DS  H                     BACKOUT FLAG
PBOD$SRC     EQU 1                     B.O. REC ASSOC W/ACTION SOURCE
PBOD$TRG     EQU 2                     B.O. REC ASSOC W/ACTION TARGET
PBODFLG2     DS  H                     BACKOUT FLAG
PBOD$BKO     EQU 1                     MEMBER HAS BEEN BACKED-OUT
PBOD$BKI     EQU 2                     MEMBER HAS BEEN BACKED-IN
*
PBODDSN      DS  CL44                  DATA SET NAME
PBODCMBR     DS  CL8                   CURRENT MEMBER
PBODNMBR     DS  CL8                   NEW MEMBER  (BLANK IF DELETED)
PBODSMBR     DS  CL8                   SAVE MEMBER (BLANK IF NEW MBR)
*
         SPACE ,
*
PBODSIZE  EQU    *-$PBODREQ            LNG OF BOD TABLE ENTRY
```

6.9.1.2 Field Descriptions

Brief descriptions of the fields in the $PBODREQ DSECT follow.

Field	Description
PBODLEN	Size of the Exit control block ($PBODREQ), in bytes. Not modifiable by any exits.

Field	Description
PBODVERS	Version number to identify the Exit control block ($PBODREQ) written by the exit handler. This should always be 1. Use the PBOD$VER label to verify that the correct version of this DSECT was passed to your exit. Not modifiable by any exits.
PBODID	Code that identifies this as the Backout request block, for use in rapid identification when you are scanning your modules visually. Value is always PBOD. Not modifiable by any exit point.
PBODSEQ#	Sequential number set by Endevor to tell the user exit the number of records it has requested.
PBODFLG1	Backout source flag. Values are: ■ 1 -- PBOD$SRC. Backout record is associated with action source. ■ 2 -- PBOD$TRG. Backout record associated with action target.
PBODFLG2	Backout history flag. Values are: ■ 1 -- PBOD$BKO. Member has been backed out. ■ 2 -- PBOD$BKI. Member has been backed in.
PBODDSN	Backout data set name.
PBODCMBR	Current member.
PBODNMBR	New member (blank if deleted).
PBODSMBR	Save member (blank if new member)
PBODSIZE	Size of $PBODREQ DSECT.

6.9.1.3 PACKAGE-EXIT-BACKOUT-BLOCK control block layout

```
01  PACKAGE-EXIT-BACKOUT-BLOCK.
    05  PBOR-LENGTH             PIC 9(4)  COMP SYNC.
    05  PBOR-CURRENT-VERSION    PIC 9(4)  COMP SYNC.
    05  PBOR-BLOCK-ID           PIC X(4).
    05  PBOR-SEQUENCE-NUMBER    PIC S9(4) COMP.
    05  PBOR-BACKOUT-FLAG1      PIC  9(4) COMP.

        88  PBOR-BO-REC-ASSOC-WITH-SOURCE VALUE 1.
        88  PBOR-BO-REC-ASSOC-WITH-TARGET VALUE 2.

    05  PBOR-BACKOUT-FLAG2      PIC  9(4) COMP.

        88  PBOR-MEMBER-BEEN-BACKED-OUT   VALUE 1.
        88  PBOR-MEMBER-BEEN-BACKED-IN    VALUE 2.
```

```
05  PBOR-DATASET-NAME       PIC X(44).
05  PBOR-CURRENT-MEMBER     PIC X(8).
05  PBOR-NEW-MEMBER         PIC X(8).
05  PBOR-SAVE-MEMBER        PIC X(8).
```

6.9.1.4 Field Descriptions

Brief descriptions of the fields in the PACKAGE-EXIT-BACKOUT-BLOCK control block layout follow.

Field	Description
PBOR-LENGTH	Size of the Exit control block (PACKAGE-EXIT-BACKOUT-BLOCK), in bytes. Not modifiable by any exits.
PBOR-CURRENT-VERSION	Version number to identify the Exit control block (PACKAGE-EXIT-BACKOUT-BLOCK) written by the exit handler. This should always be 1. Not modifiable by any exits.
PBOR-BLOCK-ID	Code that identifies this as the Backout request block, for use in rapid identification when you are scanning your modules visually. Value is always PBOD. Not modifiable by any exit point.
PBOR-SEQUENCE-NUMBER	Sequential number set by Endevor to tell the user exit the number of records it has requested.
PBOR-BACKOUT-FLAG1	Backout source flag. Values are: ▪ Equates: PBOR-BO-REC-ASSOC-WITH-SOURCE – Value: 1 – Meaning: Backout record is associated with action source. ▪ Equates: PBOR-BO-REC-ASSOC-WITH-TARGET – Value: 2 – Meaning: Backout record associated with action target.
PBOR-BACKOUT-FLAG2	Backout history flag. Values are: ▪ Equates: PBOR-MEMBER-BEEN-BACKED-OUT – Value: 1 – Meaning: Member has been backed out. ▪ Equates: PBOR-MEMBER-BEEN-BACKED-IN – Value: 2 – Meaning: Member has been backed in.
PBOR-DATASET-NAME	Backout data set name.
PBOR-CURRENT-MEMBER	Current member.

Field	Description
PBOR-NEW-MEMBER	New member (blank if deleted).
PBOR-SAVE-MEMBER	Save member (blank if new member)

6.10 Package Ship Request Block

6.10.1 Eighth Information Block

This is the eighth information block passed to the exit. This block contains backout data set information. This block is only built on request from the user exit. None of the fields are modifiable by exit programs.

6.10.1.1 $PSHPREQ DSECT

```
$PSHPREQ DSECT
*          ***********************************************
*          *  BLOCK INITIALIZATION DATA                  *
*          ***********************************************
PSHPLEN    DC    AL2(PSHPSIZE)         LENGTH OF BOD TABLE ENTRY
PSHPVERS   DC    AL2(PSHP$VER)        VERSION OF BOD TABLE BLK
PSHP$VER   EQU   1                      RELEASE 1
PSHPID     DC    CL4'PSHP'             ID OF CONTROL BLOCK
*          ***********************************************
*          *  BLOCK TOTAL DATA                           *
*          ***********************************************
PSHPSEQ#     DS  H                  SEQUENTIAL # OF THIS ENTRY
*          ***********************************************
*          *  SHIPMENT DATA FIELDS                       *
*          ***********************************************
PSHPPKID    DS  CL16               ID OF PACKAGE SHIPPED
PSHPDSUB    DS  CL7                  DATE SUBMITTED (DDMONYY)
PSHPTSUB    DS  CL5                  TIME SUBMITTED (HH:MM)
PSHP#DST    DS  H                  # OF DESTINATIONS IN RECORD
***
***  THE FOLLOWING FIELDS OCCUR PSHP#DST TIMES
***
PSHPDEST     DS  CL8                   SHIPMENT DESTINATION
PSHPTYPE     DS  CL3                   SHIPMENT TYPE
PSHP$PKG     EQU C'P'                  PKG = PACKAGE SHIPPED
PSHP$BKT     EQU C'B'                  BKO = BACKOUT SHIPPED
PSHPCOMP     DS  CL3                   COMPLEMENTARY FILES INDICATOR
PSHPC$YS     EQU C'Y'                  YES = COMPLEMENT FILES REQ'ED
PSHPC$NO     EQU C'N'                  NO  = NO COMPLEMENTARY FILES
*
*-----------------------------------------------------------------------*
*  COMPLETION CODES:                                                    *
*      FUNCTION COMPLETED:           RC=##  - RC=###  - R=####         *
*      CANCELLED BY EXIT:            XC=##  - XC=###  - X=####         *
*      ABEND DETECTED:               S/###  - U####                    *
*      GENERIC COMPLETION:           EXEC'D - ABEND                    *
*-----------------------------------------------------------------------*

