CA IDMS - 19.0
Administrating CA Endevor and DB for Bridge

Date: 15-Jan-2018
This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or 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 YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2017 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
# Table of Contents

**Bridge Functions** ................................................................. 6  
When to Use the Bridge ............................................................... 7  
How the Bridge Executes ............................................................. 7  

**Commonly-Used Terms** .......................................................... 7  

**Installation Procedures** .......................................................... 9  
Overview ....................................................................................... 10  
Prerequisites ................................................................................ 10  
  Security Considerations ........................................................ 10  
  Programs .................................................................................... 10  
Installation Summary ................................................................. 11  
CA Endevor/DB for CA IDMS Procedures ....................................... 11  
  Install the Bridge .................................................................... 11  
    Using a Standalone CCDB ...................................................... 11  
CA Endevor SCM Procedures ....................................................... 12  
  Constructing the Bridge Configuration Module ......................... 12  
    Code the Source Module ....................................................... 13  
    Configuration Example ......................................................... 14  
    Building the Configuration Table ......................................... 16  
  JCL to Assemble and Link-Edit the Configuration Table ............... 16  
  Customizing the Defaults Table .............................................. 17  
    Defaults Table Customization Steps ..................................... 17  
  Defining the Bridge to ISPF ..................................................... 18  
    ISPLLIB Allocations ............................................................. 18  
    CA IDMS Central Version Information ................................... 18  
  Adding Bridge-Specific Data Sets ............................................ 19  
    Load Module Libraries ......................................................... 19  

**Basic Operations** .................................................................... 19  
Overview ....................................................................................... 19  
Identifiers ...................................................................................... 20
Using the Bridge ................................................................. 24
Overview ........................................................................... 25
Prerequisites ....................................................................... 25
Phase I -- Selection .............................................................. 25
Step 1 Create Selection File .................................................. 28
  Input ............................................................................... 28
  MIGRATE Statement Examples ......................................... 29
  Output ............................................................................ 31
Step 2 Modify Selection File .................................................. 31
  Edit the Inventory Source ............................................... 32
  Edit the Existing Selection File ......................................... 32
  Rebuild the Selection File ................................................ 32
Step 3 Correlate File with Target ............................................ 33
  Input ............................................................................... 33
  Output ............................................................................ 33
Step 4 Modify Selection File .................................................. 34
  Edit the Inventory Source ............................................... 35
  Edit the Existing Selection File ......................................... 35
Step 5 Finalize Selection File ................................................ 35
Phase II -- Migration ............................................................. 35
  Input for the Source Job .................................................. 37
  Output from the Source Job .............................................. 37
  Input for the Target Job .................................................... 38
  Output from the Target Job .............................................. 38
Phase III -- Confirmation .................................................... 38
  Step 1 Create a Confirm File ............................................. 38
  Step 2 Create Source CLEs .............................................. 39
Administrating CA Endevor and DB for Bridge

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

Using CA Endevor SCM, 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 CA Endevor SCM activities in a CA Endevor/DB for CA IDMS 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 (see page 6)
  - When to Use the Bridge (see page 7)
  - How the Bridge Executes (see page 7)
  - Commonly-Used Terms (see page 7)

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 CA Endevor SCM activities in a CA Endevor/DB for CA IDMS 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 for CA IDMS reports used with a 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 CA Endevor SCM 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 CA Endevor SCM and CA Endevor/DB for CA IDMS separately, and want to tie together changes made to non-dictionary entities with changes made to dictionary entities.
- You are currently using CA Endevor SCM and want to take advantage of CA Endevor/DB for CA IDMS's release management support.

