Endevor to CA-Endevor/DB

Bridge Administrator Guide

15.0

Computer Associates™

Contents

Chapter 1: Introduction

Chapter 2: Installation Procedures

Chapter 3: Basic Operations

Chapter 4: Using the Bridge

Chapter 5: Change History Reports

Chapter 6: Messages and Codes

Chapter

1

Introduction

Overview

Endevor to CA-Endevor/DB Bridge (the Bridge) integrates the release management facilities in the Endevor environment with the database capabilities of CA-Endevor/DB. It serves as the link between the two Endevor products that automatically capture, track, and control changes:

- Endevor — The software management product used to monitor inventory elements in a traditional non-dictionary application environment.
- CA-Endevor/DB — The product used to monitor entities in a CA-IDMS/DB dictionary environment.

Bridge Features

The Bridge aids Endevor users in the release management process by giving them access to the capabilities inherent in a database environment.

Using Endevor, you can track the history of an inventory element at the source. With the addition of the Bridge, you have access to the migration data for an inventory element and can obtain the following information:

- The identity of promoted elements and entities.
- The origin of promoted elements and entities (source).
- The destination of promoted elements and entities (target).

The Bridge allows you to log Endevor activities in a CA-Endevor/DB Change Control Database (CCDB) and to perform release management functions that enable you to monitor your development activity and maintain the integrity of your systems.

Bridge Functions

You can perform the following functions with the Bridge:

- Track the migration of an inventory element from one environment to another. For example, you can track the migration of an inventory element from a development to a production environment.
- Track and review changes to dictionary and non-dictionary entities from a central location. That is, you can log all Endevor activities in a CA-Endevor/DB Change Control Database (CCDB).
- Migrate both dictionary and non-dictionary entities under the control of the CCDB, so that when you migrate a CA-IDMS application from one dictionary to another, you can also migrate non-dictionary structures (for example, COBOL programs or CA-Culprit reports used with an CA-ADS application).
- Monitor dictionary and non-dictionary changes under one unified Change Control Identifier (CCID). You can assign a common set of CCIDs to Endevor changes and dictionary changes and simultaneously monitor both sets of entities.
- Perform an analysis of the impact of proposed changes.

The following functions are not performed by the Bridge:

- Security measures including Preauthorization, Signout/Signin, Lock, and all security class data except NM-Mode and Migrate.
- Source management for CA-IDMS data dictionary entities.

When to Use the Bridge

Your organization benefits most from the Bridge product in the following circumstances:

- You are currently using both Endevor and CA-Endevor/DB separately, and want to tie together changes made to non-dictionary entities with changes made to dictionary entities.
- You are currently using Endevor and want to take advantage of CA-Endevor/DB's release management support.

How the Bridge Executes

The Bridge program, C1DBBRDG, is an exit module that is executed in addition to other Endevor user exits. As with other Endevor user exits, the Bridge is transparent to the user.

The sequence of events at run time is as follows:

1. The Endevor user exit, if any, is executed.
2. The Bridge program, C1DBBRDG, is executed and, depending on the exit, performs as follows:
   - At Exit 2, the Bridge validates actions, using the CCDB contents.
   - At Exit 3, the Bridge logs actions in the CCDB. (If you already have your own logging process set up at this exit, double logging occurs.)
   - At Exits 5 and 6 the Bridge performs general housekeeping activities.

Note: The Bridge does not utilize Exits 1 and 4.

Commonly-Used Terms

The terms used throughout this document are part of standard CA-Endevor/DB terminology. A number of these terms are defined below for your convenience.

Term	Definition
Archive Data Set	The data set that contains output from a Endevor Archive and/or Transfer action. The archive data set can be any sequential file where RECFM = VB, LRECL has a minimum value of 800, and DSORG=PS.
CCDB	The Change Control Database in CA-Endevor/DB that maintains a complete log of all changes made to a dictionary entity. The CCDB also stores information about users, migration activity, and security structures.
C1-ELEMENT	In the Bridge, the entity type assigned to an inventory element being monitored in a CCDB. See also the definition for Inventory Element.
Dictionary	The CA-IDMS Integrates Data Dictionary (IDD) which contains data definitions, modules, documentation, and run time information for CA-IDMS components.
DBNAME /DICTNAME	In a multiple database/dictionary environment, a keyword in command syntax that identifies a particular group of databases. The group can include a CCDB, an IDD, and/or other databases. The keyword is followed by the name of the database/dictionary from which information is to be retrieved. The dictionary or database name that is specified refers to an entry in the CA-IDMS Database Name Table (DB Table).

Term	Definition
Entity	An object monitored by the Change Monitor in a CCDB. For example, an IDD object such as a dialog or a map, or a non-dictionary inventory element. An entity is identified by its name, version, and type.
Environment	A repository of definitions within Endevor. An environment comprises a Master Control File and one or more libraries that contain the source for the entities. An environment is subdivided into two stages.
Identifier	The information that uniquely identifies the entity being tracked, independent of its location. The complete identifier of an entity consists of its name, version number, and type.
CA-IDMS Central Version (CA-IDMS/CV)	In CA-IDMS, a central copy of the database manager. This mode of operation allows multiple application programs to execute concurrently, sharing a single DBMS. The Bridge always runs under CA-IDMS/CV.
Inventory Element	An object stored in an Endevor environment (previously referred to as a C1-ELEMENT). An inventory element is identified by its system, subsystem, name, type, version, and level. In a CCDB used under the Bridge, an inventory element is automatically logged with a type of C1-ELEMENT.
Level	In Endevor, a version qualifier that identifies different instances of an inventory element. For example, version 5 level 22 appears as 5.22. All levels of a given inventory element can coexist.
Stage	In Endevor, a subdivision of the environment. An inventory element can be in Stage 1 or in Stage 2. Typically, you use Stage 1 as a transient holding area where you build the inventory elements before migrating them to Stage 2. Typically, you use Stage 2 for further testing before migrating the inventory elements to a production environment.
Subsystem	In Endevor, a subgroup of a system; the secondary level within the environment hierarchy. Each inventory element is in a subsystem within a system.
System	In Endevor, a logical group of inventory elements as they apply to major applications, departments, or work areas within an organization; the top level within the hierarchy of the environment. All inventory elements are assigned to a system.

Term	Definition
Type	The form of the element or entity. Types indicate how the element/entity is created (the source language used) and how it is manipulated. For example, COBOL, JCL, and copy books are *types of inventory elements* in Endevor. Processes, dialogs, maps, and inventory elements are *types of entities* in CA-Endevor/DB.
Version	A number that identifies an iteration of an inventory element in an Endevor environment or a dictionary entity in the CCDB. Multiple versions of a single entity can coexist in a dictionary, whereas only a single version of an inventory element can exist at any one time in an Endevor environment.

Chapter 2

Installation Procedures

Overview

This chapter provides you with information about the procedures to follow in the CA-Endevor/DB and Endevor operating environments when installing the Bridge.

Prerequisites

Before installing and using the Bridge, you must successfully install Endevor, Release 3.7 (or higher) and CA-Endevor/DB, at every site involved in the migration process. It is also helpful if your staff using the Bridge have:

- An understanding of the Endevor and CA-Endevor/DB environments in your organization.
- Familiarity with the online screens and reporting capabilities of CA-Endevor/DB that display change history and migration data.

Because the Bridge is designed for use with both Endevor and CA-Endevor/DB, you will find it useful to have the following documents available for reference:

- *Endevor Installation Guide*
- *CA-Endevor/DB User Guide*
- *CA-Endevor/DB Installation Guide*
- *CA-Endevor/DB Administrator Guide*

Installation Summary

Installation of the Bridge involves the following steps:

1. In CA-Endevor/DB, run the CA-Endevor/DB install. The Bridge modules will be installed as part of the regular installation process.
2. In Endevor, perform the following tasks:
   - Customize the Defaults and Bridge Configuration control tables.
   - Define the Bridge to ISPF.
   - Modify the skeleton JCL.
3. Verify that the Bridge is operational.

CA-Endevor/DB Procedures

Installing the Bridge within CA-Endevor/DB makes the Change Monitor available for use in an Endevor environment. You use the Bridge to activate the Change Monitor. In turn, the Change Monitor records Change Log Entries (CLEs) in a designated CA-Endevor/DB Change Control Database (CCDB).

Install the Bridge

The Bridge modules are installed as a part of the normal CA-Endevor/DB product installation process. If you are installing both Endevor and CA-Endevor/DB, install Endevor first.

If you already have CA-Endevor/DB installed and are subsequently installing Endevor, run the MSGCOPY job found in the CA-Endevor/DB installation library.

Using a Standalone CCDB

In most cases, you only modify the startup JCL and recycle the system because the CA-Endevor/DB installation has already constructed the CCDBs used by the Change Monitor. However, if you have an Endevor environment that does not correspond to any CA-IDMS/DB dictionary, and you want to instrument it with the Bridge, you should refer to JOB 6 of the CA-Endevor/DB installation. You will need to duplicate portions of this job to define a new CCDB environment.

Endevor Procedures

This section discusses the steps in the Endevor environment that are needed to complete the Bridge installation. The Bridge is controlled by two tables:

- C1DBCNFG — An application-specific configuration table that establishes the correspondence between an environment and a CCDB
- C1DEFLTS — The Endevor Defaults table. In addition to providing information for these control modules, you also need to update ISPF and TSO specifications.

Read this section for the following information:

- **Constructing the Bridge configuration module** — Specifying the mapping between Endevor and CA-Endevor/DB components of the Bridge.
- **Customizing the Defaults control table** — Incorporating Bridge-specific macros in the Defaults table.
- **Defining the Bridge to ISPF** — Tailoring your ISPF logon procedure or CLIST to allocate ddnames and loadlibs for the Bridge.
- **Adding Bridge-specific data sets** — Updating the JCL library for use with the Bridge.

Constructing the Bridge Configuration Module

In this part of the installation, you code, assemble, and link-edit C1DBCNFG, the Bridge configuration table. (Sample JCL for the configuration table is stored as member C1DBCNFG in the CA-Endevor/DB JCL installation library.) To configure the Bridge for your particular application, code the following macro statements:

1. One **NDVRC1DB TYPE= MAIN** statement that contains the site identifier. The MAIN macro must be the first statement in the sequence.
2. One or more **NDVRC1DB TYPE = MAP** statements. Each MAP entry links an inventory area (identified by an environment/stage/system combination) to a CCDB or data dictionary known to a particular CA-IDMS/CV system.

   Note: You can map a single CCDB/dictionary to multiple systems and/or stages within an environment, but you cannot map a single CCDB to multiple environments. (See the example, later in this section, for more information on how to map Bridge components).
3. One **NDVRC1DB TYPE = END** statement. This must be the last macro statement in the sequence.
4. An Assembler END statement.