PSHPC$RC    EQU C'R'                  CC IS A RETURN CODE
PSHPC$XX    EQU C'X'                  CC IS AN EXIT CANCELLATION
PSHPC$SA    EQU C'S'                  CC IS A SYSTEM ABEND
PSHPC$UA    EQU C'U'                  CC IS A USER ABEND
PSHPC$EX    EQU C'E'                  FUNCTION EXEC'D (RC UNAVAILABLE)
PSHPC$AB    EQU C'A'                  FUNCTION ABND'D (RC UNAVAILABLE)
PSHPCCST    DS  CL6                  SHIPMENT STAGING COMPLETION CODE
```

```
PSHPCCTR    DS  CL6                 SHIPMENT TRANSMISSION COMPL CODE
PSHPCCRM    DS  CL6                 REMOTE COPY/DELETE COMPL'TN CODE
            DS  3CL6                ** RESERVED **
PSHPD$LN    EQU *-PSHPDEST          LENGTH OF A SINGLE DEST ENTRY
            DS  9CL(PSHPD$LN)       UP TO NINE MORE ENTRIES IN LIST
PSHP$TLN    EQU *-PSHPDEST          TOTAL LENGTH OF DEST INFO LIST
*
         SPACE ,
*
PSHPSIZE  EQU    *-$PSHPREQ           LNG OF BOD TABLE ENTRY
```

6.10.1.2 Field Descriptions

Brief descriptions of the fields in the $PSHPREQ DSECT follow.

Field	Description
PSHPLEN	Size of the Exit control block ($PSHPREQ), in bytes. Not modifiable by any exits.
PSHPVERS	Version number to identify the Exit control block ($PSHPREQ) written by the exit handler. This should always be 1. Use the PSHP$VER label to verify that the correct version of this DSECT was passed to your exit. Not modifiable by any exits.
PSHPID	Code that identifies this as the Ship request block, for use in rapid identification when you are scanning your modules visually. Value is always PSHP. Not modifiable by any exits.
PSHPSEQ#	Sequential number set by Endevor to tell the user exit the number of records it has requested.
PSHPPKID	Package ID associated with the ship record.
PSHPDSUB	Date SHIP request was submitted. Format *ddmmmyy*.
PSHPTSUB	Time SHIP request was submitted. Format *hh:mm*.
PSHPDEST	Shipment destination.
PSHPTYPE	Shipment type. Label values are: ■ P -- PSHP$PKG; Package shipped ■ B -- PSHP$BKT; Backout shipped
PSHPCOMP	Complementary file indicator. Label values are: ■ Y -- PSHPC$YS; Ship complementary files ■ N -- PSHPC$NO; Do not ship complementary files.

Field	Description
PSHPCCST	Completion code for shipment staging. The label that follows the field identifies the value in the field. Label values are: ■ R -- PSHPC$RC; the value is a return code ■ X -- PSHPC$XX; value is an exit cancellation. ■ S -- PSHPC$SA; value is a system abend. ■ U -- PSHPC$UA; value is a user abend. ■ E -- PSHPC$EX; action was executed but return code is unavailable. ■ A -- PSHPC$AB; action abended but return code is unavailable.
PSHPCCTR	Completion code for shipment transmission. Labels and acceptable values are the same as for PSHPCCST.
PSHPCCRM	Completion code for remote copy delete. Labels and acceptable values are the same as for PSHPCCST.
PSHPSIZE	Size of $PSHPREQ DSECT.

6.10.1.3 PACKAGE-EXIT-SHIPMENT-BLOCK control block layout

```
*****************************************************************
*    PACKAGE-EXIT-SHIPMENT-BLOCK.                               *
*****************************************************************

 01  PACKAGE-EXIT-SHIPMENT-BLOCK.
     05  PSHP-LENGTH             PIC 9(4)  COMP SYNC.
     05  PSHP-CURRENT-VERSION    PIC 9(4)  COMP SYNC.
     05  PSHP-BLOCK-ID           PIC X(4).
     05  PSHP-SEQUENCE-NUMBER    PIC S9(4) COMP.
     05  PSHP-PACKAGE-ID         PIC X(16).
     05  PSHP-DATE-SUBMITTED     PIC X(7).
     05  PSHP-TIME-SUBMITTED     PIC X(5).
     05  PSHP-DESTINATION-NUMBER PIC S9(4) COMP.
     05  PSHP-DESTINATION-DATA   OCCURS 0 TO 10 TIMES
                    DEPENDING ON PSHP-DESTINATION-NUMBER.
         10  PSHP-SHIP-DESTINATION   PIC X(8).
         10  PSHP-SHIPMENT-TYPE.
             20  FILLER              PIC X(1).
             88  PSHP-PACKAGE-SHIPPED     VALUE 'P'.
             88  PSHP-BACKOUT-SHIPPED     VALUE 'B'.
             20  FILLER              PIC X(2).
         10  PSHP-COMPLEMENT-FILES-FLAG.
             20  FILLER              PIC X(1).
             88  PSHP-COMPLEMENT-REQUIRED    VALUE 'Y'.
             88  PSHP-NO-COMPLEMENT-REQUIRED VALUE 'N'.
             20  FILLER              PIC X(2).

*
*      COMPLETION CODES:
```

```
*
*      FUNCTION COMPLETED:  RC=##  - RC=###  - R=####
*      CANCELLED BY EXIT:   XC=##  - XC=###  = X=####
*      GENERIC COMPLETION:  EXEC'D - ABEND
*

         10  PSHP-STAGING-COMPL-CODE  PIC X(6).
         10  PSHP-TRANSMI-COMPL-CODE  PIC X(6).
         10  PSHP-REMOTE-CPY-DEL-CODE PIC X(6).

         88  PSHP-CC-IS-A-RETURN-CODE     VALUE 'R'.
         88  PSHP-CC-IS-AN-EXIT-CANCEL    VALUE 'X'.
         88  PSHP-CC-IS-A-SYSTEM-ABEND    VALUE 'S'.
         88  PSHP-CC-IS-A-USER-ABEND      VALUE 'U'.
         88  PSHP-FUNCTION-EXECUTED       VALUE 'E'.
         88  PSHP-FUNCTION-ABENDED        VALUE 'A'.