How the Bridge Executes

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

The sequence of events at run time is as follows:

1. The CA Endevor SCM 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 for CA IDMS terminology. A number of these terms are defined below for your convenience.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive Data Set</td>
<td>The data set that contains output from a CA Endevor SCM 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.</td>
</tr>
<tr>
<td>CCDB</td>
<td>The Change Control Database in CA Endevor/DB for CA IDMS that maintains a complete log of all changes made to a dictionary entity. The CCDB also stores information on users, migration activity, and security structures.</td>
</tr>
<tr>
<td>C1-ELEMENT</td>
<td>In the Bridge, the entity type assigned to an inventory element being monitored in a CCDB. See also the definition for Inventory Element.</td>
</tr>
<tr>
<td>Dictionary</td>
<td>The CA IDMS Integrated Data Dictionary (IDD) which contains data definitions, modules, documentation, and run time information for CA IDMS components.</td>
</tr>
<tr>
<td>DBNAME/DICTNAME</td>
<td>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).</td>
</tr>
<tr>
<td>Entity</td>
<td>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.</td>
</tr>
<tr>
<td>Environment</td>
<td>A repository of definitions within CA Endevor SCM. 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.</td>
</tr>
<tr>
<td>Identifier</td>
<td>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.</td>
</tr>
<tr>
<td>CA IDMS Central Version</td>
<td>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.</td>
</tr>
<tr>
<td>Inventory Element</td>
<td>An object stored in a CA Endevor SCM environment (previously referred to as a C1-ELEMENT). An inventory element is identified by its system, subsystem, name, type, version, and level.</td>
</tr>
<tr>
<td>Level</td>
<td>In CA Endevor SCM, 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.</td>
</tr>
<tr>
<td>Stage</td>
<td>In CA Endevor SCM, 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.</td>
</tr>
<tr>
<td>Subsystem</td>
<td>In CA Endevor SCM, a subgroup of a system; the secondary level within the environment hierarchy. Each inventory element is in a subsystem within a system.</td>
</tr>
</tbody>
</table>
### Term | Definition
--- | ---
System | In CA Endevor SCM, 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.
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 CA Endevor SCM. Processes, dialogs, maps, and inventory elements are types of entities in CA Endevor/DB for CA IDMS.
Version | A number that identifies an iteration of an inventory element in a CA Endevor SCM 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 a CA Endevor SCM environment.

- Installation Procedures (see page 9)
- Basic Operations (see page 19)
- Using the Bridge (see page 24)

## Installation Procedures

### Contents
- Overview (see page 10)
- Prerequisites (see page 10)
  - Security Considerations (see page 10)
  - Programs (see page 10)
- Installation Summary (see page 11)
- CA Endevor/DB for CA IDMS Procedures (see page 11)
  - Install the Bridge (see page 11)
    - Using a Standalone CCDB (see page 11)
- CA Endevor SCM Procedures (see page 12)
  - Constructing the Bridge Configuration Module (see page 12)
    - Code the Source Module (see page 13)
    - Configuration Example (see page 14)
    - Building the Configuration Table (see page 16)
  - JCL to Assemble and Link-Edit the Configuration Table (see page 16)
  - Customizing the Defaults Table (see page 17)
    - Defaults Table Customization Steps (see page 17)
  - Defining the Bridge to ISPF (see page 18)
    - ISPLLIB Allocations (see page 18)
    - CA IDMS Central Version Information (see page 18)
  - Adding Bridge-Specific Data Sets (see page 19)
    - Load Module Libraries (see page 19)
Overview

This section provides you with information on the procedures to follow in the CA Endevor/DB for CA IDMS and CA Endevor SCM operating environments when installing the Bridge.

Prerequisites

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

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

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

- *CA Endevor/DB for CA IDMS Using section*
- *Installing section -- z/OS*
- *Administrating section*

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 *Administrating section*.

Programs

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Program</th>
<th>Host System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection</td>
<td>NDVRDSEL</td>
<td>Source</td>
</tr>
<tr>
<td>Correlation/impact analysis</td>
<td>NDVRDCOR</td>
<td>Target</td>
</tr>
<tr>
<td>Migration</td>
<td>NDVRDLVR/NDVRC1</td>
<td>Source and Target</td>
</tr>
</tbody>
</table>
Sample copies of the JCL for the Bridge programs are stored in the CA Endevor/DB for CA IDMS JCL installation library.