Code the Source Module

The syntax for the Bridge configuration module (C1DBCNFG) is listed below:

```
NDVRC1DB TYPE=MAIN,SITEID=site-number
NDVRC1DB TYPE=MAP,
         SYSCTL=idms.sysctl,
         DICTNAM=dictname,
         LOGACTN=R/O,
         CCIDVAL=R/O,
         ENVRMNT=env-name,
         STAGE#=1/2/*,
         SYSTEM=system-name
NDVRC1DB TYPE=END
END
```

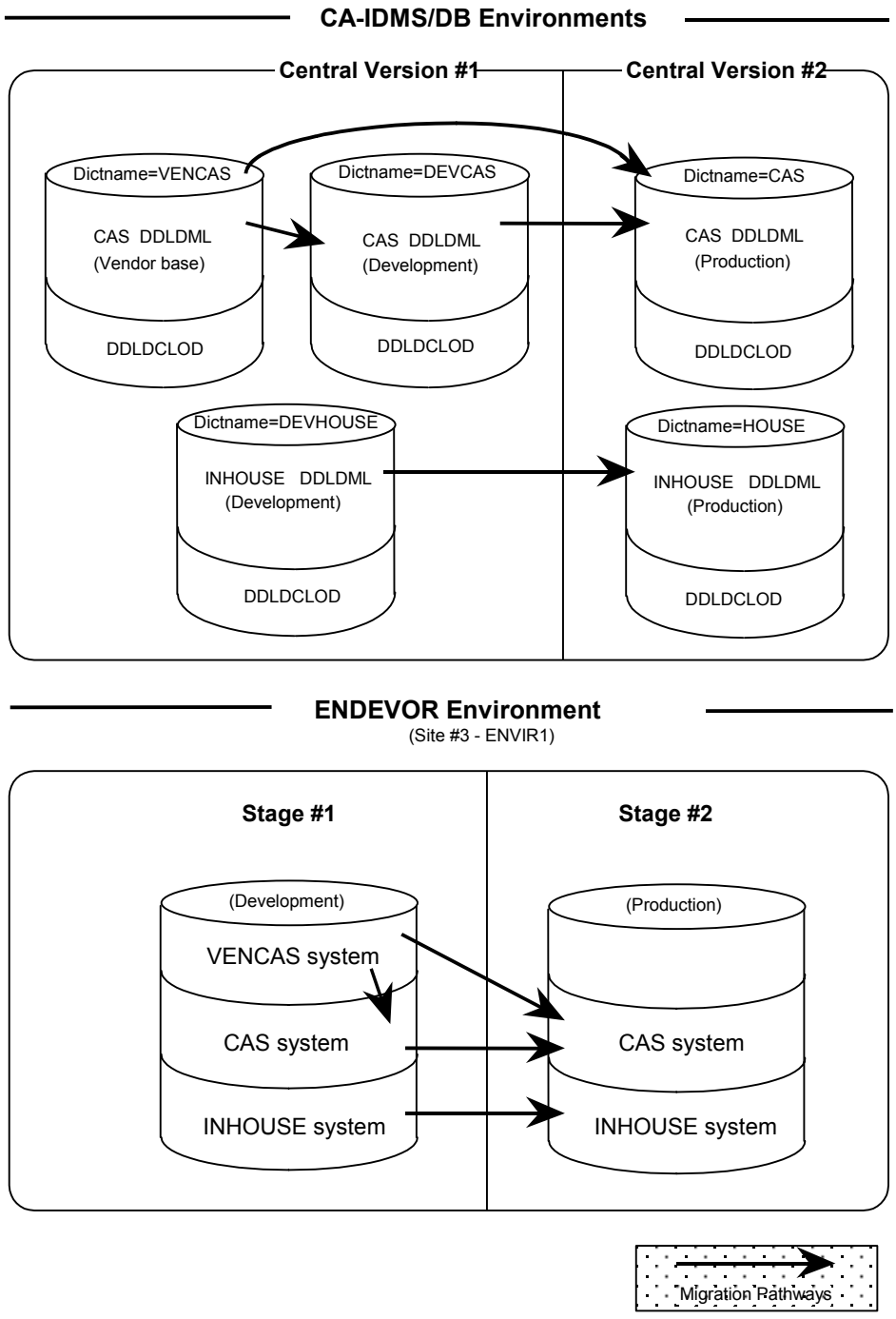
Each of the Bridge configuration module parameters and their variables are described below.

Parameter	Description	Values
SITEID = *site-number*	The identifier of the site where the entities are being migrated. This site id must match the site id value in the Defaults table.	A 1-character alphanumeric value. (Required)
SYSCTL = *idms.sysctl*	The system control file that determines which CA-IDMS/ CV system the Bridge uses at run time.	A standard data set name. (Required)
DICTNAM = *dictname*	The identifier of the IDD/CCDB that corresponds to the named Endevor inventory area. The database must be known to the CA-IDMS/CV system named in the SYSCTL parameter.	Options: ■ The 1- to 8-character name of a CA-IDMS/DB database name. ■ A blank, in quotation marks (' '), if the inventory area maps to the primary dictionary. (Required)
LOGACTN = R/O	An indication of whether the Bridge should log actions in the CCDB When logging is required, actions are denied if the Bridge cannot gain access to the appropriate CCDB	Options: ■ **R** — Logging is required; (the default). ■ **O** — Logging is optional.
CCIDVAL = R/O	An indication of whether the CCID specified in the action must be defined in the CCDB. When validation is required, actions are denied if the CCID is not known to the CCDB.	Options: ■ **R** — Always validate the CCID (the default) ■ **O** — Validation is optional.

Parameter	Description	Values
ENVRMNT = *env-name*	The name of the environment you are mapping to the CCDB/IDD identified in the DICTNAM parameter. You cannot map a single CCDB to multiple environments.	The 1- to 8-character name of the Endevor environment.
STAGE# = 1/2/*	One or both stages (within the environment named in the ENVRMNT parameter) mapping to the dictionary named on the DICTNAM parameter.	Options: ■ **1** — Map from Stage #1 ■ **2** — Map from Stage #2 ■ ***** — Map from both stages (the default).
SYSTEM = *system-name*	The Endevor system (within the environment from the ENVRMNT parameter) mapping to the dictionary named on the DICTNAME parameter.	Options: ■ The 1- to 8-character name of a specific system. ■ An asterisk (*) signifying all systems in the environment (the default).

Configuration Example

The following figure illustrates a sample scenario where you are migrating inventory elements between systems in the Development environment and between Stage 1 and Stage 2 environments.



To implement the Bridge between the Endevor and CA-Endevor/DB systems in this example, you would code the following Configuration module:

```
NDVRC1DB TYPE=MAIN,SITEID=3
NDVRC1DB TYPE=MAP,
    SYSCTL=IDMS.SYSCTL,DICTNAME=VENCAS,
    ENVRNMNT=ENVIR1,STAGE#=1,SYSTEM=VENCAS
NDVRC1DB TYPE=MAP,
    SYSCTL=IDMS.SYSCTL,DICTNAM=DEVCAS,
    ENVRNMNT=ENVIR1,STAGE#=1,SYSTEM=CAS
NDVRC1DB TYPE=MAP,
    SYSCTL=IDMS.SYSCTL,DICTNAM=DEVHOUSE,
    ENVRNMNT=ENVIR1,STAGE#=1,SYSTEM=INHOUSE
NDVRC1DB TYPE=MAP,
    SYSCTL=IDMS.SYSCTL,DICTNAM=CAS,
    ENVRNMNT=ENVIR1,STAGE#=2,SYSTEM=CAS
NDVRC1DB TYPE=MAP,
    SYSCTL=IDMS.SYSCTL,DICTNAM=HOUSE,
    ENVRNMNT=ENVIR1,STAGE#=2,SYSTEM=INHOUSE
NDVRC1DB TYPE=END
END
```

To build the Configuration table:

1. Run the JCL listed on the next page to assemble and link-edit the configuration module. As noted earlier, the sample for this job is distributed as **C1DBCNFG** in the CA-Endevor/DB installation JCL library.
2. To activate the configuration table, IPL your system or refresh the APF-authorized loadlib that contains the module.

Tip: To coordinate maintenance of the Configuration module and the Defaults table, we recommend copying the C1DBCNFG steps into **BC1JDEFT**, the job that assembles and link-edits the Defaults table. See JCL to Assemble and Link-Edit the Defaults Table, in the following section, for information on the Defaults table JCL.

JCL to Assemble and Link-Edit the Configuration Table

```
//*(Insert site-specific JOB statement)
//STEP1   EXEC PGM=your.assembler,REGION=2048K,
//             PARM='DECK,NOLOAD,NORLD,NOXREF'
//SYSPRINT  DD SYSOUT=*
//SYSLIB    DD DSN=endevor.db.distmac,
//             DISP=SHR
//SYSPUNCH  DD DSN=&&ASMOP,DISP=(NEW,PASS),UNIT=TDISK,
//             DCB=BLKSIZE=80,SPACE=(TRK,(5,1))
//SYSUT1    DD DSN=&&ASMWRK1,UNIT=TDISK,SPACE=(TRK,(5,1))
//SYSUT2    DD DSN=&&ASMWRK2,UNIT=TDISK,SPACE=(TRK,(5,1))
//SYSUT3    DD DSN=&&ASMWRK3,UNIT=TDISK,SPACE=(TRK,(5,1))
//SYSIN     DD *
  C1DBCNFG TITLE 'ENDEVOR DB BRIDGE CONFIGURATION TABLE'
           NDVRC1DB TYPE=MAIN,SITEID=site-number
           NDVRC1DB TYPE=MAP,
                    SYSCTL=idms.sysctl,
                    DICTNAM=dictname,
                    LOGACTN=R/O,
                    CCIDVAL=R/O,
                    ENVRMNT=env-name,
                    STAGE#=1/2/*,
                    SYSTEM=system-name
              .
              .
              .
           NDVRC1DB TYPE=END
           END
/*
//STEP2   EXEC PGM=IEWL,PARM='LIST,NCAL,XREF,SIZE=(256K,64K)',
//             COND=(0,NE)
//SYSLIN    DD  DISP=(OLD,DELETE),DSN=&&ASMOP
//SYSLMOD   DD  DSN=uprfx.uqual.LOADLIB(C1DBCNFG),
//             DISP=SHR
//SYSUT1    DD  UNIT=TDISK,SPACE=(CYL,(5,3))
//SYSPRINT DD  SYSOUT=*
//*
```

Where:

- ***your.assembler*** is the name of the Assembler your site uses.
- ***endevor.db.distmac*** is the name of the source library you allocated when installing CA-Endevor/DB.
- ***uprfx.uqual.LOADLIB*** is the name of the APF-authorized LINKLIST library you allocated when installing Endevor. (The load module must be named **C1DBCNFG**.)

> **Tip:** To avoid having to IPL the system or refresh the loadlib unnecessarily, we recommend that you use a Steplib to test the assembly and link-edit before actually using the APF-authorized loadlib**.**

Customizing the Defaults Table

The Defaults table (C1DEFLTS) contains the TYPE=ENVRNMNT macros that define the environments you are using. Sample JCL for the Defaults table is stored as member *uprfx.uqual*.JCL(BC1JDEFT) on the Endevor installation tape.

To customize the Defaults table:

1. Include a **TYPE=ENVRNMNT** macro for each environment. A sample from the ENVRNMNT macro is shown below.

```
C1DEFLTS TYPE=ENVRNMNT,
    .
    .
    .
    ENDBACT=Y,         E/MVS DB BRIDGE OPTION (Y/N)   X
    ENDBAVL=Y,         E/MVS DB BRIDGE OPTION (Y/N)   X
    .
    .
    .
```

2. Edit the **ENDBACT** and the **ENDBAVL** parameter in the ENVRNMNT macro. The **ENDBACT** parameter pertains specifically to the Bridge and must be turned on (set to YES) for each environment you are activating under the Bridge. The **ENDBAVL** parameter must be set to YES if the **ENDBACT** is turned on.
3. Assemble and link-edit the table, using the JCL listed in JCL to Assemble and Link-Edit the Defaults Table. Make sure the edited macros are sequenced correctly in the job stream. They must appear in the following order:
   - TYPE=MAIN macro
   - TYPE=ENVRNMNT
   - TYPE=END

   Step 1 assembles the macros and passes the assembled Defaults table to Step 2. Step 2 link-edits the table and stores the completed table in *iprfx.iqual*.LOADLIB(C1DEFLTS) where *iprfx.iqual* is the name assigned to the load library when Endevor was installed.
4. To activate the Defaults table, IPL your system or refresh the APF-authorized loadlib that contains the module.

Refer to the *Endevor Installation Guide* if you need more detailed information about the Defaults table and its parameters.

Defining the Bridge to ISPF

In order to make the components of the Bridge available during ISPF sessions, update your ISPF logon procedure or TSO CLIST as follows:

- Modify the data sets allocated for ISPLLIB.
- Optionally, add the SYSCTL data set.
- Also, if you use two or more CV systems, allocate a comparable number of data sets in your CLIST.

ISPLLIB Allocations

Modify the ALLOCATE statement for ISPLLIB as follows:

```
ALLOC F(ISPLLIB) DS( - 'endevor.loadlib', -  'idms.loadlib', -  ) SHR
```

Where:

- ***endevor.loadlib*** is the name of the load module library (created when CA-Endevor/DB was installed) that contains the CA-Endevor/DB system programs.
- ***idms.loadlib*** is the name of the load module library that contains the CA-IDMS/DB system programs (IDMSINTB, etc.).

CA-IDMS Central Version Information

Optionally, add the following statements:

```
free  f(sysctl)
alloc  f(sysctl) ds( - 'idms.sysctl'') shr
```

Where:

- ***idms.sysctl*** is the name of the SYSCTL file used to access the CA-IDMS/DB Central Version that services the CCDB database.

Note: If your Configuration table (C1DBCNFG) specifies the same SYSCTL data set name on ***all*** NDVRC1DB statements, you can allocate the SYSCTL file in your TSO CLIST. However, If you have multiple SYSCTL files, do not specify SYSCTL in the TSO CLIST.