         10  FILLER                   PIC X(18).

     EJECT
```

6.10.1.4 Field Descriptions

Brief descriptions of the fields in the PACKAGE-EXIT-SHIPMENT-BLOCK control block layout follow.

Field	Description
PSHP-LENGTH	Size of the Exit control block (PACKAGE-EXIT-SHIPMENT-BLOCK), in bytes. Not modifiable by any exits.
PSHP-CURRENT-VERSION	Version number to identify the Exit control block (PACKAGE-EXIT-SHIPMENT-BLOCK) written by the exit handler. This should always be 1. Not modifiable by any exits.
PSHP-BLOCK-ID	Code that identifies this as the Ship request block, for use in rapid identification when you are scanning your modules visually. Value is always PSHP. Not modifiable by any exits.
PSHP-SEQUENCE-NUMBER	Sequential number set by Endevor to tell the user exit the number of records it has requested.
PSHP-PACKAGE-ID	Package ID associated with the ship record.
PSHP-DATE-SUBMITTED	Date SHIP request was submitted. Format *ddmmmyy*.
PSHP-TIME-SUBMITTED	Time SHIP request was submitted. Format *hh:mm*.
PSHP-SHIP-DESTINATION	Shipment destination.
PSHP-SHIPMENT-TYPE	Shipment type. Label values are: ■ P -- PSHP-PACKAGE-SHIPPED: Package shipped ■ B -- PSHP-BACKOUT-SHIPPED: Backout shipped

Field	Description
PSHP-COMPLEMENT-FILES-FLAG	Complementary file indicator. Label values are: ■ Y -- PSHP-COMPLEMENT-REQUIRED: Ship complementary files ■ N -- PSHP-NO-COMPLEMENT-REQUIRED: Do not ship complementary files.
PSHP-STAGING-COMPL-CODE	Completion code for shipment staging. The label that follows the field identifies the value in the field. Label values are: ■ R -- PSHP-CC-IS-A-RETURN-CODE: the value is a return code ■ X -- PSHP-CC-IS-AN-EXIT-CANCEL: value is an exit cancellation. ■ S -- PSHP-CC-IS-A-SYSTEM-ABEND: value is a system abend. ■ U -- PSHP-CC-IS-A-USER-ABEND: value is a user abend. ■ E -- PSHP-FUNCTION-EXECUTED: action was executed but return code is unavailable. ■ A -- PSHP-FUNCTION-ABENDED: action abended but return code is unavailable.
PSHP-TRANSMI-COMPL-CODE	Completion code for shipment transmission. Labels and acceptable values are the same as for PSHP-STAGING-COMPL-CODE.
PSHP-REMOTE-CPY-DEL-CODE	Completion code for remote copy delete. Labels and acceptable values are the same as for PSHP-STAGING-COMPL-CODE.

6.11 Package SCL Request Block

6.11.1 Ninth Information Block

This is the ninth information block passed to the exit. This block contains package SCL information. This block is only built on request from the user exit. None of the fields are modifiable by exit programs.

6.11.1.1 $PSCLREQ DSECT

```
*           ************************************
*           **     PACKAGE EXIT SCL RECORD    **
*           ************************************
$PSCLREQ DSECT
*           **************************************
*           **     BLOCK INITIALIZATION FIELDS **
*           **************************************
PSCLLEN     DC  AL2(PSCLSIZE)
PSCLVERS    DC  AL2(PSCL$VER)
PSCL$VER   EQU  1
PSCLID      DC  CL4'PSCL'
*           ************************************
*           **     BLOCK TOTAL DATA           **
*           ************************************
PSCLSEQ#    DS  H                      SEQUENTIAL NUMBER OF THIS ENTRY
*           ************************************
*           **     SCL RECORDS                **
*           ************************************
         SPACE 1
PSCL#SMT     DS  H                      NBR OF SCL STATEMENTS IN RECORD
**
** THE FOLLOWING FIELDS OCCUR PSCL#SMT TIMES
**
PSCLSTMT     DS  CL80                   SCL STATEMENT
PSCLSLNG     EQU *-PSCLSTMT             LNG OF INDIVIDUAL SCL SEG
             DS  9XL(PSCLSLNG)          ROOM FOR NINE MORE STATEMENTS
             DS  0D
PSCLSC$L     EQU *-PSCLSTMT              LENGTH OF SCL STATEMENT LIST
PSCLSIZE     EQU *-$PSCLREQ              PHYSICAL RECORD LENGTH
*********************************************************************
**             E N D   O F   $ P A P P T B L                       **
*********************************************************************
```

6.11.1.2 Field Descriptions

The following list describes the fields in the $PSCLREQ DSECT.

Field	Description
PSCLLEN	Size of the Exit control block ($PSCLREQ), in bytes. Not modifiable by any exits.

Field	Description
PSCLVERS	Version number to identify the Exit control block ($PSCLREQ) written by the exit handler. This should always be 1. Use the PSCLVER label to verify that the correct version of this DSECT was passed to your exit. Not modifiable by any exits.
PSCLID	Code that identifies this as the SCL request block, for use in rapid identification when you are scanning your modules visually. Value is always PSCL. Not modifiable by any exits.
PSCLSEQ#	Sequential number set by Endevor to tell the user exit the number of records it has requested.
PSCL#STM	Number of SCL statements in request block. Maximum is 10 statements.
The following four fields occur once for each SCL statement:	
PSCLSTMT	Actual SCL statement. The block is large enough to hold 10 statements.
PSCLSLNG	Length of the SCL statement.
PSCLSC$L	Length of SCL statement list.
PSCLSIZE	Size of the $PSCLREQ DSECT.

6.11.1.3 PACKAGE-EXIT-SCL-RECORD control block layout

```
*****************************************************************
*    PACKAGE-EXIT-SCL-BLOCK.                                    *
*****************************************************************

 01  PACKAGE-EXIT-SCL-BLOCK.
     05  PSCL-LENGTH            PIC 9(4)  COMP SYNC.
     05  PSCL-CURRENT-VERSION   PIC 9(4)  COMP SYNC.
     05  PSCL-BLOCK-ID          PIC X(4).
     05  PSCL-SEQUENCE-NUMBER   PIC S9(4) COMP.
     05  PSCL-SCL-STATEMENT-NBR PIC S9(4) COMP.
     05  PSCL-SCL-DATA          OCCURS 0 TO 10 TIMES
                   DEPENDING ON PSCL-SCL-STATEMENT-NBR.
         10  PSCL-STATEMENT     PIC X(80).
     EJECT
```

6.11.1.4 Field Descriptions

The following list describes the fields in the PACKAGE-EXIT-SCL-BLOCK control block layout:

Field	Description
PSCL-LENGTH	Size of the Exit control block (PACKAGE-EXIT-SCL-BLOCK), in bytes. Not modifiable by any exits.
PSCL-CURRENT-VERSION	Version number to identify the Exit control block (PACKAGE-EXIT-SCL-BLOCK) written by the exit handler. This should always be 1. Not modifiable by any exits.
PSCL-BLOCK-ID	Code that identifies this as the SCL request block, for use in rapid identification when you are scanning your modules visually. Value is always PSCL. Not modifiable by any exits.
PSCL-SEQUENCE-NUMBER	Sequential number set by Endevor to tell the user exit the number of records it has requested.
PSCL#STM	Number of SCL statements in request block. Maximum is 10 statements.
The following field occurs once for each SCL statement:	
PSCL-STATEMENT	Actual SCL statement. The block is large enough to hold 10 statements.

6.11.2 Tenth Information Block

This is the tenth information block passed to the exit. This block contains DistribuLink product collection information. This block is only built on request form the user exit. None of the fields are modifiable by exit programs.

6.11.2.1 $PCOLREQ DSECT—Product Collection Request Block

```
$PCOLREQ DSECT
*          ***********************************************
*          *  BLOCK INITIALIZATION DATA                  *
*          ***********************************************
PCOLLEN    DC    AL2(PCOLSIZE)         LENGTH OF BOD TABLE ENTRY
PCOLVERS   DC    AL2(PCOL$VER)         VERSION OF BOD TABLE BLK
PCOL$VER   EQU   1                     RELEASE 1
PCOLID     DC    CL4'PCOL'             ID OF CONTROL BLOCK
*          ***********************************************
*          *  BLOCK TOTAL DATA                           *
*          ***********************************************
PCOLSEQ#     DS  H                  SEQUENTIAL # OF THIS ENTRY
*-------------------------------------------------------------------*
PCOLPKID    DS  CL16                ID OF PACKAGE COLLECTED
```