**Installation Summary**

Installation of the Bridge involves the following steps:

1. In CA Endevor/DB for CA IDMS, run the CA Endevor/DB for CA IDMS install. The Bridge modules will be installed as part of the regular installation process.

2. In CA Endevor SCM, 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 for CA IDMS Procedures**

Installing the Bridge within CA Endevor/DB for CA IDMS makes the Change Monitor available for use in a CA Endevor SCM 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 for CA IDMS Change Control Database (CCDB).

**Install the Bridge**

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

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

**Using a Standalone CCDB**

In most cases, you only modify the startup JCL and recycle the system because the CA Endevor/DB for CA IDMS installation has already constructed the CCDBs used by the Change Monitor. However, if you have a CA Endevor SCM 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 for CA IDMS installation. You will need to duplicate portions of this job to define a new CCDB environment.
CA Endevor SCM Procedures

This section discusses the steps in the CA Endevor SCM 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 CA Endevor SCM 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 CA Endevor SCM and CA Endevor/DB for CA IDMS 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 for CA IDMS 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.
3. One **NDVRC1DB TYPE = END** statement. This must be the last macro statement in the sequence.

⚠️ **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).

**Code the Source Module**

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

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITEID</td>
<td>The identifier of the site where the entities are being migrated. This site id must match the site id value in the Defaults table.</td>
<td>A 1-character alphanumeric value. (Required)</td>
</tr>
<tr>
<td>SYSCTL</td>
<td>The system control file that determines which CA IDMS/CV system the Bridge uses at run time.</td>
<td>A standard data set name. (Required)</td>
</tr>
<tr>
<td>DICTNAM</td>
<td>The identifier of the IDD/CCDB that corresponds to the named CA Endevor SCM inventory area. The database must be known to the CA IDMS/CV system named in the SYSCTL parameter.</td>
<td>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)</td>
</tr>
<tr>
<td>LOGACTN</td>
<td>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</td>
<td>Options: R -- Logging is required; (the default). O -- Logging is optional.</td>
</tr>
<tr>
<td>CCIDVAL</td>
<td>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.</td>
<td>Options: R -- Always validate the CCID (the default) O -- Validation is optional.</td>
</tr>
<tr>
<td>ENVRMNT</td>
<td>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.</td>
<td>The 1- to 8-character name of the CA Endevor SCM environment.</td>
</tr>
<tr>
<td>STAGE#</td>
<td>One or both stages (within the environment named in the ENVRMNT parameter) mapping to the dictionary named on the DICTNAM parameter.</td>
<td>Options: 1 -- Map from Stage #1 2 -- Map from Stage #2 * -- Map from both stages (the default).</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>The CA Endevor SCM MVS system (within the environment from the ENVRMNT parameter) mapping to the dictionary named on the DICTNAME parameter.</td>
<td>Options: The 1- to 8-character name of a specific system.</td>
</tr>
<tr>
<td>Parameter Description</td>
<td>Values</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>An asterisk (*) signifying all systems in the environment (the default).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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.
IDMSDB--Configuration Example

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

```
NDVRC1DB TYPE=MAIN, SITEID=3
NDVRC1DB TYPE=MAP,
    SYSCTL=IDMS.SYSCTL, DICTNAME=VENCAS,
    ENVNRNMNT=ENVIR1, STAGE#=1, SYSTEM=VENCAS
NDVRC1DB TYPE=MAP,
    SYSCTL=IDMS.SYSCTL, DICTNAME=DEVCAS,
    ENVNRNMNT=ENVIR1, STAGE#=1, SYSTEM=DEVCAS
```
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 for CA IDMS installation JCL library.

2. To activate the configuration table, IPL your system or refresh the APF-authorized loadlib that contains the module.

**Note:** 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

```plaintext
/*(Insert site-specific JOB statement)
//STEP1 EXEC PGM=your.assembler,REGION=2048K,
   // PARM='DECK,NOLOAD,NORLD,NORXREF'
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DSN=your.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=ids.sysctl,
   // DICTNAM=dictname,
   // LOGACTN=R/O,
   // CCIDVAL=R/O,
   // ENVRMNT=env-name,
   // STAGE#=1/2/*,
   // SYSTEM=sys-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.
```
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 CA Endevor SCM installation media.