Adding Bridge-Specific Data Sets

The installation job for Endevor creates a JCL skeleton library called *iprfx.iqual*.ISPSLIB. Within this library, you need to update member **C1SB3000**, adding two new DD statements.

Load Module Libraries

Add DD statements for STEPLIB, as follows:

```
//steplib  dd dsn=endevor.loadlib,disp=shr
//         dd dsn=idms.loadlib,disp=shr
```

Where:

- ***endevor.loadlib*** is the name of the load module library, created during the CA-Endevor/DB installation process, that contains system programs used by the Bridge.
- ***idms.loadlib*** is the name of the load module library containing the CA-IDMS system programs.

Chapter

3

Basic Operations

Overview

This chapter provides you with information that will help you understand the coordination between the CA-Endevor/DB and Endevor factions of the Bridge.

Read this chapter for the following information:

- Identifiers – Synchronizing CA-Endevor/DB and Endevor naming conventions
- Mask characters – Using mask characters for retrieving information
- Action codes – How they are translated by the Bridge
- Version/level numbers – How they are handled by the CCDB

Identifiers

The Bridge translates Endevor element names into the corresponding entity names that are used in the release management process. To request information from the CCDB online facility, you type a fully- or partially-qualified Endevor element name in the field that asks for the entity name.

System Identification

The Bridge creates audit trails and cross-reference records to reflect migration activity in the CCDBs of the target and source environments. Each environment involved in migration as a source or target is identified by a system name and a DBNAME/DICTNAME in the dictionary descriptor record contained in the CCDB. The system name is used to provide unique system identification when the source and target environments share the same dictname. This often happens when the environments involved reside on separate machines or CA-IDMS/CVs.

You can set up system names using the Dictionary Functions online submenu. See the *CA-Endevor/DB Administrator Guide* for full instructions on establishing a system name.

Endevor Element Names

An Endevor inventory element is identified by its system name, subsystem name, element name, element type, and stage number.

The Bridge concatenates the above identifiers to form the entity name used in the CCDB. This entity name is what appears online and in reports that display change history information.

For example, you could have an Endevor inventory element that has the following identifiers:

- System = **SYSTEMA**
- Subsystem = **SUBSYSA1**
- Element name = **TESTMAC1**
- Type = **COBOL**
- Stage = **1**

When the inventory element is updated and a Change Log Entry (CLE) is written to the CCDB, the Bridge translates the inventory element name into an CA-Endevor/DB entity name as follows:

```
SYSTEMA SUBSYSA1TESTMAC1  COBOL   1
```

This is the way you would identify the inventory element when requesting information from the CCDB. Note that spaces have been left after the system name, element name, and element type.

Name Segment Requirements

Each name segment has a required length. When not using delimiters, you need to pad the name segment with blanks to maintain the required length. The number of characters required for each segment is shown in the table below.

Component	Length
System name	8 characters
Subsystem name	8 characters
Element name	10 characters
Type	8 characters
Stage	1 digit (1 or 2)

Delimiters

When using the CA-Endevor/DB online facilities, you can place periods or commas as name segment delimiters. The commas separate the system name from the subsystem name, the subsystem name from the element, and so on. For example, to use delimiters when requesting the inventory element in the previous example, you would specify:

```
SYSTEMA.SUBSYSA1.TESTMAC1.COBOL.1
```

Note that you do not use spaces to pad name segments when you are delimiting with periods or commas.

Mask Characters

The mask character provides a means of retrieving entity names that match a particular criteria. You can specify as much or as little of the entity name as you want; the more information you give, the more specific the list that is returned. Use the name mask when you want to:

- **Enter partial entity names or name segments** – For example, if you know the element is in System A, but are unsure of the exact subsystem name, you could enter a partial subsystem name of SUBSYS*. This would provide you with a list of all elements in System A with subsystems that begin with the letters SUBSYS.
- **Verify that a particular element exists** – You can use the name mask to retrieve a list of all elements meeting the selected criteria to ensure that a specific element(s) does exist in the CCDB.

Using Asterisks

Asterisks (*) are used as mask characters. The information that the system retrieves depends on where you place the asterisk and the additional information you supply. The system returns entity names matching all the criteria you specify in your request.

To use the mask feature, type the beginning portion of the name segment followed immediately by an asterisk. You can place the asterisk in the following positions:

- **As the last nonblank character in the name.** For example:

  ```
  SYSTEM*
  ```

  In this example, all entities beginning with the characters S-Y-S-T-E-M are retrieved, no matter what the subsystem, element, type, or stage.

- **As the last nonblank character in a particular name segment**. For example:

  ```
  SYSTEMA.SUBSYS*.TESTMAC1.COBOL.1
  ```

  In this example, all Stage 1 COBOL inventory elements in SystemA named Testmac1 are retrieved from subsystems that begin with the characters S-U-B-S-Y-S.

- **As the last nonblank character in both the full name and a name segment.** For example:

  ```
  SYS*.SUBSYS*
  ```

  In this example, all entities are retrieved with names that begin with the characters S-Y-S**,** subsystems that begin with S-U-B-S-Y-S, and any element name, type, and stage.

- **As a single character in a name segment.** For example:

  ```
  SYSTEMA.SUBSYS*A.TESTMAC1.COBOL.1
  ```

  In this example all subsystems are retrieved that begin with the characters S-U-B-S-Y-S-, end with an A, and match the other name segments.

- **In place of a name segment.** For example:

  ```
  SYSTEMA.*.*.*.1
  ```

  In this example, all Stage 1 entities from SystemA are retrieved.

Using Periods or Commas

You can use periods or commas as a mask placeholder. For example, to see change history for all Stage 1 macro-type entities in SystemA, specify the following:

```
SYSTEMA..MACRO.1
```

In this example, the first period marks the end of the system name, the second period designates the subsystem name, and the third period designates the element name. This entry would generate a list of all the matching elements.

You can combine the comma and asterisk to retrieve an even larger list of elements. For example, to request all elements belonging to systems beginning with the characters S-Y-S in Stage 1, with a type of MACRO, and any subsystem and element name, specify the following:

```
SYS*,,,MACRO,1
```

Version and Level Numbers

An Endevor element identifier includes a version and a level number. The **version number** is designated by the user. There can be only *one* version of a particular Endevor element at a given time. The **level number** is assigned by Endevor. There can be up to 100 levels or sets of changes within a particular version.

The Bridge translates the Endevor version and level number information to an CA-Endevor/DB version number which appears in the version field of reports and change history information. For example, if an Endevor inventory element is identified as version 1, level 4 (1.4), the Change Log Entry identifies this element as a version 1 entity.

Note: The level number is recorded in the text of the Change Log Record and appears on the change history information detail screens for the Endevor element. (Refer to Chapter 5, "Change History Reports," for a discussion of the Bridge online facilities.)

Action Codes

Endevor uses action commands (for example, ADD, UPDATE, DELETE) to maintain the source and related files for inventory elements. Similarly, CA-Endevor/DB captures and tracks actions against a particular dictionary.

When an action is performed against an inventory element, the Bridge translates the action code into a CCDB equivalent. The value is recorded in the Change Log Entry, is written to the CCDB, and is placed in the Action field on reports and change history information screens. As documentation, the Endevor action command value is placed on change history information screens.

The following table lists the Endevor action commands and the equivalent CA-Endevor/DB action codes that are recorded as Change Log Entries. Note that in the case of Move, Archive, and Transfer, the Change Log Entry can vary, depending on the circumstances.

Endevor Command	Action Logged in CLEs
ADD	A(dd).
ARCHIVE	D(elete) if the entity is deleted. No action logged if the entity is not deleted.
COPY	No action logged.
DELETE	**D**(elete).
UPDATE	**M** (odify).
GENERATE	M(odify).
LIST	No action logged.
MOVE	**M**(odify) at the target. **D**(elete) at the source.
PRINT	No entry logged.
RESTORE	**M**(odify) at the target.
RETRIEVE	No entry logged.
SIGNIN	No entry logged.
TRANSFER	**M** (odify) at the target. **D**(elete) at the source.
UPDATE	**M**(odify) the target entity.

Chapter

4

Using the Bridge

Overview

This chapter provides you with information about the Bridge release management facilities and the migration process. The release management architecture, in conjunction with the Change Monitor and the Change Control Database (CCDB), controls the migration process for both dictionary and nondictionary entities.

Release management involves the following phases:

- **Phase I**: **Selection —** Creating a selection list of Endevor inventory elements for migration, based on input selection criteria and Change Log information. The selection process also enables you to analyze the impact of the proposed migration.
- **Phase II: Migration —** Transferring entities on the final selection list from the source environment to the target environment.
- **Phase III: Confirmation —** Creating audit trails for the source and target environments, documenting the origin and destination of the migrated entities.

This chapter describes these phases, particularly as they apply to the migration of Endevor inventory elements.

Prerequisites

Before you begin using the Bridge, it is useful to have made the necessary security arrangements and to know the jobs needed to implement the release management process.

Security Considerations

The following mode and migrate values are required for users executing the release management programs. They can be set in the Security Class or by using NDVR-GLOBAL:

```
MIGRATE = Y
NM-MODE = Y
```

For instructions on modifying security parameters and an explanation of the options, see the discussion of security classes in the *CA-Endevor/DB Administrator Guide.*

Programs

The following table summarizes the programs used for the different functions within the release management process.

Function	Program	Host System
Selection	NDVRDSEL	Source
Correlation/impact analysis	NDVRDCOR	Target
Migration	NDVRDLVR/NDVRC1	Source and Target
Confirmation (extract)	NDVRDCF1 NDVRDCF2	Target Source

Sample copies of the JCL for the Bridge programs are stored in the CA-Endevor/DB JCL installation library.

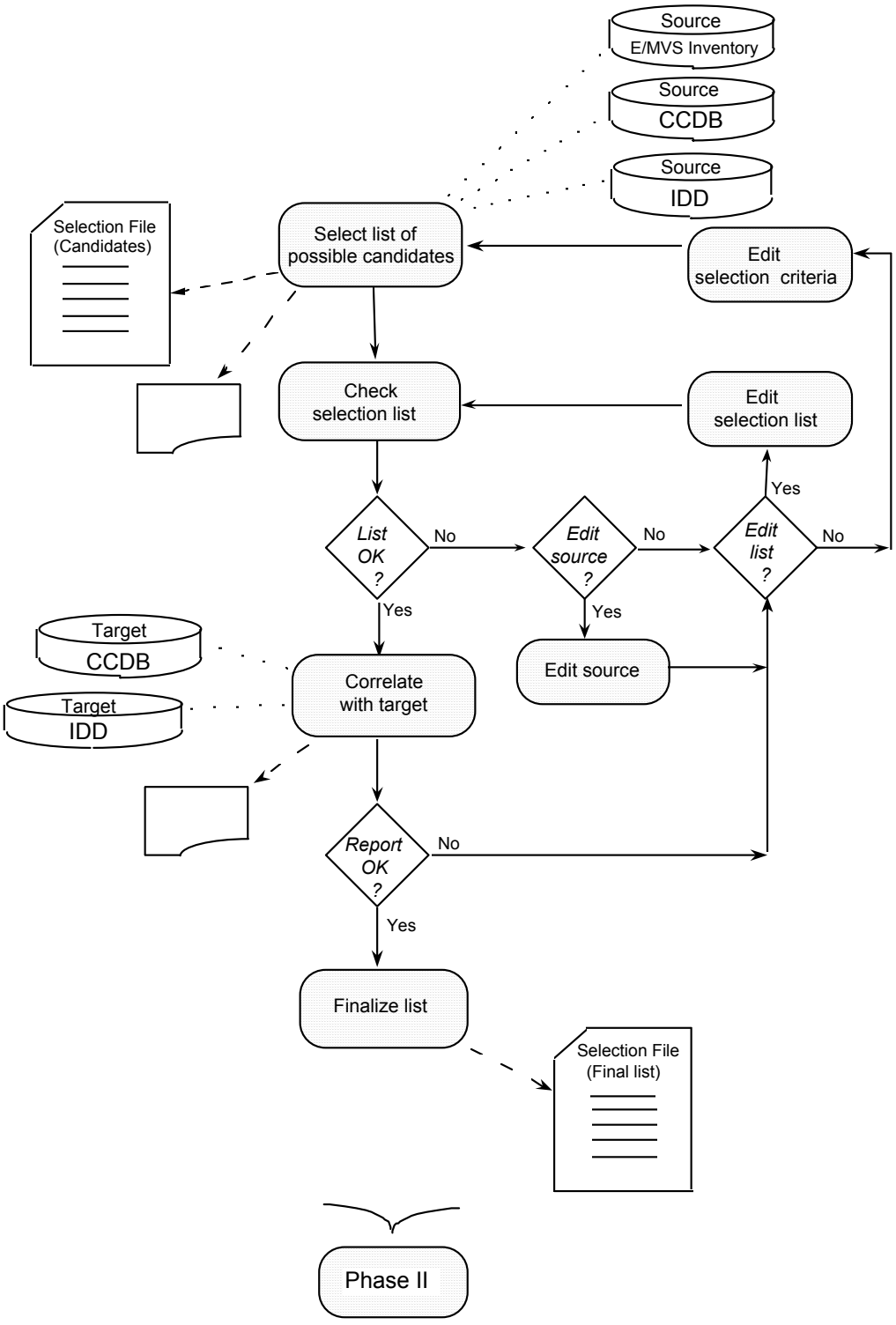
Phase I — Selection

The release management process begins by selecting entities for migration based on the information contained in the CCDB Change Log records and your input selection criteria. Only entities that have been modified in the source environment since the last migration to the target environment are initially eligible. The selection criteria for migration can include Change Control Identifiers (CCIDs), userids, status, and date/time ranges. At the successful conclusion of this phase, you have a machine-readable file of the selected dictionary and non-dictionary entities. This file is used as input to the other migration procedures. The following illustration shows the selection process.