```
PCOLTYPE    DS  CL3                 TYPE OF PACKAGE COLLECTED
PCOL$PKG    EQU C'P'                  PKG = PACKAGE COLLECTED
PCOL$BKT    EQU C'B'                  BKO = BACKOUT COLLECTED
*-------------------------------------------------------------------*
*  COMPLETION CODES:                                                *
*      FUNCTION COMPLETED:           RC=##  - RC=###  - R=####      *
*      CANCELLED BY EXIT:            XC=##  - XC=###  - X=####      *
*      ABEND DETECTED:               S/###  - U####                 *
*-------------------------------------------------------------------*
PCOLCCST    DS  CL6                   COLLECTION STAGING COMPL CODE
PCOLCCCR    DS  CL6                   COLLECTION REQUEST COMPL CODE
PCOLC$RC    EQU C'R'                  CC IS A RETURN CODE
PCOLC$XX    EQU C'X'                  CC IS AN EXIT CANCELLATION
PCOLC$SA    EQU C'S'                  CC IS A SYSTEM ABEND
PCOLC$UA    EQU C'U'                  CC IS A USER ABEND
*-------------------------------------------------------------------*
```

```
PCOLDSUB    DS  CL7                   DATE SUBMITTED (DDMONYY)
PCOLTSUB    DS  CL5                   TIME SUBMITTED (HH:MM)
PCOLCONN    DS  CL8                   CONNECTION ID
PCOLSTARTD  DS  CL8                   DISTRIBULINK START COLLECT DATE
PCOLSTARTT  DS  CL8                   DISTRIBULINK START COLLECT TIME
PCOLDLNAME  DS  CL20                  DISTRIBULINK PRODUCT NAME
PCOLDLDESC  DS  CL30                  DISTRIBULINK PRODUCT DESCRIPTION
PCOLDLVERS  DS  CL15                  DISTRIBULINK PRODUCT VERSION
PCOLDLNUMB  DS  CL15                  DISTRIBULINK PRODUCT NUMBER
PCOLDLVEND  DS  CL15                  DISTRIBULINK PRODUCT VENDOR
PCOLDLOPSY  DS  CL15                  DISTRIBULINK PRODUCT OP SYS
PCOLDLESCP  DS  CL1                   DISTRIBULINK PRODUCT ESCAPE CHAR
PCOLDLTYPE  DS  CL1                   DISTRIBULINK PRODUCT TYPE
PCOLDL$EXT  EQU C'E'                  . EXTERNAL
PCOLDL$GEN  EQU C'G'                  . GENERATION
PCOLDL$NOR  EQU C'N'                  . NORMAL
PCOLDLINST  DS  CL1                   DISTRIBULINK "INSTALLED WHEN"
PCOLDL$DELD EQU C'D'                  . DELIVERED
PCOLDL$INST EQU C'I'                  . INSTALLED
*
         SPACE ,
*
PCOLSIZE  EQU    *-$PCOLREQ            LNG OF BOD TABLE ENTRY
         MEND
```

6.11.2.2 PACKAGE-EXIT-COLLECT-BLOCK control block layout

```
*****************************************************************
*    PACKAGE-EXIT-COLLECT-BLOCK.                                *
*****************************************************************

 01  PACKAGE-EXIT-COLLECT-BLOCK.
     05  PCOL-LENGTH             PIC 9(4)  COMP SYNC.
     05  PCOL-CURRENT-VERSION    PIC 9(4)  COMP SYNC.
     05  PCOL-BLOCK-ID           PIC X(4).
     05  PCOL-SEQUENCE-NUMBER    PIC S9(4) COMP.
     05  PCOL-PACKAGE-ID         PIC X(16).
     05  PCOL-TYPE-COLLECTD      PIC X(3).

             88  PCOL-PACKAGE-COLLECTD    VALUE 'P'.
```

```
            88  PCOL-BACKOUT-COLLECTD    VALUE 'B'.

***************************************************************
*  COMPLETION CODES:                                          *
*      FUNCTION COMPLETED:           RC=##  - RC=###  - R=####  *
*      CANCELLED BY EXIT:            XC=##  - XC=###  - X=####  *
*      ABEND DETECTED:               S/###  - U####             *
***************************************************************

     05  PCOL-STAGING-COMP-CODE  PIC X(6).
     05  PCOL-REQUEST-COMP-CODE  PIC X(6).

            88  PCOL-CC-RETURN-CODE      VALUE 'R'.
            88  PCOL-CC-EXIT-CANCELLD    VALUE 'X'.
            88  PCOL-CC-SYSTEM-ABEND     VALUE 'S'.
            88  PCOL-CC-USER-ABEND       VALUE 'U'.

     05  PCOL-DATE-SUBMITTED           PIC X(7).
     05  PCOL-TIME-SUBMITTED           PIC X(5).
     05  PCOL-CONNECT-ID               PIC X(8).
     05  PCOL-START-COLLECT-DATE       PIC X(8).
     05  PCOL-START-COLLECT-TIME       PIC X(8).
     05  PCOL-COLLECT-PRODUCT-NAME     PIC X(8).
     05  PCOL-COLLECT-PRODUCT-DESC     PIC X(20).
     05  PCOL-COLLECT-PRODUCT-VERSION  PIC X(15).
     05  PCOL-COLLECT-PRODUCT-NUMBER   PIC X(15).
     05  PCOL-COLLECT-PRODUCT-VENDOR   PIC X(15).
     05  PCOL-COLLECT-PRODUCT-OPSYS    PIC X(15).
     05  PCOL-COLLECT-PRODUCT-ESC-TYPE PIC X(1).
     05  PCOL-COLLECT-PRODUCT-TYPE     PIC X(1).

            88  PCOL-PROD-TYPE-EXTERNAL  VALUE 'E'.
            88  PCOL-PROD-TYPE-GENERATN  VALUE 'G'.
            88  PCOL-PROD-TYPE-NORMAL    VALUE 'N'.

     05  PCOL-COLLECT-PRODUCT-WHEN     PIC X(1).

            88  PCOL-PROD-DELIVERED      VALUE 'D'.
            88  PCOL-PROD-INSTALLED      VALUE 'I'.
```

6.12 Package Exit Example

6.12.1 Streamlining the Package Approval Process

You can use package exits to streamline the package approval process. Consider an organization that requires three approver groups to sign off on all packages. These approver groups are Programmers (Group G1), QA (Group G2), and Production (Group G3). The organization also requires that Group G1 must approve the package before Group G2, and that Group G2 must approve the package before Group G3.

To implement this approval procedure, the site administrator:

- Writes an after-cast exit that notifies each approver in Group G1 that there is a package to review.

Note: Program BC1PAPNT, in *iprfx.iqual*.CONLIB, is a sample exit program that can be used for after-cast notification of approvers.

- Writes after-review exits to notify:
  - Each approver in Group G2 that there is a package to approve.
  - Each approver in Group G3 that there is a package to approve.
- Writes List exits to build the lists for each approver group.

Chapter 7. Exit Program Examples

This chapter contains the following sections:

- Overview
- Sample Exit 2 & 6 Program and their association with @CCIDTBL
- The Notification Utility

7.1 Overview

This chapter illustrates the use of the Notification Utility and exits 2 and 6. The Notification Utility is used to send a message when an change takes place. The Notitification Utility is discussed in detail in the *Utilities Guide*. Exits 2 & 6 are used to create a CCID table for authorization-checking and then to delete that CCID table when the system is brought down.

7.2 Exits 2 and 6

This example illustrates exits 2 and 6. In this example, exit 2 enforces the use of specific CCIDs for ADD or UPDATE action requests, and exit 6 deletes the CCID table built by the @CCIDTBL macro.

Four components are used by this example, all of which are supplied at installation in the Endevor source library.

Component name	Description
@CCIDTBL	Macro that creates a table of CCIDs that are valid for ADD/UPDATE requests. Each CCID is qualified by environment and system.
CCIDTBL	Table of CCID rules, created by @CCIDTBL.
C1UEXT02	Exit 2 program. The logic here examines each ADD/UPDATE action request to determine the environment, system, and CCID specified for that action, then scans the CCID rule table for a match. If it finds a match based on all three fields--or if it does not find the environment/system combination at all--exit 2 returns to Endevor with a return code of zero (0). If it finds a match on environment and system but not on CCID, exit 2 returns an error (return code 8) to Endevor.
C1UEXT06	Exit 6 (job termination) program. The logic here deletes the CCID table at job termination.

7.2.1 @CCIDTBL

7.2.2 Building the CCID Table for Exit 2

This macro builds the CCID table that will be used by exit 2 to ensure that a user has permission to perform the requested action.