Defaults Table Customization Steps

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
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 `ipfx.iqual.LOADLIB(C1DEFLTS)` where `ipfx.iqual` is the name assigned to the load library when CA Endevor SCM was installed.

4. To activate the Defaults table, IPL your system or refresh the APF-authorized loadlib that contains the module.

Refer to the *Installing section -- z/OS* if you need more detailed information on 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:

1. Modify the data sets allocated for ISPLLIB.
2. Optionally, add the SYSCTL data set.
3. 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:

```plaintext
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 for CA IDMS was installed) that contains the CA Endevor/DB for CA IDMS 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:

```plaintext
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 CA Endevor SCM creates a JCL skeleton library called iprfix.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 for CA IDMS 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.

Basic Operations

Contents

- Overview (see page 19)
- Identifiers (see page 20)
  - System Identification (see page 20)
  - CA Endevor SCM Element Names (see page 20)
    - Name Segment Requirements (see page 21)
    - Delimiters (see page 21)
- Mask Characters (see page 21)
  - Using Asterisks (see page 22)
  - Using Periods or Commas (see page 22)
- Version and Level Numbers (see page 23)
- Action Codes (see page 23)

Overview

This section provides you with information that will help you understand the coordination between the CA Endevor/DB for CA IDMS and CA Endevor SCM functions of the Bridge.
Read this section for the following information:

- Identifiers - Synchronizing CA Endevor/DB for CA IDMS and CA Endevor SCM 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 CA Endevor SCM 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 CA Endevor SCM 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 *Administrating section* for full instructions on establishing a system name.

**CA Endevor SCM Element Names**

A CA Endevor SCM 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 a CA Endevor SCM 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 a CA Endevor/DB for CA IDMS 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.

<table>
<thead>
<tr>
<th>Component</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>System name</td>
<td>8 characters</td>
</tr>
<tr>
<td>Subsystem name</td>
<td>8 characters</td>
</tr>
<tr>
<td>Element name</td>
<td>10 characters</td>
</tr>
<tr>
<td>Type</td>
<td>8 characters</td>
</tr>
<tr>
<td>Stage</td>
<td>1 digit (1 or 2)</td>
</tr>
</tbody>
</table>

**Delimiters**

When using the CA Endevor/DB for CA IDMS 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

A CA Endevor SCM 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 CA Endevor SCM element at a given time. The level number is assigned by CA Endevor SCM. There can be up to 100 levels or sets of changes within a particular version.

The Bridge translates the CA Endevor SCM version and level number information to a CA Endevor/DB for CA IDMS version number which appears in the version field of reports and change history information. For example, if a CA Endevor SCM 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 CA Endevor SCM element. (Refer to Section 5, “Change History Reports,” for a discussion of the Bridge online facilities.)

Action Codes

CA Endevor SCM uses action commands (for example, ADD, UPDATE, DELETE) to maintain the source and related files for inventory elements. Similarly, CA Endevor/DB for CA IDMS 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 CA Endevor SCM action command value is placed on change history information screens.

The following table lists the CA Endevor SCM action commands and the equivalent CA Endevor/DB for CA IDMS 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.

<table>
<thead>
<tr>
<th>CA Endevor SCM Command</th>
<th>Action Logged in CLEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>A(dd).</td>
</tr>
<tr>
<td>ARCHIVE</td>
<td>D(lete) if the entity is deleted. No action logged if the entity is not deleted.</td>
</tr>
<tr>
<td>COPY</td>
<td>No action logged.</td>
</tr>
</tbody>
</table>
Using the Bridge