The selection process involves the following steps:

- Step 1: Create the file of entities to migrate.
- Step 2: Modify the file, the source, and/or the selection criteria based on any exceptions reported by the Selection job.
- Step 3: Correlate the list with corresponding entities in the target environment.
- Step 4: Modify the list based on any exceptions reported by the Correlation job.
- Step 5: Finalize the migration file.

Each of these steps is discussed in detail in this section.


Source
E/MVS Inventory
Source
CCDB
Source
IDD
Selection File
(Candidates)
Select list of
possible candidates
Edit
selection criteria
Check
selection list
Edit
selection list
List
OK
?
No
Edit
source
?
No
Edit
list
?
No
Yes
Yes
Yes
Target
CCDB
Target
IDD
Correlate
with target
Edit source
Report
OK
?
No
Yes
Finalize list
Selection File
(Final list)
Phase II

Step 1: Create Selection File

Select the entities to be migrated from the source to the target environment by running the selection processor (NDVRDSEL). This job reads the signon and input command syntax in the input file (NDVRIPT) and creates a sequential Entity Out file (NDVRENO) that contains the list of entities.

Input

To select entities for the migration list:

1. Obtain a copy of the JCL for running NDVRDSEL. Sample JCL is available as SAMPDSEL in the CA-Endevor/DB installation JCL library.
2. Code a TARGET parameter to supply siteid and environment information, as follows:

```
TARGET {SYSTEM|NODENAME} [IS|=] node-name
    {DBNAME|DICTNAME} [IS|=] dictname
    SITEID = site-identifier
    ENVIRONMENT = target-env-name
```

3. Specify the following input parameters to indicate that you are using the CCDB database as input and that the file being created is not executable:

```
INPUT = DATABASE
MODE = TRIAL
```

4. Establish INCLUDE, EXCLUDE, and/or WARN selection criteria. For more information on these commands, refer to the *CA-Endevor/DB Administrator Guide*.
5. Specify one or more MIGRATE statements. To avoid confusion as to the source and target MAP entries, use specific source and target system and stage IDs for each MIGRATE statement. Note that the source and target system, subsystem, type, and stage IDs can be different, so that in effect, a MIGRATE command is a renaming rule for an inventory element. The Bridge uses this information to determine which inventory elements (if any) should be renamed when they migrate to the target system.

```
MIGRATE  [SYSTEM mvs-system]
    [SUBSYSTEM mvs-subsystem]
    [TYPE mvs-type] [STAGE[NUMBER]mvs-stage#]
    TO   [SYSTEM mvs-system]
    [SUBSYSTEM mvs-subsystem]
    [TYPE mvs-type]  [STAGE [NUMBER]mvs-stage#]
```

   The previous figure displays an example of an input file. You can specify multiple rules that work independently or in conjunction with each other. To determine if any part of the name needs to be changed when the record is moved to the target system, the Bridge systematically applies the set of rules to each source element in turn. For more information, see the examples at the end of this section.

6. Execute the NDVRDSEL job. The following is a sample input file:

```
SIGNON DICTNAME SRCDICT USER EDBADMIN.
```

```
TARGET SYSTEM = 'QA' DICTNAME = 'D'
   SITEID = '1' ENVIRONMENT = QA.
MIGRATE SYSTEM DEVELOP TO SYSTEM QA.
MIGRATE SUBSYSTEM INTERNL TO SUBSYSTEM EXTERNL.
MIGRATE TYPE COBOL TO TYPE COBXX.
INPUT IS DATABASE.
EXPAND IDD HIERARCHY RELATIONSHIPS.
INCLUDE USER = DBADMIN.
INCLUDE CCID = DEVELOP.
EXCLUDE STATUS = NEVER-MIGRATE.
MODE = TRIAL.
```

MIGRATE Statement Examples

The following examples are based on a source environment that contains the inventory elements listed in the table below:

System	Subsystem	Type	Stage
DEVELOP	9708	COBOL	1
	9801	COBOL	2

Example 1

Change the stage number only.

To rename the stage component of inventory elements migrating from Stage #2 in the source DEVELOP and TEST systems to Stage #1 in the target systems, use the following statement:

```
MIGRATE STAGE 2 TO STAGE 1.
```

Example 2

Selectively change subsystem names within a given system.

As shown in the table below, the goal of this example is to change the 9708 subsystem name to INTERNL (within the DEVELOP system only), when the inventory elements are migrated to the target system QA. Note that subsystem 9801 does not change.

	System	Subsystem	Stage	Type
Source element	DEVELOP	9708	1	COBOL
Target element	QA	INTERNL	1	COBOL
Source element	DEVELOP	9801	1	COBOL
Target element	QA	9801	1	COBOL

To accomplish this change, use the following statement:

```
MIGRATE SYSTEM DEVELOP SUBSYSTEM 9708
    TO SYSTEM QA SUBSYSTEM INTERNL.
```

Example 3

Globally change subsystem names for both systems.

This example changes the name of the 9708 subsystem to INTERNL in both the DEVELOP and the TEST systems. It also changes the type for the TEST system.

	System	Subsystem	Stage	Type
Source element	DEVELOP	9708	1	COBOL
Target element	QA	INTERNL	1	COBOL
Source element	DEVELOP	9801	1	COBOL
Target element	QA	9801	1	COBOL
Source element	TEST	9708	1	COBOL
Target element	QATEST	INTERNL	1	COBXX
Source element	TEST	9801	1	COBOL
Target element	QATEST	9801	1	COBXX

To accomplish these changes, use the following statements:

```
MIGRATE SYSTEM DEVELOP TO SYSTEM QA
MIGRATE SYSTEM TEST TYPE COBOL TO SYSTEM QATEST TYPE COBXX
MIGRATE SUBSYSTEM 9708 TO SUBSYSTEM INTERNL
```

Example 4

Selectively change system and subsystem names.

This example changes the 9708 subsystem name to EMPl in the DEVELOP system and to INTERNL in the TEST system.

		System	Subsystem	Stage	Type
Element 1:	Source	DEVELOP	9708	1	COBOL
	Target	QA	EMPL	1	COBOL
Element 2:	Source	DEVELOP	9801	1	COBOL
	Target	QA	9801	1	COBOL
Element 3:	Source	TEST	9708	1	COBOL
	Target	QATEST	INTERNL	1	COBOL
Element 4:	Source	TEST	9801	1	COBOL
	Target	QATEST	9801	1	COBOL

To accomplish these changes, use the following statements:

```
MIGRATE SYSTEM DEVELOP SUBSYSTEM 9708
              TO SYSTEM QA SUBSYSTEM EMPL.        (RULE 1)
MIGRATE SYSTEM DEVELOP TO SYSTEM QA.              (RULE 2)
MIGRATE SYSTEM TEST TO SYSTEM QATEST.             (RULE 3)
MIGRATE SUBSYSTEM 9708 TO SUBSYSTEM INTERNL.      (RULE 4)
```

MIGRATE rules are applied in the order in which you enter them. In this example, the rules are applied as follows.

- Element 1: **Rule 1 matches the system and subsystem names** and is applied. Rule 2 does not match because the system name has already been changed by Rule 1. Rule 3 does not match the system name. Rule 4 does not match because the subsystem name has already been changed by Rule 1.
- Element 2: Rule 1 does not match the subsystem name. **Rule 2 matches the system name** and is applied. Rule 3 does not match the system name and Rule 4 does not match the subsystem name.
- Element 3: Rules 1 and 2 do not match the system name. **Rule 3 matches the system name** and is applied. **Rule 4 matches the subsystem name** and is applied.
- Element 4: Rules 1 and 2 do not match the system name. **Rule 3 matches the system name** and is applied. Rule 4 does not match the subsystem name.

Output

When the NDVRDSEL job executes successfully, it produces the following output:

- The **Selection File** (NDVRENO) with control information identifying the source and target Change Control Databases (CCDBs), and the list of entities selected for migration.

  The Selection File has an ENT (entity) statement for each entity named with an INCLUDE statement. The ENT statement is commented out with an asterisk (*) if that entity is also EXCLUDEd. You can edit this file with any source code editor.

- The **Control Report** set (NDVRLST) that includes:
  - The *Input Command Listing* that displays the commands you specified in the input command file (NDVRIPT).
  - The *Compiled Command Listing* that displays the input command file as seen by the command interpreter within NDVRDSEL.
  - The *Entity List Exception Listing* that itemizes entities with WARN or EXCLUDE conditions.
  - The *End-of-Job Statistics* summary with information related to the work the system performed during execution of the job.

- The **Detail Report** (NDVRDTL) that lists all the entities and a summary of their rules and statistics. This report is particularly useful if you need to edit the Selection File.
- The **Utility Report** (NDVRUTL) that flags potential problem areas requiring further investigation. It lists all closely related entities modified out of sequence. This report is primarily for the CA-Endevor/DB administrator.

Step 2: Modify Selection File

Based on the reports you received as output in Step 1, you may want to make modifications to the Selection file. If you are satisfied with the results after reviewing the Entity List Exception Listing and/or the Detail report, continue to Step 3.

Otherwise, your options are as follows:

- Edit the inventory source
- Edit the existing selection file, or
- Rebuild the selection file.

Edit the Inventory Source

Make the necessary changes, if any, to your Endevor inventory elements. Then, edit the existing selection file or rebuild the selection. See the following sections for more information.

Edit the Existing Selection File

1. Use a standard editor to modify the Selection File. The figure shown in Phase II - Migration shows the syntactical format of the output file. You can manually add entities by inserting an ENT statement. To include an EXCLUDEd entity that is listed in the file, remove the leading asterisk (*). The syntax for the entity statement is shown below; be sure to place all ENT statements after the LIST FOLLOWS command.

   ```
   ENT type entity-name vvvv.
   ```

   Note: Endevor inventory elements always have an entity type of C1-Element.

2. Execute NDVRDSEL using the following input parameters:

   ```
   MODE = TRIAL.
   INPUT = FILE.
   ```

   These parameters cause the Selection File to be read back into NDVRDSEL as NDVRENI. The program processes the edited file against the selection criteria and the CCDB produces the following:

- A new Selection File, output as NDVRENO.
- A new set of reports.