```
         MACRO
         @CCIDTBL &ENV=,&SYSTEM=,&CCID=,&ENTRY=
         GBLB &FIRST
*******************************************************************
*   CHECK IF ENTRY= SPECIFIED                                     *
*******************************************************************
         AIF  ('&ENTRY' EQ 'END').RTN200
         AIF  (K'&ENTRY NE 0).ERR100
*******************************************************************
*   VALIDATE PARAMETERS                                           *
*******************************************************************
         AIF  (K'&ENV EQ 0 OR K'&ENV GT 8).ERR101
         AIF  (K'&SYSTEM EQ 0 OR K'&SYSTEM GT 8).ERR101
         AIF  (K'&CCID EQ 0 OR K'&CCID GT 12).ERR101
*******************************************************************
*   FIRST TIME LOGIC                                              *
*******************************************************************
```

```
         AIF  (&FIRST).RTN100
&FIRST    SETB 1
CCIDTBL   CSECT
*********************************************************************
*   GENERATE CCID TABLE                                             *
*********************************************************************
RTN100   ANOP
         DC   CL4'CCID',CL8'&ENV',CL8'&SYSTEM',CL12'&CCID'
         MEXIT
RTN200   ANOP
         DC   CL4'CCID',XL28'00'       END OF TABLE INDICATOR
         END
         MEXIT
*********************************************************************
*   ERRORS                                                          *
*********************************************************************
ERR100   ANOP
         MNOTE 12,'ENTRY=END MUST BE SPECIFIED AS LAST CARD IN TABLE'
         MEXIT
ERR101   ANOP
         MNOTE 12,'ENV= PARAMETER MUST BE 1 TO 8 CHARACTERS'
         MEXIT
ERR102   ANOP
         MNOTE 12,'SYSTEM= PARAMETER MUST BE 1 TO 8 CHARACTERS'
         MEXIT
ERR103   ANOP
         MNOTE 12,'CCID= PARAMETER MUST BE 1 TO 12 CHARACTERS'
         MEND
```

7.2.2.1 Rules

This table gives the rules for the CCID table. It is created by macro @CCIDTBL.

```
@CCIDTBL ENV=BST,SYSTEM=TESTXXXX,CCID=F.BUCKLEY
@CCIDTBL ENV=BST,SYSTEM=TESTXXXX,CCID=J.GRIFFIN
@CCIDTBL ENV=BST,SYSTEM=TESTXXXX,CCID=P.MCCULLOUGH
@CCIDTBL ENV=BST,SYSTEM=TESTXXXX,CCID=S.VELLOFF
@CCIDTBL ENTRY=END
```

7.2.3 Sample Exit 2 Program

7.2.3.1 Overview

This program checks to see if the CCID associated with an action has been defined. The program loads a table called CCIDTBL (see preceding section), then searches it for a match on environment name, system name, and CCID name. The program cancels the action if it does not find a match. If it does not find an environment/system entry for a CCID, it considers that CCID valid. Likewise, if no CCID has been entered, the program considers the CCID valid. Program C1UEXT06 (termination exit) deletes the table during termination.

```
C1UEXT02 TITLE 'Endevor EXIT 2 EXAMPLE'
***********************************************************************
*   REGISTERS ON ENTRY:                                              *
*               0(R1) --> $ECBDS      EXIT CONTROL BLOCK              *
*               4(R1) --> $ENVDS      ENVIRONMENT BLOCK               *
*   REGISTER USAGE:                                                   *
*               R3     -> STATUS INDICATOR                            *
*               R5     -> CCIDS                                       *
*               R6     -> $ENDVS                                      *
*               R7     -> $ECBDS                                      *
*               R8     -> $REQPDS                                     *
*               R12    -> BASE PROGRAM                                *
*               R13    -> STACK USED FOR STANDARD IBM USAGE           *
***********************************************************************
         TITLE '$ECBDS : EXIT CONTROL BLOCK'
***********************************************************************
*   EXIT CONTROL BLOCK                                                *
***********************************************************************
         $ECBDS
         TITLE '$ENVDS : ENVIRONMENT BLOCK'
***********************************************************************
*   ENVIRONMENT BLOCK                                                 *
***********************************************************************
         $ENVDS
         TITLE '$REQPDS: REQUEST INFORMATION BLOCK'
***********************************************************************
*   REQUEST INFORMATION BLOCK                                         *
***********************************************************************
         $REQPDS
         TITLE 'CCIDDS : CCID TABLE'
***********************************************************************
*   DSECT FOR LOADMODULE CCIDTBL                                      *
***********************************************************************
CCIDS    DSECT
CCIDEYE  DS    CL4                    EYECATCHER 'CCID'
CCIDENV  DS    CL4                    ENVIRONMENT NAME
CCIDSYS  DS    CL4                    SYSTEM NAME
CCIDCCID DS    CL4                    CCID NAME
CCIDSLN  EQU   *-CCIDS                DSECT LENGTH
         TITLE 'REGISTER EQUATES'
***********************************************************************
*   REGISTER EQUATES                                                  *
***********************************************************************
R0       EQU   0
```

```
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         TITLE 'MAIN : MAINLINE LOGIC'
*********************************************************************
*   MAINLINE LOGIC                                                  *
*********************************************************************
C1UEXT02 CSECT
         SAVE  (14,12),,'EXAMPLE OF EXIT 2'
         LR    R12,R15                POINT AT THIS PROGRAM
        USING  C1UEXT02,R12
         L     R7,0(R1)               POINT AT $ECBDS
        USING  $ECBDS,R7
         L     R6,4(R1)               POINT AT $ENVDS #1
        USING  $ENVDS,R6
         L     R8,ECBREQA             POINT AT $REQPDS#1
        USING  $REQPDS,R8
        USING  CCIDS,R5
         XR    R3,R3                  CLEAR CCID-MUST-MATCH INDICATOR
         CLC   REQCCID,BLANKS         Q: WAS CCID SPECIFIED ?
         BE    MAINEXIT               A: NO, EXIT IN SUCCESS
         SPACE
*********************************************************************
*   LOAD THE CCIDTBL, IF NECESSARY                                  *
*********************************************************************
         ICM   R5,B'1111',ECBUEXT     Q: TABLE ALREADY BEEN LOADED?
         BNZ   MAIN0100               A: YES, CHECK FOR CCID MATCH
         LOAD  EP=CCIDTBL,ERRET=MAIN0800
         ST    R0,ECBUEXT             SAVE THE TABLE ADDR IN THE ECB
         LR    R5,R0                    AND IN R5
         SPACE
*********************************************************************
*   VALIDATE CCID ONLY FOR ADD AND UPDATE ACTIONS                   *
*********************************************************************
         CLC   CCIDEYE,=CL4'CCID'     Q: IS TABLE ENTRY VALID?
         BNE   MAIN0810               A: NO, BAD TABLE
         CLC   ECBFUNC,=A(ECB$ADD)    Q: IS THE ACTION "ADD" ?
         BE    MAIN0100               A: YES, CHECK ENVNEXT
         CLC   ECBFUNC,=A(ECB$UPD)    Q: IS THE ACTION "UPDATE" ?
         BE    MAIN0100               A: YES, CHECK ENVNEXT
         B     MAINEXIT                 OTHERWISE EXIT.
         SPACE
*********************************************************************
*   POINT R6 AT THE TARGET $ENVDS BLOCK : ENVNEXT                   *
*********************************************************************
MAIN0100 DS    0H
```