Contents
- Overview (see page 25)
- Prerequisites (see page 25)
- Phase I -- Selection (see page 25)
  - Step 1 Create Selection File (see page 28)
    - Input (see page 28)
    - MIGRATE Statement Examples (see page 29)
    - Output (see page 31)
  - Step 2 Modify Selection File (see page 31)
    - Edit the Inventory Source (see page 32)
    - Edit the Existing Selection File (see page 32)
    - Rebuild the Selection File (see page 32)
  - Step 3 Correlate File with Target (see page 33)
    - Input (see page 33)
    - Output (see page 33)
  - Step 4 Modify Selection File (see page 34)
    - Edit the Inventory Source (see page 35)
    - Edit the Existing Selection File (see page 35)
  - Step 5 Finalize Selection File (see page 35)
- Phase II -- Migration (see page 35)
  - Input for the Source Job (see page 37)
  - Output from the Source Job (see page 37)
  - Input for the Target Job (see page 38)
  - Output from the Target Job (see page 38)
Overview

This section provides you with information on 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 CA Endevor SCM 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 section describes these phases, particularly as they apply to the migration of CA Endevor SCM 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.

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.
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 for CA IDMS 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 Administrating section.

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 INTERNAL TO SUBSYSTEM EXTERNAL.
   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:

<table>
<thead>
<tr>
<th>System</th>
<th>Subsystem</th>
<th>Type</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVELOP</td>
<td>9708</td>
<td>COBOL</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>9801</td>
<td>COBOL</td>
<td>2</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>System</th>
<th>Subsystem</th>
<th>Stage</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source element</td>
<td>DEVELOP</td>
<td>9708</td>
<td>1</td>
</tr>
<tr>
<td>Target element</td>
<td>QA</td>
<td>INTERNL</td>
<td>1</td>
</tr>
<tr>
<td>Source element</td>
<td>DEVELOP</td>
<td>9801</td>
<td>1</td>
</tr>
<tr>
<td>Target element</td>
<td>QA</td>
<td>9801</td>
<td>1</td>
</tr>
</tbody>
</table>

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.
To accomplish these changes, use the following statements:

```sql
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 EMPI in the DEVELOP system and to INTERNL in the TEST system.

To accomplish these changes, use the following statements:

```sql
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 INTERNAL (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 for CA IDMS 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 CA Endevor SCM 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:** CA Endevor SCM 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 for CA IDMS 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 Administrating section for more information on 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 for CA IDMS 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 for CA IDMS 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 CA Endevor SCM 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.
   
   \[ \text{ENT type entity-name vvvv.} \]
   
   Note: CA Endevor SCM inventory elements always have an entity type of C1-ELEMENT.

2. Run NDVRDSEL using the following input parameters:
   \[ \text{INPUT = FILE.} \]
   \[ \text{MODE = TRIAL.} \]

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:

\[ \text{INPUT = FILE.} \]
\[ \text{MODE = EXECUTE.} \]

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, NDVRDLDVR, builds transfer files and the CA Endevor SCM 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 for CA IDMS 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 CA Endevor SCM SET OPTIONS command that applies to Move and Transfer actions. In Release 3.8 of CA Endevor SCM, these values are as follows:

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYPASS DELETE PROCESSOR</td>
<td>FORCE</td>
</tr>
<tr>
<td>BYPASS ELEMENT DELETE</td>
<td>COMMENT 'comment-string'</td>
</tr>
<tr>
<td>BYPASS GENERATE PROCESSOR</td>
<td>OVERRIDE SIGNOUT</td>
</tr>
<tr>
<td>CCID ccid-name</td>
<td>SYNCHRONIZE</td>
</tr>
<tr>
<td>IGNORE GENERATE FAILED</td>
<td>WITH HISTORY</td>
</tr>
</tbody>
</table>

   The remaining command language syntax for the input file is detailed in the Administrating section.

   - Execute NDVRDLVR/NDVRC1.

Output from the Source Job

The migration process produces the following output for CA Endevor SCM environments.

- An **SCL file** that contains CA Endevor SCM Restore commands for each inventory element being migrated.

- If the target environment is at another site, an **SCL file** that contains CA Endevor SCM 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 CA Endevor SCM batch processor.
Input for the Target Job

Run NDVRC1 in the target environment. A sample job, SAMPC1J2, is available in the CA Endevor/DB for CA IDMS 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 for CA IDMS 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.