3. Review the reports and continue the validation process until you are satisfied.

Rebuild the Selection File

1. Edit the selection criteria parameters.
2. Run NDVRDSEL using the following input parameters:

```
    MODE = TRIAL.
    INPUT = DATABASE.
SOURCE SYSTEM [NAME] [IS | =] source-system-name
      {DBNAME | DICTNAME} [IS | =] dictname | ' '
       VERIFY DATE = MM/DD/YY TIME = hh:mm:ss.
TARGET {SYSTEM|NODENAME} [IS|=] node-name
   {DBNAME|DICTNAME} [IS|=] dictname
    SITEID [IS|=] site-identifier
    ENVIRONMENT [IS|=] target-env-name.
MIGRATE [SYSTEM mvs-system ] [SUBSYSTEM mvs-subsystem]
     [TYPE mvs-type]  [STAGE[NUMBER]mvs-stage#]
  TO  [SYSTEM mvs-system ] [SUBSYSTEM mvs-subsystem]
     [TYPE mvs-type ]  [STAGE [NUMBER]mvs-stage#] .
     .
     .
     .
[MODE     [IS | =]  {TRIAL | EXECUTE | BACKOFF} .]
[INPUT    [IS | =]  {FILE | DATABASE} .]
[SIGNOUT  [TO] {USER | CCID} [NAME]  [IS | =]  signout-name ]
[INCLUDE
          [FROM [DATE]  [IS | =]  mm/dd/yy]
          [THRU [DATE]  [IS | =]  mm/dd/yy]
          [ALL]
            [{MGRP | CCID | USER} [NAME][IS | =] select-name
          ['']
          [WHERE STATUS [NAME]  [IS | =] status-value]  ].
[EXCLUDE    [WHERE] STATUS [NAME]  [IS | =] status-value
             [WITHIN CCID [NAME]  [IS | =] status-value]  ].
[EXPAND    IDD {CHANGE | HIERARCHY} [RELATIONSHIPS]. ]
[WARN       [WHERE]
          [CCID  [IS | =] MULTIPLE]
          [CCID  [IS | =] NULL]
          [USER  [IS | =] MULTIPLE]
          [USER  [IS | =] NULL]  .]
LIST FOLLOWS  .
 ENT type entity name vvvv.
 ENT type entity name vvvv.
      .
      .
      .
```

Step 3: Correlate File with Target

When you have finished editing the selection list, run the correlation processor to produce an impact analysis report. The correlation job examines the history of changes in the target environment to determine if candidates have been modified since they were last migrated from the source environment. Also, it determines which entities might be adversely affected by the migration, or might prevent successful transfer. In this manner, reversion of applied fixes or parallel development conflicts are captured, and discrepancies can be resolved before they cause problems.

Input

To correlate the selection list with the target and produce an impact analysis report:

1. Obtain a copy of the JCL for running NDVRDCOR. This program reads the entity names and control information contained in the Selection File. A sample job, SAMPDCOR, is available in the CA-Endevor/DB installation JCL library. (The ddname of the Selection File is NDVRENI.)
2. Use the input file (NDVRIPT) to supplement the control statements in the Selection File. Refer to the *CA-Endevor/DB Administrator Guide* for more information about the SIGNON, IGNORE, and EXPAND statements.
3. Execute the NDVRDCOR program.

Output

When the job executes successfully, you end with the following output:

- The **Control Report** set (NDVRLIST) that includes:
  - The *Input Command Listing* that displays the commands you specified in the input command file (NDVRIPT).
  - The *Compiled Command Listing* that combines the statements in the Selection File (NDVRENI) with those in the input file (NDVRIPT).
  - The *Migration Entity Exceptions* that lists entities on the Selection list that have been modified in the target system since they were last migrated. If no migration activity is logged in the CCDB, the last migration is assumed to have occurred at the beginning of recorded history for the target. (Note that this report only appears when there are exceptions.)
  - The *Expansion Entity Exceptions* report that lists related entities that have been modified since the last migration. This report is for the CA-Endevor/DB administrator.

  - The *End-of-Job Statistics* summary with information relating to the work performed by the system during execution.
- The **Detail Report** set (NDVRDTL) that includes:
  - The *Compiled Command Listing*.
  - The *Correlation Detail Listing* that itemizes all entities from the target that were involved in the impact analysis. Any IGNORE rules you applied to exception conditions are listed in this report.
  - The *End-of-Job Statistics* summary for the correlation job.
- The **Utility Report** set (NDVRUTL) that includes:
  - The *Input Entity List File* listing that records the control information and the list of entities in the Selection File (NDVRENI).
  - The *Target Entity Exceptions* report that lists any entities that will stop the migration process. (Note that this report only appears if you use the EXPAND statement and it only applies to CA-Endevor/DB entities.)
  - The *End-of-Job* Statistics summary for the correlation job.

Step 4: Modify Selection File

Based on the reports you received as output in Step 3, you may want to make modifications to the Selection File. If you are satisfied with the results after reviewing the reports, continue to Step 5.

Otherwise, your options are as follows:

- Edit the inventory source, or
- Edit the existing Selection File.

Edit the Inventory Source

1. Make any necessary changes to your Endevor inventory elements.
2. Optionally, build a new Selection File, rebuild the existing file, or proceed to Step 5.

Edit the Existing Selection File

1. Use a standard editor to modify the Selection File. You can manually add entities by inserting an ENT statement. To include an EXCLUDEd entity that is listed in the file, remove the leading asterisk (*).

   Syntax for the entity statement is shown below; be sure to place all ENT statements after the LIST FOLLOWS command.

```
ENT type entity-name vvvv.
```

Note: Endevor inventory elements always have an entity type of C1-ELEMENT.

2. Run NDVRDSEL using the following input parameters:

```
INPUT = FILE.
MODE = TRIAL.
```

3. Review the reports and edit the file, continuing the validation process until you are satisfied.
4. Run NDVRDCOR to correlate the new file with the target.

Step 5: Finalize Selection File

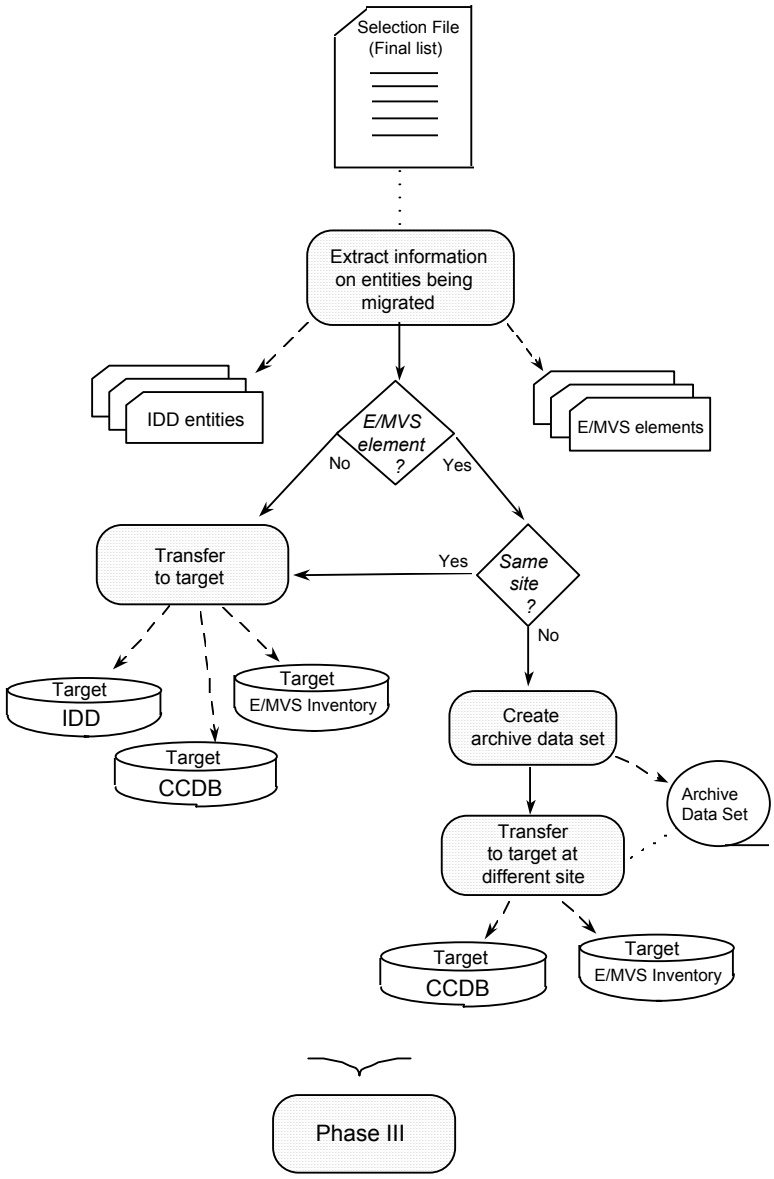
To obtain the final Selection File that can be used as input to the migration processor, execute NDVRDSEL using the following input parameters:

```
INPUT = FILE.
MODE = EXECUTE.
```

In Execute mode, the Selection File you produced in Trial mode is read in as input (NDVRENI), and the final Selection File is produced as NDVRENO.

Phase II — Migration

You can now use the finalized selection list to migrate the entities. The Delivery processor, NDVRDLVR, builds transfer files and the Endevor batch processor, NDVRC1 updates the target environment. During migration, the Change Monitor on the target environment is run in a special migration mode to indicate that the environment modifications are the result of migration. The figure below summarizes the migration process.



Input for the Source Job

To begin the migration process, create transfer files as follows:

1. Obtain a copy of the JCL for running NDVRDLVR/NDVRC1. A sample job, SAMPC1J1, is available in the CA-Endevor/DB installation JCL library.
2. Specify the necessary information for the input file (NDVRIPT). To set SCL options, you need to include the following parameter; code it immediately after the SIGNON statement.

   ```
   SET SCL [OPTIONS] option option option ...
   ```

where:

option can be any value supported by the Endevor SET OPTIONS command that applies to Move and Transfer actions. In Release 3.8 of Endevor, these values are as follows:

BYPASS DELETE PROCESSOR	FORCE
BYPASS ELEMENT DELETE	COMMENT '*comment-string*'
BYPASS GENERATE PROCESSOR	OVERRIDE SIGNOUT
CCID *ccid-name*	SYNCHRONIZE
IGNORE GENERATE FAILED	WITH HISTORY

The remaining command language syntax for the input file is detailed in the *CA-Endevor/DB Administrator Guide*.

3. Execute NDVRDLVR/NDVRC1.

Output from the Source Job

The migration process produces the following output for Endevor environments.

- An **SCL file** that contains Endevor Restore commands for each inventory element being migrated.
- If the target environment is at another site, an **SCL file** that contains Endevor Archive commands for each inventory element being migrated.
- The **Control Report** set (NDVRLST) that includes:
  - An *Input Command Listing* of the syntax you specified in the NDVRIPT input file.
  - An *Entity File Listing*.
  - A *Process Summary*.
- A standard output listing from the Endevor batch processor.

Input for the Target Job

Run NDVRC1 in the target environment. A sample job, SAMPC1J2, is available in the CA-Endevor/DB installation JCL library.

Output from the Target Job

After the job is run, you have the following output:

- Migration Change Log Entries (CLEs) on the target environment are assigned a CLE action code of V, and are stamped with a footprint containing the exact date and time received, the source system identifier, and the date and time the inventory element was selected at the source environment.
- If the target environment is at a different site, the job produces an **archive data set** that contains the source name and information for the inventory elements being migrated.

Phase III — Confirmation

After the migration is implemented successfully, you extract a confirmation file from the target. This file identifies the target name, date and time of the migration, and each entity that was migrated. At the source, this file is used to create migration Change Log Entries (CLEs). A complete audit trail of the migration now exists on the source and the target.

Migration CLEs are used by future Selection and Correlation jobs as part of the ongoing quality assurance/maintenance cycle. In addition, the CLEs are used for comprehensive migration activity reports.

Step 1: Create a Confirm File

The confirmation file extract program reads the Selection File to determine which CLEs to extract from the target CCDB. Extracts are done on the basis of source system, dictname, and date/time values.

To extract information for the confirmation file:

1. Obtain a copy of the JCL for NDVRDCF1. A sample job, SAMPDCF1, is available on the CA-Endevor/DB installation JCL library.
2. Execute the NDVRDCF1 job.

Step 2: Create Source CLEs

Use the file you created in the previous step as input for the next job, the job that completes the confirmation process at the source. This job creates migration Change Log Entries in the source CCDB (the CLE action code = C), and signs in the migrated entities to make them available for entry into the development cycle.

To complete the confirmation process:

1. Obtain a copy of the JCL for NDVRDCF2.
2. Execute the NDVRDCF2 job.

You have now completed the Release Management cycle for this group of entities.

Chapter

5

Change History Reports

Overview

The CA-Endevor/DB Change Monitor creates an audit trail of all updates made to a CCDB. You can review change history information in both online and batch modes, using the CA-Endevor/DB online facilities and/or selected CA-Culprit reports. This chapter describes both the online and batch reporting facilities.