```
         L     R6,ENVNEXT              POINT AT $ENVDS #2
00038000
         SPACE
00039000
*********************************************************************
*   LOOP THROUGH CCID TABLE LOOKING FOR MATCH                       *
*********************************************************************
MAIN0110 DS    0H
         CLC   ENVENVM,CCIDENV         Q: ENVIRONMENT NAME MATCH?
         BNE   MAIN0120                A: NO, CHECK NEXT TABLE ENTRY
         CLC   ENVSYSTM,CCIDSYS        Q: SYSTEM NAME MATCH?
         BNE   MAIN0120                A: NO, CHECK NEXT TABLE ENTRY
         CLC   REQCCID,CCIDCCID        Q: CCID MATCH?
         BE    MAINEXIT                A: YES, CCID IS VALID
         LA    R3,1                    SAY CCID MATCH MUST OCCUR
MAIN0120 DS    0H                        AND READ NEXT
         LA    R5,CCIDSLN(,R5)         POINT TO NEXT ENTRY
         CLC   CCIDEYE,=CL4'CCID'      Q: IS TABLE ENTRY VALID?
         BNE   MAIN0810                A: NO, BAD TABLE
         CLI   CCIDENV,X'00'           Q: LAST TABLE ENTRY
         BNE   MAIN0110                A: NO, CHECK NEXT TABLE ENTRY
         SPACE
*********************************************************************
*   ALL ENTRIES HAVE BEEN REVIEWED; CHECK IF MATCH MUST OCCUR       *
*********************************************************************
         LTR   R3,R3                   Q: MUST MATCH OCCUR ?
         BZ    MAINEXIT                A: NO, ALLOW ACTION TO OCCUR
         B     MAIN0820                DO NOT ALLOW ACTION TO OCCUR
         SPACE
*********************************************************************
*   LOAD FAILURE FOR MODULE CCIDTBL                                 *
*********************************************************************
MAIN0800 DS    0H
         MVC   ECBMSG(L'ERROR1),ERROR1 LOAD FAILURE FOR MODULE CCIDTBL
         MVC   ECBMSGCD,=CL4'0001'     SET LOAD FAILURE INDICATOR
         MVC   ECBRTCD,=A(ECB$FAIL)    TELL NDVR NOT TO PERFORM ACTION
         B     MAINEXIT                  AND RETURN
         SPACE
*********************************************************************
*   MODULE CCIDTBL CONTAINS INVALID ENTRY                           *
*********************************************************************
MAIN0810 DS    0H
         MVC   ECBMSG(L'ERROR2),ERROR2 INVALID TABLE ENTRY
         MVC   ECBMSGCD,=CL4'0002'     SET LOAD FAILURE INDICATOR
         MVC   ECBRTCD,=A(ECB$FAIL)    TELL NDVR NOT TO PERFORM ACTION
         B     MAINEXIT                  AND RETURN
         SPACE
*********************************************************************
*   INVALID CCID REQUEST FOR ACTION - CCID NOT FOUND IN TABLE CCIDTBL *
*********************************************************************
MAIN0820 DS    0H
         MVC   ECBMSG(L'ERROR3),ERROR3 INVALID TABLE ENTRY
         MVC   ECBMSGCD,=CL4'0003'     SET LOAD FAILURE INDICATOR
         MVC   ECBRTCD,=A(ECB$FAIL)    TELL NDVR NOT TO PERFORM ACTION
         B     MAINEXIT                  AND RETURN
         SPACE
*********************************************************************
*   RETURN TO CALLER                                                *
```

```
*********************************************************************
MAINEXIT DS    0H
         XR    R15,R15                 CLEAR R15 FOR NEATNESS
         RETURN (0,12)                   AND RETURN
         SPACE
         TITLE 'PROGRAM CONSTANTS'
*********************************************************************
*   PROGRAM CONSTANTS                                               *
*********************************************************************
BLANKS   DC    CL80' '
ERROR1   DC    C'LOAD FAILURE FOR MODULE CCIDTBL '
ERROR2   DC    C'CCIDTBL NOT PROPERLY DEFINED '
ERROR3   DC    C'INVALID CCID REQUESTED FOR THIS ENV/SYS '
         END
```

7.2.4 Sample Exit 6 Program

7.2.4.1 Overview

Exit 6 deletes the CCID table built by the @CCIDTBL macro at job termination.

```
C1UEXT06 TITLE 'Endevor for EXIT 6 EXAMPLE'
*********************************************************************
*   DESCRIPTION:  THIS PROGRAM WILL DELETE THE CCID TABLE LOADED BY   *
*                 EXIT 2 (C1UEXT02).  REVIEW C1UEXT02 FOR MORE        *
*                 INFORMATION.                                        *
*                                                                     *
*   REGISTERS ON ENTRY:                                               *
*                 0(R1) -> $ECBDS      EXIT CONTROL BLOCK             *
*                                                                     *
*   REGISTERS USAGE:                                                  *
*                 R6    -> $CCIDS      CCID TABLE                     *
*                 R7    -> $ECBDS      EXIT CONTROL BLOCK             *
*                 R12   -> BASE PROGRAM                               *
*                 R13   -> STACK USED FOR STANDARD IBM LINKAGE        *
*                                                                     *
*********************************************************************
         TITLE '$ECBDS:  EXIT CONTROL BLOCK'
*********************************************************************
*   EXIT CONTROL BLOCK                                                *
*********************************************************************
         $ECBDS
         TITLE 'CCIDDS:  CCID TABLE'
*********************************************************************
*   DSECT FOR LOAD MODULE CCIDTBL                                     *
*********************************************************************
CCIDS    DSECT
CCIDEYE  DS    CL4                     EYECATCHER 'CCID'
CCIDENV  DS    CL8                     ENVIRONMENT NAME
CCIDSYS  DS    CL8                     SYSTEM NAME
CCIDCCID DS    CL12                    CCID NAME
CCIDSLN  EQU   *-CCIDS                 DSECT LENGTH
         TITLE 'REGISTER EQUATES'
*********************************************************************
*   REGISTER EQUATES                                                  *
*********************************************************************
R0       EQU   0
```