Online Reporting

You can review Change Log Entries (CLEs), the entries that document the changes made to an entity, on the following CA-Endevor/DB screens:

- **Entity and Entity Change History** (option 4) — Displays all changes made to selected entities, in chronological order.
- **CCID and CCID Change History** (option 5) — Displays all changes made under a specific CCID, in chronological order.
- **User and User Change History** (option 7) — Displays all changes made by a specific user, in chronological order.
- **Dictionary and Dictionary History** (option 8) — Displays all changes made in the dictionary, in chronological order.

Because procedures and information are similar for all four options, it is not necessary to go through each screen in detail. Instead, this section displays the screens for the Entity and Entity Change History option, intending that you use the accompanying explanation as a sample for the other screens.

Main Menu

```
CA-ENDEVOR/DB  15.0 CAABE0            MAIN FUNCTION MENU             1/15/99 NDVRU000
USER ===> EDBADMIN               DICTNAME ===> SRCDICT              MODE ===> UPDATE


OPTION ===> 4

             1  - SIGNIN/SIGNOUT FUNCTIONS
             2  - AUTHORIZATION FUNCTIONS
             3  - LOCK FUNCTIONS
             4  - ENTITY AND ENTITY CHANGE HISTORY
             5  - CCID AND CCID CHANGE HISTORY
             6  - STATUS AND STATUS ASSOCIATIONS
             7  - USER AND USER CHANGE HISTORY
             8  - DICTIONARY AND DICTIONARY HISTORY
             9  - MANAGEMENT GROUPS AND CCIDS
            10  - ENDEVOR/DB CONTROL FUNCTIONS
            11  - ENDEVOR/DB SIGNON FUNCTION
            12  - RETURN TO IDMS/DC
```

To view Change Log information:

1. Start your CA-Endevor/DB session.
2. Type **4** in the OPTION field of the Main Function Menu to select the Entity and Entity Change History.
3. Press ENTER. CA-Endevor/DB displays the Entity Functions menu.

Entity Menu

```
CA-ENDEVOR/DB  15.0 CAABE0              ENTITY FUNCTIONS              1/15/99 NDVRU400
USER ===> EDBADMIN                 DICTNAME ===> SRCDICT              MODE ===> UPDATE


OPTION ===> 5

  1  - BROWSE ENTITY DESCRIPTORS          2  - ADD A NEW ENTITY DESCRIPTOR
  3  - MODIFY ENTITY DESCRIPTORS          4  - DELETE ENTITY DESCRIPTORS
  5  - BROWSE ENTITY CHANGE HISTORY       6  - BROWSE ENTITY STATUS HISTORY

ENTITY:                                            (IF OPTIONS 1, 2, 3, 4, 5, 6 )
   NAME        ===>
   TYPE        ===> C1
   VERSION     ===>

CHANGE-LOG SELECTION CRITERIA:                     (IF OPTIONS 5, 6 )
   START DATE  ===>                                END DATE ===> 1/15/99
   START TIME  ===>                    END TIME ===>
   ACTION CODE ===>

DICTIONARY     ===> SRCDICT                          (IF OPTION 5 )
```

To browse change history information for inventory elements:

1. Type **5** (Browse Entity Change History) in the OPTION field.
2. Type **C1** in the entity TYPE field. (C1 is an acceptable abbreviation for C1-ELEMENT, the type that identifies an Endevor inventory element.)
3. Optionally, provide information that further refines the list of entries you want to display. Refer to the *CA-Endevor/DB User Guide* for information about specific fields on the screen.

   Note: When requesting an entity that is an Endevor inventory element (that is, has a type of C1-Element), specify all components of the name: system, subsystem, element name, type, and stage. To review name components, valid delimiters, and the use of mask characters, see Chapter 3, "Basic Operations."
4. Press ENTER to display a list of all changes made to MVS entities to date.

The List Screen

The Entity/Change-Log Entry/User/CCID List screen displays changes, by entity, in descending chronological order where the latest changes are listed first.

```
CA-ENDEVOR/DB  15.0 CAABE0   ENTITY/CHANGE-LOG ENTRY/USER/CCID LIST   1/15/99  NDVRU420
USER ===> EDBADMIN                DICTNAME ===> SRCDICT         MODE ===> UPDATE


ACTION ===> BROWSE
             ENTITY NAME              TYP VERS LOG-ENTRY   USER        CCID
_ FINANCE GENLEDG FINGL01    COB     1 C1     1 D 08/19/97 EDBADMIN
s FINANCE GENLEDG FINGL02    COB     1 C1     1 I 08/16/97 EDBADMIN
_ FINANCE GENLEDG FINGL03    COB     1 C1     1 O 08/16/97 EDBADMIN
_ FINANCE GENLEDG PROGX      COB     1 C1     1 D 08/19/97 EDBADMIN
_ FINANCE GENLEDG PROGDATE   COB     1 C1     1 I 07/19/97 EDBADMIN
_ FINANCE GENLEDG PROGSOB    COB     1 C1     1 O 07/16/97 EDBADMIN
_
```

To select the entities for which you want change history information:

1. Type any nonblank character next to one or more of the listed entities.
2. Press ENTER to display the Detail screen.

The Detail Screen

The Entity/Change-Log Entry/User/CCID Detail screen provides you with change history information for each of the entities you selected on the previous screen. The screen displays entity information (name, version, and type), the information that is in the Change Log for the particular entity, user information (name, security class, and current CCID), and information on the current CCID.

```
CA-ENDEVOR/DB  15.0 CAEABE0 ENTITY/CHANGE-LOG ENTRY/USER/CCID DETAIL  1/15/99  NDVRM420
USER ===> EDBADMIN                DICTNAME ===> SRCDICT          MODE ===> UPDATE


ACTION ===> BROWSE
*****************************  ENTITY INFORMATION  *****************************
NAME    ===> FINANCE GENLEDG FINGL01   COB     1              VERSION ===>    1
TYPE    ===> C1-ELEMENT
COMMENT ===>
************************  CHANGE-LOG ENTRY INFORMATION  ***********************
DATE    ===> 08/19/97           TIME ===> 7:56:24            ACTION  ===> D
COMMENT ===>
SITE/ENV ===> 2 DEMO       VV.LL ===> 01.00  ACTN ===> DELETE   RCODE ===> 0000
*****************************  USER INFORMATION  *****************************
NAME         ===> EDBADMIN                                     LOCKED  ===> N
SECURITY CLS ===> NDVR-GLOBAL
CURRENT CCID ===>

COMMENT      ===>
*****************************  CCID INFORMATION  *****************************
NAME    ===>               SECURITY CLASS ===>                    LOCKED ===>
COMMENT ===>
```

To display the information you want, you can navigate between the screens as follows:

- From the Detail screen:
  - If you selected more than one entity, press ENTER to display the next selection.
  - Press ENTER to return to the List screen when all entities are displayed.
- From the List screen:
  - Select additional entities to review and press ENTER to return to the Detail screen.
  - To return to the Entity menu, press ENTER when no entities are marked for selection.

A field-by-field description of the two sections that apply specifically to the Bridge is listed below. For a complete explanation of the other fields on the Detail screen, refer to the chapter on entity descriptors in the *CA-Endevor/DB User Guide.*

Entity Information

This section of the Detail screen displays the following information about the entity that was changed.

Field	Information Displayed
Name	The CA-Endevor/DB entity name created for the Endevor inventory element, consisting of system, subsystem, element name, type, and stage.
Version	Version number for the entity.
Type	When using the Bridge, the type for an inventory element is always C1-ELEMENT.
Comment	Any comments associated with the entity.

Change-Log Entry Information

This section of the Detail screen displays the following information about the Change Log Entry.

Field	Information Displayed
Date	The date the change log entry was written.
Time	The time the change log entry was written.
Action	The CA-Endevor/DB action that resulted in the change entry. (The command displayed in the C1 CMD field is the Endevor equivalent of this CA-Endevor/DB action.)
Comment	Any comments associated with this change log entry, created from the comments associated with the Endevor action.
C1 CMD	The Endevor action that initiated the change log entry. (The code listed in the ACTION field is the CA-Endevor/DB equivalent of this command.) Refer to Chapter 3, "Basic Operations," for the list of Endevor actions used in the Bridge and the corresponding CA-Endevor/DB codes.
VV.LL	The Endevor version and level number.
C1 RCODE	The Endevor return code.

Batch Reporting

Change history and related information is also available by running CA-Culprit reports. The reports provide you with additional details about Endevor inventory elements, Change Log Entries, and pre- and post-migration activity.

The following table lists the reports that pertain to the Bridge. A sample of each report is displayed in this section.

Report Number	Report Title	Module Name
51	Chronological Change Log Detail	NDVRPT51
52	User Change Log Detail	NDVRPT52
53	CCID Change Log Detail	NDVRPT53
54	Endevor Element Change Log Detail	NDVRPT54
65	Endevor Element Post Migration Activity	NDVRPT65
66	Endevor Element Source Migration Summary	NDVRPT66
67	Endevor Element Target Migration Summary	NDVRPT67

Running the Reports

To run a report, include the following =COPY commands in the CA-Culprit job stream:

```
//SYSIN     DD   *
=COPY NDVRpt00            ¬Global housekeeping module
=COPY NDVRname            ←Module with company name
=COPY NDVRRCLG            ←Input module for Change Log
=COPY NDVRpt nn           ←Report module name
/*
```

To produce multiple reports in a single job run, include an =COPY command for each report you want.

For example, to run the Chronological Change-Log Detail (report 51) and the User Change-Log Detail (report 53), code the following SYSIN file:

```
    //SYSIN     DD   *
=COPY NDVRPT00
=COPY NDVRNAME
=COPY NDVRRCLG
=COPY NDVRPT51
=COPY NDVRPT53
/*
```

Refer to the chapter on reports in the *CA-Endevor/DB User Guide* for details on the SYS010 input file parameters and a complete listing of the JCL for running the Culprit job.

Selection Criteria

When you run a CA-Culprit report job, the system retrieves all relevant change log entries in the CCDB unless you specify additional selection criteria. The following table lists the Change-Log fields you can reference in a SELECT WHEN statement.

Field	Values	Example
LOG DATE	8 character date formatted as yyyymmdd	SEL WHEN LOG DATE GT 19970815
USER	1-8 character user name in single quotes	SEL WHEN USER EQ 'JONES'
ACTION	1 character code (A, M, D, R, S, T, P, C, V) in single quotes	SEL WHEN ACTION EQ 'V'
CCID	1-12 character change control ID in single quotes	SEL WHEN CCID 'GL*'AND 'AP*'
ELEMENT SYSTEM	1-8 character name of the Endevor system in single quotes	SEL WHEN ELEMENT-SYSTEM EQ 'PRODSYS'
ELEMENT SUBSYSTEM	1- to 8-character name of the subsystem in single quotes.	SEL WHEN ELEMENT SUBSYSTEM EQ 'FINANCE'
ELEMENT NAME	1- to 10-character name of the element in single quotes.	SEL WHEN ELEMENT NAME EQ 'PARTS-*'
ELEMENT TYPE	1- to 8-character name of the Endevor type in single quotes.	SEL WHEN ELEMENT-TYPE EQ 'MACRO'
ELEMENT STAGE	1 character numeric (1, 2) of the Endevor stage in single quotes.	SEL WHEN ELEMENT STAGE EQ '2'

The SELECT statement follows the last =COPY statement in the input stream. It must begin in column 2.

Chronological Change-Log Detail (Report 51)

The Chronological Change Log Detail report lists each CLE logged in the CCDB, in chronological order. For each environment, within a particular site, the report supplies the date and time the CLE was logged; an action code indicating the type of change (when appropriate, the Endevor action, return code, status, and comments are also included); the entity name, represented by its Endevor components (system, subsystem, element, type, and stage identifiers); the version and level of the inventory element (VV.LL); the identifier of the user who made the change; and the Change Control ID of the entity.

The report sample displayed below was produced with the following SYSIN file:

```
    //SYSIN     DD   *
=COPY NDVRPT00
=COPY NDVRNAME
=COPY NDVRRCLG
=COPY NDVRPT51
 SEL WHEN ELEMENT NAME EQ 'TESTMAC'
/*
```

```
REPORT NO. 51                        COMPUTER ASSOCIATES INTERNATIONAL, INC.                1/27/99  12:21 PAGE     1
CA=ENDEVOR/DB 15.0 CAABE0           ENDEVOR CHRONOLOGICAL CHANGE-LOG REPORT
                                        ** PUT YOUR COMPANY NAME HERE **
CHANGE LOG FOR SITEID: 0 ENVIRONMENT: QA
DATE --- TIME ---   ACTION ----------------------    SYSTEM    SUBSYS    ELEMENT    TYPE      STG   VV.LL     USER ---  CCID -----
12/22/98 07:56:17   A  (C1 ADD      RCODE 0000)     FINANCE   GENLEDG   GLREPORT   COB        1     1.00     EDBADMIN  YR2000

01/16/99 10:28:11   A  (C1 ADD      RCODE 0000)     FINANCE   GENLEDG   C1DBCNFG   ASM        1     1.00     EDBADMIN  YR2000

01/16/99 10:31:33   M  (C1 MOVE     RCODE 0000)     FINANCE   GENLEDG   C1DBCNFG   ASM        1     0.00     EDBADMIN  YR2000

01/16/99 10:31:33   M  (C1 MOVE     RCODE 0000)     FINANCE   GENLEDG   C1DBCNFG   ASM        2     1.00     EDBADMIN  YR2000

01/16/99 10:33:51   D  (C1 DELETE   RCODE 0000)     FINANCE   GENLEDG   C1DBCNFG   ASM        2     1.00     EDBADMIN  YR2000
```

User Change-Log Detail (Report 52)

The Endevor User Change-Log Detail report lists the change log entries associated with a given user. The sample report displayed below was produced with the following SYSIN file:

```
    //SYSIN     DD   *
=COPY NDVRPT00
=COPY NDVRNAME
=COPY NDVRRCLG
=COPY NDVRPT52
 SEL WHEN USER EQ 'JSBUid1'
/*
```

```
REPORT NO. 52                    COMPUTER ASSOCIATES INTERNATIONAL, INC.                    01/27/99  15:29 PAGE     1
CA-ENDEVOR/DB 15.0 CAABE0        ENDEVOR USER CHANGE-LOG DETAIL REPORT
                                      ** PUT YOUR COMPANY NAME HERE **
CHANGE MADE BY USER: EDBADMIN
DATE --- TIME ---   ACTION ----------------------  SITE  ENVIRON   SYSTEM    SUBSYS    ELEMENT     TYPE      STG  VV.LL  CCID -----
12/22/98 07:56:17   A  (C1 ADD       RCODE 0000)   0    QA        FINANCE   GENLEDG   GLREPORT    COB       1    1.00   YR2000

01/16/99 10:28:11   A  (C1 ADD       RCODE 0000)   0    QA        FINANCE   GENLEDG   C1DBCNFG    ASM       1    1.00   YR2000

01/16/99 10:31:33   M  (C1 MOVE      RCODE 0000)   0    QA        FINANCE   GENLEDG   C1DBCNFG    ASM       1    0.00   YR2000

01/16/99 10:53:35   M  (C1 MOVE      RCODE 0000)   0    QA        FINANCE   GENLEDG   C1DBCNFG    ASM       2    1.00   YR2000

01/16/99 10:33:47   D  (C1 DELETE    RCODE 0000)   0    QA        FINANCE   GENLEDG   S1RPT07     ASM       2    1.00   YR2000
```

CCID Change-Log Detail (Report 53)

The Endevor CCID Change-Log Detail report lists each Endevor entity update associated with a given CCID. The entries, listed in ascending chronological order, include the action associated with the update, the site, environment, system, subsystem, element name type and stage of the entity, its version and level, and the name of the user that initiated the change.

The report sample displayed below was produced with the following SYSIN file:

```
    //SYSIN     DD   *
=COPY NDVRPT00
=COPY NDVRNAME
=COPY NDVRRCLG
=COPY NDVRPT53
 SEL WHEN CCID EQ 'BSTMAINT'
/*
```

```
REPORT NO. 53                    COMPUTER ASSOCIATES INTERNATIONAL, INC.                    01/27/99 15:31 PAGE     1
CA-ENDEVOR/DB 15.0 CAABE0        ENDEVOR CCID CHANGE-LOG DETAIL REPORT
                                      ** PUT YOUR COMPANY NAME HERE **
CHANGE MADE UNDER CCID: YR2000
DATE --- TIME ---   ACTION ----------------------  SITE  ENVIRON   SYSTEM    SUBSYS    ELEMENT     TYPE      STG  VV.LL  USER ---
12/22/98 16:10:05   A  (C1 ADD       RCODE 0000)   0    QA        FINANCE   GENLEDG   GLREPORT    COB       1    1.00   EDBADMIN

01/16/99 10:28:15   A  (C1 ADD       RCODE 0000)   0    QA        FINANCE   GENLEDG   C1DBCNFG    ASM       1    1.00   EDBADMIN

01/16/99 10:31:04   D  (C1 MOVE      RCODE 0000)   0    QA        FINANCE   GENLEDG   C1DBCNFG    ASM       1    1.00   EDBADMIN

01/16/99 10:31:33   D  (C1 MOVE      RCODE 0000)   0    QA        FINANCE   GENLEDG   C1DBCNFG    ASM       2    1.00   EDBADMIN

01/16/99 10:33:56   D  (C1 DELETE    RCODE 0000)   0    QA        FINANCE   GENLEDG   C1DBCNFG    ASM       2    1.00   EDBADMIN
```

Element Change-Log Detail (Report 54)

The Endevor Element Change-Log Detail report lists each Change Log entry for a given inventory element. The sample report displayed below was produced with the following SYSIN file:

```
    //SYSIN     DD   *
=COPY NDVRPT00
=COPY NDVRNAME
=COPY NDVRRCLG
=COPY NDVRPT54
 SEL WHEN ENTITY NAME EQ
 * 'SYSTEM1 SYBSYSLATESTMAC   MACRO   1'
/*
```

```
REPORT NO. 54                  COMPUTER ASSOCIATES INTERNATIONAL, INC.          01/27/99  15:32 PAGE   1
CA-ENDEVOR/DB 15.0 CAABE0     ENDEVOR ELEMENT CHANGE LOG DETAIL REPORT
                                 ** PUT YOUR COMPANY NAME HERE **
CHANGE LOG FOR ELEMENT: 0,QA,FINANCE ,GENLEDG, C1DBCNFG  ,ASM     ,1   VERSION:  0
DATE --- TIME --- ACTION -------  USER --- CCID ------- ADDITIONAL CHANGE-RELATED INFORMATION -----------------------------------

01/16/99 10:31:33 D (C1 MOVE    ) EDBADMIN YR2000       ELEMENT VV.LL:    1.00   C1 RETURN CODE: 0000
```

Element Post Migration Activity (Report 65)

The Endevor Element Post Migration Activity report contains a list of each inventory element modified in the target environment since it was last migrated. This report is instrumental in identifying production or quality assurance "fixes" made independently of the development system. The Correlation Processor uses a similar extraction path when identifying the potential reversion of these fixes by new migrations.

```
REPORT NO. 65                        COMPUTER ASSOCIATES INTERNATIONAL,INC.            01/27/99 PAGE     1
CA-ENDEVOR/DB 15.0 CAABE0       ENDEVOR ELEMENT POST MIGRATION ACTIVITY REPORT
                                      ** PUT YOUR COMPANY NAME HERE **
CCDB DICTNAME:  DEMO       AT SITE/NODE:  TUFTS
                                                                                  CHANGE        DATE LAST      TIME LAST
SITE  ENVIRON    SYSTEM     SUBSYS     ELEMENT      TYPE       STG   VV.LL          ACTIONS          ACTION         ACTION

2     DEMO       FINANCE    GENLD01    TESTMAC      MACRO       1     1.02              1        01/27/99       12:33:13
2     DEMO       FINANCE    GENLD01    TESTMAC      MACRO       2     1.03              1        01/27/99       12:33:17
  RECORDS WRITTEN FOR REPORT 65 --         9
```

Element Source Migration Summary (Report 66)

The Endevor Element Source Migration Summary is run against the source environment and contains information on each inventory element received by the target environment, including the number of times it was exported and date it was last exported. All inventory elements participating in a migration are displayed in sequence by element name.

```
REPORT NO. 66                              COMPUTER ASSOCIATES INTERNATIONAL,INC                          01/27/99 PAGE     1
CA-ENDEVOR/DB 15.0 CAABE0                ENDEVOR ELEMENT SOURCE MIGRATION SUMMARY
                                            ** PUT YOUR COMPANY NAME HERE **
CCDB DICTNAME:  QA         AT SITE/NODE:  TUFTS
]------- ENDEVOR     ELEMENTS TRACKED IN THIS CCDB --------]  TIMES DATE LAST ]---------- TARGET OF MIGRATION ----------]  DATE TO
SITE ENVIRON  SYSTEM   SUBSYS    ELEMENT    TYPE     STG VERS     OUT  EXPORTED SITE ENVIRON  SYSTEM   SUBSYS   TYPE     STG  TARGET
 2   QA       SYSTEM1  SUBSYS1A TESTMAC    MACRO     1  1.00       1  01/16/99   1  DEMO     FINANCE  GENLD01  MACRO     1 01/16/99
                                                                   1  01/16/99   2  DEMO     FINANCE  GENLD01  MACRO     1 01/27/99
 2   QA       SYSTEM1  SUBSYS1A TESTMAC    MACRO     2    1       1  01/16/99   2  DEMO     FINANCE  GENLD01  MACRO     2 01/16/99
 2   QA       SYSTEM1  SUBSYS1A TESTMAC2   MACRO     2    1       1  06/01/99   1  DEMO     FINANCE  GENLD01  MACRO     2 01/16/99
  RECORDS WRITTEN FOR REPORT 66 --       10
```

Element Target Migration Summary (Report 67)

The Endevor Element Target Migration Summary is produced from the perspective of the target environment and contains a list of each inventory element received from the source environment, the number of times it was imported, and the date it was last imported. All inventory elements participating in a migration are listed in sequence within the C1-ELEMENT type.

```
REPORT NO. 67                          COMPUTER ASSOCIATES INTERNATIONAL, INC.                          01/27/99 PAGE     1
CA-ENDEVOR/DB 15.0 CAABE0                ENDEVOR ELEMENT TARGET MIGRATION SUMMARY
                                           ** PUT YOUR COMPANY NAME HERE **
CCDB DICTNAME:  DEMO       AT SITE/NODE:  TUFTS
]------- ENDEVOR     ELEMENTS TRACKED IN THIS CCDB -------]  TIMES DATE LAST ]---------- SOURCE OF MIGRATION ----------] DATE FROM
SITE ENVIRON  SYSTEM   SUBSYS   ELEMENT    TYPE     STG VERS     IN  IMPORTED SITE ENVIRON  SYSTEM   SUBSYS   TYPE     STG SOURCE
 2   QA       DEMO     DEMO     TESTMAC    MACRO     1  1.00      3  01/02/99  1  QA       SYSTEM1  SUBSYS1A MACRO     0 01/12/99
                                                                  1  01/02/99  1  QA       SYSTEM1  SUBSYS1A MACRO     1 01/12/99
                                                                  1  01/02/99  1  QA       SYSTEM1  SUBSYS1B MACRO     1 01/12/99
                                                                  1  12/29/98  2  QA       SYSTEM1  SUBSYS1A MACRO     1 01/12/99
 2   DEMO     FINANCE  GENLEDG  TESTMAC    MACRO     2  1.00      1  01/12/99  1  QA       SYSTEM1  SUBSYS1A MACRO     0 01/12/99
                                                                  2  01/12/99  2  QA       SYSTEM1  SUBSYS1A MACRO     2 01/12/99
 2   DEMO     FINANCE  GENLEDG  TESTMAC2   MACRO     2  1.00      3  01/12/99  1  QA       SYSTEM1  SUBSYS1A MACRO     0 01/12/99
                                                                  1  01/12/99  1  QA       SYSTEM1  SUBSYS1A MACRO     2 01/12/99
                                                                  1  01/12/99  1  QA       SYSTEM1  SUBSYS1B MACRO     2 01/12/99  RECORDS WRITTEN
FOR REPORT 67 --       15
```

Chapter

6

Messages and Codes

Overview

When the Bridge encounters any problems during execution, it issues error messages. During online processing, the error condition is indicated by an ISPF message in the upper right corner of the screen. You can request the long form of the ISPF error message by pressing **PF1** (the HELP PF key). The full message is displayed on the second line of the screen.

The ISPF error messages are a summary of the full text of the error message. The full message text appears in:

- The Execution report, the action listing produced by job C1SB3000 during batch processing.
- The ***userid.CC1TEMPRn.MSGS*** data set, the online equivalent of the Execution Report. You can browse this data set when the error is encountered.