```
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         TITLE 'MAIN:  MAINLINE LOGIC'
*********************************************************************
*   MAINLINE LOGIC                                                  *
*********************************************************************
C1UEXT06 CSECT
         SAVE  (14,12),,'EXAMPLE OF EXIT 6'
         LR    R12,R15                 POINT AT THIS PROGRAM
        USING  C1UEXT06,R12
         L     R7,0(,R1)               POINT AT $ECBDS
        USING  $ECBDS,R7
         ICM   R6,B'1111',ECBUEXT      Q: HAS TABLE BEEN LOADED?
         BZ    MAINEXIT                A: NO, RETURN TO CALLER
        USING  CCIDS,R6
         CLC   CCIDEYE,=CL4'CCID'      Q: IS THIS THE CCID TABLE?
         BNE   MAINEXIT                A: NO, DO NOT DELETE TABLE
         DELETE EP=CCIDTBL             DELETE THE CCID LOAD MODULE
         XC    ECBUEXT,ECBUEXT         CLEAR CCID TABLE ADDRESS
         SPACE
MAINEXIT DS    0H
         XR    R15,R15                 CLEAR R15 FOR NEATNESS
         RETURN (14,12)                RETURN TO CALLER
         END
```

7.3 Notification Utility Exit Programs

Several exit programs are provided with Endevor for use with the Notification utility. The source for all of the exit programs is located in the *iprfx.iqual*.source library. After tailoring them to meet the requirements of your site, these exit programs can be used in place of, or in addition to, your own programs. For those users who write their own exit programs, the samples provided with Endevor can serve as examples of the kinds of programs you would need to write.

JCL to assemble and link the exit programs is also included with Endevor. This JCL, named BC1JASMX, is located in *iprfx.iqual*.jcllib.

The following list describes each of Endevor's exit programs:

- **XIT7MAIL**—Sends notification to Endevor package approvers via the SMTP protocol (email). It can be called from the post-cast, post-execution, pre-reset, and post-backout exit points.
- **XIT7APNT**—Extracts internal approver groups and notifies each user via the Notification utility. You can choose whether you want to send the notification using the TSO, TPX, or XMIT protocol by modifying the source. It can be called from the post-cast, post-execution, pre-reset, and post-backout exit points.
- **XIT7APNC**—The COBOL version of XIT7APNT.
- **XIT7NTFY**—A package exit program that sends notification to users via the TSO protocol. You can also choose to send the notification using either the TPX or XMIT protocol by modifying the source. It can be called from any of Endevor's exit points.
- **XIT7NTFC**—The COBOL version of XIT7NTFY.
- **XIT2NTFY**—An exit program for pre- and post-action exits that notifies users via the TSO protocol. You can also choose to send the notification using either the TPX or XMIT protocol by modifying the source. It can be called from the pre- and post-action exit points.

For more information on sending messages using the Notification utility, refer to the *Exits Guide*.

Index

Special Characters

@C1UEXIT EXIT#=7 4-8
$ECBDS DSECT 1-19, 3-4
$ELMDS DSECT 3-42, 3-45
$ENVDS DSECT 3-31, 3-32
$FILDS DSECT 3-72
$PACTREQ DSECT 6-53
$PAPPREQ DSECT 6-65
$PBODREQ DSECT 6-72
$PFILDS DSECT 6-50
$PHDRDS DSECT 6-44
$PREQPDS DSECT 6-34
$PSCLREQ DSECT 6-81
$PSHPREQ DSECT 6-76
$REQPDS DSECT 3-20

A

Action
  after, in exit 3 2-11
  before, in exit 2 2-5
  codes, in COBOL 6-25
After-commit programs 4-26
Anchor IDs
  and multiple user exits 1-19
  using 6-8, 6-20
Approver
  data 6-5
  notification
    sample package exits & XIT7APNT 7-10
Archive action 4-19
Assembler
  calling conventions 5-2
  control blocks
    for exit 1 2-3
    for exit 2 2-5
    for exit 3 2-11
    for exit 4 2-15
    for exit 5 2-19
    for exit 6 2-21
Assembler *(continued)*
  control blocks *(continued)*
    list and explanation of 6-4
  ENVFCBA fields 2-7
  example of 5-3
Authorization parameter, for @C1UEXIT macro 1-3

B

Backin package action 4-20
Backout
  data 6-5, 6-73
    extracting 6-5
    set information 6-73
  dataset information 6-76
  package action 4-22
Building
  a CCID table in exit 2 7-3
  messages 2-25

C

C1BMXJCG 5-2
C1UEXITS
  source
    identify as an Endevor exit 1-18
    macro statements 1-17
    TYPE=START 1-17
  table 1-1
C1UEXT03 1-20
C1UEXT06 7-5
Calling conventions
  COBOL example 5-4
  in register 1 3-11
  supported, for user exits 1-2
Cancel an action 2-5
CAST package action 4-23
CCID table (CCIDTBL)
  in defining an action 7-5
  in exit 2 & 6 7-3

Changing
DDname 2-15
input to action 2-5
COBOL
control blocks 6-5
for exit 1 2-3
for exit 3 2-12
list and explanation of 6-5
defining multiple programs 1-3
Fields
source & target 2-9, 2-13, 2-17
representations 1-3
Collecting statistics 2-21
Collection data 6-5
Comparing input & output information 1-4
Control block layouts (COBOL)
ELEMENT-MASTER-INFO-BLOCK 3-53
EXIT-CONTROL-BLOCK 3-11, 3-13
PACKAGE-EXIT-ACTION-BLOCK 6-59
PACKAGE-EXIT-APPROVER-MAP 6-70
PACKAGE-EXIT-BACKOUT-BLOCK 6-73
PACKAGE-EXIT-BLOCK 6-25
PACKAGE-EXIT-FILE-BLOCK 6-51
PACKAGE-EXIT-HEADER-BLOCK 6-46
PACKAGE-EXIT-SCL-RECORD 6-82
PACKAGE-EXIT-SHIPMENT-BLOCK 6-78
PACKAGE-REQUEST-BLOCK 6-38
REQUEST-INFO-BLOCK 3-23
SRC-ENVIRONMENT-BLOCK 3-36
XXX-ENVIRONMENT-BLOCK 3-35
XXX-FILE-CONTROL-BLOCK 3-74
Control blocks
availability 1-2
backout package action 4-22
CREATE 4-28
DELETE exit action 4-30
DISPLAY exit action 4-31
exit passing order (diagram) 6-3
for EXECUTE 4-33, 4-34
for exit 1 2-3
for exit 2 2-5
for exit 3 2-11
for exit 4 2-16
for exit 5 2-20
for exit 6 2-21
GENPKGID exit action 4-36
initialized (matrix) 6-6
LIST exit action 4-38
MODIFY 4-40
package 6-8, 6-34, 6-44, 6-50
exit 6-8
file 6-50
Control blocks *(continued)*
package *(continued)*
header 6-44
request 6-34
passed
to ARCHIVE exit 4-19
to BACKIN exit 4-21
to CAST exit programs 4-24
primary information block 6-5
RESET 4-42
REVIEW 4-43
SHIP exit action 4-45

D

Data
flags 6-18, 6-30
Record, availability of 4-16
request types(in sequential order) 4-16
Data flags 6-30
DDnames, changing 2-15, 6-51
Debugging 1-3
Destination definition (JOB) 5-3