Message Number Format

Each message is identified by a message number in the format **C1HnnnnC C1DBxxxx**, where:

- **C1H —** Identifies the message as a Bridge message.
- **nnnn —** Specifies the 4-digit message number.
- **C —** Indicates that this message has a severity level of **Caution (return code 08).** "Caution" means that the system encountered an error that may prevent further processing. The request was completed, but the result is probably not what was intended.
- **C1DB —** Identifies the source of the message as the Endevor Bridge.
- **xxxx —** Indicates which module encountered the error. It can have a value of **BRDG** or **LOAD**.

For additional information about message format and conventions, refer to the *Endevor Messages and Codes* manual.

The remainder of this chapter lists each of the messages issued by the Bridge. The messages are presented in ascending order, within the **C1DBBRDG** or **CLDBLOAD** module issuing the messages.

C1DBBRDG Messages

The following messages are issued by the C1DBBRDG module.

C1H0001C C1DBBRDG: INVALID $ECBDS STRUCTURE

Indicates that an attempt to use both ENDEVOR and the Bridge involves incompatible releases, or indicates that there may be an internal ENDEVOR system error. Check the release numbers of the Endevor and CA-Endevor/DB systems in use at the time the error occured to make sure they are compatible. If they are, contact CA-Endevor/DB Technical Support.

C1H0002C C1DBBRDG: INVALID $REQPDS STRUCTURE

Indicates that an attempt to use both Endevor and the Endevor Bridge involves incompatible releases, or indicates that there may be an internal Endevor system error. Check the release numbers of the Endevor and CA-Endevor/DB systems in use at the time the error occurs and make sure they are compatible. If they are, contact CA-Endevor/DB Technical Support.

C1H0003C C1DBBRDG: INVALID $ENVDS STRUCTURE

Indicates that an attempt to use both Endevor and the Bridge involves incompatible releases, or indicates that there may be an internal Endevor system error. Check the release numbers of the Endevor and CA-Endevor/DB systems in use at the time the error occurs and make sure they are compatible. If they are, contact Endevor Technical Support.

C1H0006C

C1DBBRDG: INVALID $C1DBDS STRUCTURE

Indicates that an attempt to use both Endevor and the Bridge involves incompatible releases, or indicates that there may be an internal Endevor system error. Check the release numbers of the Endevor and CA-Endevor/DB systems in use at the time the error occurs and make sure they are compatible. If they are, contact CA-Endevor/DB Technical Support.

C1H0007C

C1DBBRDG: INVALID EXIT-ID - FUNC= action (nn)','

Indicates that an attempt to use both Endevor and the Endevor Bridge involves incompatible releases, or indicates that there may be an internal Endevor system error. Check the release numbers of the Endevor and CA-Endevor systems in use at the time the error occurs and make sure they are compatible. If they are, contact CA-Endevor/DB Technical Support.

C1H0008C

C1DBBRDG: CCIDVAL=R, BUT CCID ccid IS NOT DEFINED IN THE CCDB

You have specified a CCID on the action request that is not defined in the Change Control Database. The C1DBCNFG table with which you are currently working specifies CCIDVAL=R for your current Environment, System and Stage Number. Therefore, all action CCIDs must be predefined. Check with your Endevor administrator for the proper name/spelling of the CCID and repeat the action.

C1H0009C

C1DBBRDG: INVALID $BRGDS STRUCTURE

Indicates an internal Endevor Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0010C

C1DBBRDG: $BGETSTG SERVICE REQUEST FAILED - RESULT=result

Indicates an internal Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0011C

C1DBBRDG: OPEN FOR NDVRENI FILE FAILED - RESULT=result

Indicates an internal Endevor Bridge system error. The C1DBBRDG module failed to open the NDVRENI file (the Entity List file) although a DD statement for the file is present. Make sure that the DD statement identifies an NDVRENI file produced by the NDVRDSEL job and that the file has not been corrupted. If the problem persists, contact CA-Endevor/DB Technical Support.

C1H0012C

C1DBBRDG: NDVRDSYN CALL FAILED - RCODE=nnnn - CHECK NDVRUTL OUTPUT

The C1DBBRDG has found and opened the NDVRENI file and the subroutine that reads the NDVRENI file has returned an error condition. Check the NDVRUTL listing for error messages and correct the indicated problem.

C1H0013C

C1DBBRDG: NDVRENI FILE "SOURCE" OR "TARGET" STATEMENT MISSING

The C1DBBRDG has found and read the NDVRENI file. The NDVRENI file is not correctly formatted. It is missing either a SOURCE or TARGET statement. Make sure that the NDVRENI file has been produced by the NDVRDSEL job and/or correct the file. Then, resubmit the job.

C1H0014C

C1DBBRDG: NDVRENI FILE MODE = TRIAL OR BACKOFF NOT ALLOWED

The NDVRENI file was created by NDVRDSEL running in TRIAL mode. Rerun NDVRDSEL in EXECUTE mode to create a new NDVRENI file, and then resubmit the job.

C1H0015C

C1DBBRDG: ATTEMPT TO PROCESS AN ELEMENT OUTSIDE NDVRENI SOURCE OR TARGET ENVIRONS

The NDVRENI file SOURCE and TARGET statements specify Environments. When you run the C1SB3000 job and specify the NDVRENI file, you must process only entities at the migration source environment or at the target. If you are not performing migration processing, remove the NDVRENI DD statement from the job stream.

C1H0017C

C1DBBRDG: NDVRENI FILE PROCESSING FAILED - REQUEST DENIED

The Migration processing failed during the handling of a previous action, and C1DBBRDG has consequently shut down. This message will be issued on all subsequent actions. Find and correct the original error.

C1H0018C

C1DBBRDG: OPEN FOR NDVRUTL FILE FAILED - RESULT=result (DD STATEMENT MISSING?)

The C1DBBRDG has found and read the NDVRENI file. The first event is to open the NDVRUTL listing file. The open for this file has failed. The most likely cause is a missing DD statement. If this is the case, correct the JCL and resubmit the job. If the NDVRUTL DD statement is present, this message indicates an internal Bridge system error. In this situation, contact CA-Endevor/DB Technical Support.

C1H0019C

C1DBBRDG: $BFRESTG SERVICE REQUEST FAILED - RESULT=result

Indicates an internal Endevor Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0020C

C1DBBRDG: C1DBCNFG TABLE SITEID (c) AND TARGET SITEID (c) MUST BE THE SAME

The C1DBBRDG has found and read the NDVRENI file. The NDVRENI file contains a TARGET statement that specifies a SITEID. The Endevor job, however, is running at a different site. You must run the migration target C1SB3000 job at the site specified in the NDVRENI file.

C1H0021C

C1DBBRDG: $BDELETE SERVICE REQUEST FAILED - RESULT=result

Indicates an internal Endevor Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0022C

C1DBBRDG: ATTEMPT TO LOG A CHANGE OUTSIDE THE NDVRENI SOURCE OR TARGET CCDBS

The C1DBBRDG has found and read the NDVRENI file. The NDVRENI file SOURCE and TARGET statements specify DICTNAME values, which identify CCDBs. When you run the C1SB3000 job, the C1DBCNFG table determines where each change will be logged. When you run the C1SB3000 job and specify the NDVRENI file, you must process only entities whose C1DBCNFG mapping leads to the source or target CCDB. If you are not performing migration processing, then remove the NDVRENI DD statement from the job stream.

C1H0023C

C1DBBRDG: C1DBCNFG TABLE HAS BEEN CORRUPTED

Indicates an internal Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0024C

C1DBBRDG: C1DBCNFG INDICATES A "CV SWITCH" IS NEEDED, BUT "DYNAMIC SYSCTL ALLOCATION" IS NOT ACTIVE

You have submitted a C1SB3000 job that has a SYSCTL DD statement in the JCL, or you are running online and have a permanent SYSCTL allocation for your TSO session. After inspecting the C1DBCNFG table, C1DBBRDG has determined that it must switch SYSCTL files. Because of the DD statement (or permanent allocation), it cannot do so. You must either restrict your Endevor processing to elements that are all monitored by CCDBs in the same CA-IDMS/DB CV, or else remove the DD statement and resubmit the job (or FREE the allocation and reenter Endevor).

C1H0025C

C1DBBRDG: SYSCTL SVC99 DSN=data-set-name ERR=xxxx INFO=xxxx

The ENDEVOR Bridge has attempted to dynamically allocate one of the SYSCTL files identified in the C1DBCNFG control table. The service request (SVC99) has failed. Refer to the IBM operating system publication for a definition of the error number and information codes.

C1H0026C

C1DBBRDG: SYSCTL SVC99 DSN=data-set-name ERR=xxxx INFO=xxxx

The ENDEVOR Bridge has attempted to dynamically free one of the SYSCTL files identified in the C1DBCNFG control table. The service request (SVC99) has failed. Refer to the IBM operating system publication for a definition of the error number and information codes.

C1H0027C

C1DBBRDG: LINK TO C1DBLOAD FAILED - RESULT= rrrrrrrr

The C1DBBRDG has attempted to call the C1DBLOAD module, but the call failed. The result value indicates the nature of the problem ("NOTFOUND", "MEM ERR", etc.). Make sure that your region size is sufficiently large, your JCL specifies the correct load module libraries, and resubmit the job. If the problem persists, contact CA-Endevor/DB Technical Support.

C1H9999C

C1DBBRDG: E998 SYSTEM ERROR - ENDEVOR SERVICE REQUEST NOT PROCESSED

The C1DBBRDG module has issued a service request to the Endevor change monitor, which was not processed. Check the CA-IDMS/DB CV log for possible messages that indicate the cause of the problem. If the problem persists, contact your CA-Endevor/DB Technical Support Representative.

C1DBLOAD Messages

The following messages are issued by the LOAD module.

C1H0901C

C1DBLOAD: INTERNAL ERROR (RAN OUT OF LLES)

Indicates an internal Endevor Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0902C

C1DBLOAD: INTERNAL ERROR (NO CDE FOR LLE)

Indicates an internal Endevor Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0903C

C1DBLOAD: C1DBCNFG NOT LOADED FROM AN AUTHORIZED LIBRARY

As distributed, the Endevor Bridge requires that the C1DBCNFG control table be loaded from an APF-authorized library (which should always be the same library from which the Endevor C1DEFLTS control table is loaded). If the control table is loaded successfully, but not from an authorized library, you will get this message and the Endevor session will be terminated. If desired, your Bridge administrator can modify the system to eliminate this requirement.

C1H0904C

C1DBLOAD: MULTIPLE COPIES OF C1DBCNFG LOADED

Indicates an internal Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0905C

C1DBLOAD: LOAD FOR C1DBCNFG FAILED - RESULT= rrrrrrrr

The Bridge has attempted to load the C1DBCNFG control table, but the load failed. The result value indicates the nature of the problem ("NOTFOUND," "MEM ERR," etc.). Make sure that your region size is sufficiently large, and/or that your JCL specifies the correct load module libraries, and resubmit the job. If the problem persists, contact CA-Endevor/DB Technical Support.

C1H0906C

C1DBLOAD: INVALID C1DBCNFG TABLE

The Endevor Bridge has loaded a module named C1DBCNFG, but the module is not a valid control table. Make sure that Endevor is processing with the correct load libraries, and that the C1DBCNFG module in those libraries was created by running the job documented in Chapter 2, “Installation Procedures.” If the problem persists, contact CA-Endevor/DB Technical Support.

C1H0907C

C1DBLOAD: C1DBCNFG IS INVALID - ENVRMNT=environment AND ENVRMENT=environment MAP TO THE SAME CCDB

The Endevor Bridge has loaded the C1DBCNFG control table, but has found that two of the MAP entries name the same Endevor environment and the same CCDB. This is not supported. Correct the specification of the control table in the C1DBCNFG assembly and link job, and re-create the table.

C1H0909C

C1DBLOAD: LOAD FOR C1DFLTS FAILED - RESULT= rrrrrrrr

The Bridge has attempted to load the C1DEFLTS control table, but the load failed. Indicates an internal Endevor Bridge system error. Contact CA-Endevor/DB Technical Support.

C1H0910C

C1DBLOAD: SITEID VALUES IN C1DBCNFG (c) AND C1DEFLTS (c) ARE NOT THE SAME

The Bridge has loaded the C1DBCNFG and C1DEFLTS control tables. They specify different SITEID values, and therefore cannnot be used together. It is recommended that you place the two tables in the same load library, and combine the assembly and link steps for the two tables in the same job. Find the correct tables, place them in the appropriate library, and resubmit the job.