DSECTS
$ECBDS 3-4
$ELMDS 3-42, 3-45
$ENVDS 3-31, 3-32
$FILDS 3-72
$PACTREQ 6-53, 6-56
$PAPPREQ 6-65
$PBODREQ 6-72
$PFILDS 6-50
$PHDRDS 6-44
$PREQPDS 6-34
$PSCLREQ 6-81
$PSHPREQ 6-76
$REQPDS 3-20
SMF Action Record 3-77

E

ECB-RETURN-CODE
exit point 3 2-14
exit point 4 2-18
exit point 6 2-22
Element master block 2-9, 2-13
Element master information block 3-42, 3-53
Endevor
data flags 6-30
execution 1-16
Execution Report 3-18

Endevor *(continued)*
extracting Package Exit Control block 4-8
formatting, of control blocks 6-3
initialization 4-8
Environment block
assembler 3-31
COBOL 3-35
SRC-ENVIRONMENT-BLOCK 2-13
target & source fields 2-7
TGT-ENVIRONMENT-BLOCK 2-13
EOF conditions 4-17, 6-19
EPC1UEXT
explanation of 1-2
Error messages
ECB-MESSAGE-CODE 3-18
in current activity 6-32
Examples
of package exits 6-86
Exit
control block descriptions 6-39, 6-71, 6-74, 6-79, 6-83
entry & return 3-14
point
package, enabling 4-8
points
1 - after Endevor security checks 2-3
2 - before actions 2-5
3 - after actions 2-11
4 - data set validation 2-15
5 - at session initialization 2-19
6 - before sessions 2-21
examples 1-19
requesting invalid data 4-16
program
1 - in security violation 3-18
2 - request info. block 3-23
3 - suppressing SMF records using 3-16
4 - & DDname 3-73
identifying 1-16
reaching an end point 1-1
trace log sample 1-4
TSO operating & handling 4-8
using with Endevor 1-1
writing and storing 1-2
sample of exit 2 7-5
EXIT-CONTROL-BLOCK 3-11, 3-13

F

Field descriptions
Assembler
$PACTREQ 6-56
Field descriptions *(continued)*
Assembler *(continued)*
$PAPPREQ 6-69
$PBODREQ 6-72
$PECBDS DSECT 6-13
$PFILDS 6-50
$PHDRDS 6-45
$PREQPDS 6-35
$PSCLREQ 6-81
$PSHPREQ 6-77
ECBDS 3-5
ELMDS 3-45
ENVDS 3-32
FILDS 3-72
REQPDS 3-20
COBOL
EXIT-CONTROL-BLOCK 3-13
PACKAGE-EXIT-ACTION-BLOCK 6-62
PACKAGE-EXIT-APPROVER-MAP 6-71
PACKAGE-EXIT-BACKOUT-BLOCK 6-74
PACKAGE-EXIT-BLOCK 6-25
PACKAGE-EXIT-FILE-BLOCK 6-52
PACKAGE-EXIT-HEADER-BLOCK 6-47
PACKAGE-EXIT-REQUEST-BLOCK 6-39
PACKAGE-EXIT-SCL-BLOCK 6-83
PACKAGE-EXIT-SHIPMENT-BLOCK 6-79
REQUEST-INFO-BLOCK 3-24
SRC-ELEMENT-MASTER-INFO-BLOCK 3-57
SRC-ENVIRONMENT-BLOCK 3-36
SRC-FILE-CONTROL-BLOCK 3-74
TGT-ELEMENT-MASTER-INFO-BLOCK 3-64
TGT-ENVIRONMENT-BLOCK 3-39
TGT-FILE-CONTROL-BLOCK 3-76
File blocks 2-9, 2-13, 3-74
First exit point, reaching the 1-20
Flags, setting in $PECBDS 4-8
Foreground execution 2-5
Function codes, in COBOL 6-25

H

Historic package data 6-5

I

Indicating
enabled exit points 6-16
operating environments 4-8, 6-16
Information blocks
assembler
eighth passed 6-76
fifth passed 6-53

Information blocks *(continued)*
assembler *(continued)*
fourth passed 6-50
ninth passed 6-81
second passed 6-34
seventh passed 6-72
sixth passed 6-65
third passed 6-44
COBOL
eighth passed 6-78
fifth passed 6-59
fourth passed 6-51
ninth passed 6-82
second passed 6-38
seventh passed 6-74
third passed 6-46
primary 3-11, 6-8
Interface program 1-2
Invalid request 4-16
iprfx.iqual.SOURCE library 1-3
Issuing a message 2-25

J

JCL
BC1JXIT 1-19
exit table and sample source code 1-16
stream 5-1
JOB statement 5-1
JOBLIB library 5-1
JOBNAME 5-3

L

Language support high-level 1-2
Linkage editor input 1-2
Logging
& recording in SMF 3-77
activity 2-21

M

Macros 1-17, 1-18
C1UEXITS source 1-17
Message codes & handling 2-24, 3-17
Modifications, to JOB statement 5-1
Multiple
COBOL programs, same exit point 1-3
user exits 1-19

N

Naming conventions 1-2
NFND conditions 4-17, 6-19
Not found condition 4-16
notification utility
sample exits program 7-10

P

Package
action
& functions 4-4
codes 6-26
request block 6-53
approver request block 6-65
Backout Request Block ($PBODREQ) 6-72
exit
actions 6-3
control block 4-8
enabling points 4-8
file control block ($PFILDS) 6-50
ID 6-38
processing 4-2
SCL request block 6-81
ship request block ($PSHPREQ) 6-76
Shipment Exit 5-1
Package exits
example 6-86
PACKAGE-EXIT-BLOCK
data processing requests 4-14
setting proper fields 4-8
Passing stored information to other users 1-19
PECB-REQUEST-RETURN-CODE 4-17
Post-action exits, sample 7-10
PREQPDS 6-33
Primary information block
assembler 3-4, 6-8
COBOL 3-11, 6-20
PROCEDURE DIVISION statement 5-4

R

Reaching an exit point 1-1
RECEXIT3 1-20
Record extraction 6-5
Register 1 6-3, 6-44, 6-53
Remote production data sets 5-1
Request
a record 4-14
flags 4-14
for data flags (assembler) 6-18

Request *(continued)*
for data flags (COBOL) 6-30
REQUEST-INFO-BLOCK 3-18, 3-23
Restricting COMMIT performance 4-26
Return codes 3-17
assembler
exit 1 2-4
exit 2 2-9
exit 3 2-14
exit 4 2-17
exit 5 2-20
exit 6 2-22
COBOL
exit point 3 2-14
exit point 4 2-18
exit point 5 2-20
exit point 6 2-22
package exit 4-18
returning to, calling area 5-2

S

Sample
COBOL, linkage 1-3
exit tables 1-2
internal approver notification 7-10
SETUP Routine 4-11
sample exit 7-10
Savearea 5-2, 5-5
SCL
request block 6-81
statements 6-5
SECEXIT1, calling of 1-20
Second information block 6-34
Security
checks 2-3
violation 3-14
Session
initialization 1-1, 2-19
termination exit 2-21
Setting requests for data flags 4-16
Setup
call
in TSO & batch program (assembler) 6-16
load values 4-8
fields, in COBOL 6-28
Ship data, extracting 6-5
SMF
action record 3-3, 3-77
action request record 2-11
interface 2-4
SMF *(continued)*
record 3-1
security & interface 2-4
Source fields 2-9, 2-13, 2-17
Target fields 2-17
SRC-ENVIRONMENT-BLOCK 2-9, 2-13, 3-36
Standard linkage conventions 5-2
STEPLIB 5-1
Storage
of address values 6-20
of exits & pool reserves 1-2
recommendation 1-3
Subpools 1-2
Suppressing SMF record writing 3-16

T

Target fields 2-9, 2-13
target & source fields 2-9
Termination exit 7-5
TGT-ENVIRONMENT-BLOCK 2-9, 2-13
Trace log & facility exit 1-3
Type codes, in COBOL 6-25

U

User
exit
enabled exit points 6-28
operating environments 6-28
request flags 6-30
saving and passing info. 6-32
storing 1-3
notification, sample package exit 7-10
Using
anchor IDs 1-19
exits 2 & 6 7-2
USRNAME 5-3

V

Validating return codes 2-3
Values of Anchor IDs 1-19
Viewing passed information 1-3

W

writing exits 1-2

X

XIT2NTFY 7-10
XIT7APNT approver notification 7-10
XIT7NTFY 7-10
XXX-ENVIRONMENT-BLOCK 3-35