

CA IDMS™ Dictionary Migrator

Dictionary Migrator User Guide

Release 18.5.00, 2nd Edition



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 © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA IDMS Dictionary Migrator
- CA IDMS Dictionary Migrator Assistant
- CA IDMS/DB
- CA ADS
- CA IDMS/DC

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Documentation Changes

The following documentation updates were made for the the 18.5.00, 2nd Edition release of this documentation:

- [Sample JCL](#) (see page 221)–The location of the SAMPJCL members was corrected.

Contents

Chapter 1: Introduction	13
--------------------------------	-----------

Chapter 2: General Information	15
---------------------------------------	-----------

Introduction.....	15
Migration Steps	16
Definition of Migration.....	16
When to Use CA IDMS Dictionary Migrator	16
Migrate Systems Quickly and Accurately.....	17
Take the Guesswork Out of Migration	17
CA IDMS Dictionary Migrator Is Easy to Use	18
Flexible Processing Capabilities	19
SQL Entity Migration.....	20
Improve Development Productivity.....	20
New Applications Environment	21
Existing Applications Environment.....	22
Backup and Recovery	23
Configure the Migration to Meet Your Needs	24
Migrate Entire Systems	24
Migrate a Single Dictionary Entity.....	28
Migrate in a Multiple-Machine Environment.....	28
Extract Entities from Multiple Source Dictionaries	31
Comprehensive Reports	31
Online Aid for Selecting Parameters	32
DMA: Your CA IDMS Dictionary Migrator Connection	32
Choose Parameters for Migration.....	32
DMA Functions	33
User Responsibilities.....	33
CA IDMS Dictionary Migrator Functions	33
Use DMA to Set up a Migration Easily.....	33
Alternative to Learning CA IDMS Dictionary Migrator Parameters.....	34
Menu-Driven Parameter Selection	34
Parameter Storage	35
Online Help.....	35
When to Review the Output	35
Obtain Reports.....	37
View Syntax for Migration.....	37
View JCL for Migration	37

Chapter 3: Parameters

39

Introduction.....	39
Selecting Parameters and Limiting Migration.....	40
Selection Criteria.....	42
CA IDMS Dictionary Migrator Parameter Statements	42
PROCESS Statement.....	42
EXTRACT Statement.....	42
CHANGE Statement	43
Notation Conventions and Syntax Rules	43
PROCESS Statement.....	49
Requirements.....	49
PROCESS Parameter.....	49
RUN Parameter	49
RUN Type Activities.....	50
RUN Types for a Single Job Execution	51
RUN Types for a Two-Job Execution.....	52
SQL Entity Migration.....	53
PROCESS Statement Options.....	53
Using Parameter Options in the PROCESS Statement	54
OBJDICTIONARY Parameter.....	54
OBJNODE Parameter	55
IDOBJECT Parameter.....	55
DISCONNECT Parameter	55
NEWVERSION Parameter	56
COBOLFORMAT Parameter	57
NOXREF	57
NOPRINT Parameter	57
DICTIONARY Parameter	58
NODE Parameter	59
IDSOURCE Parameter	59
LEVEL Parameter	60
EXSOURCE Parameter.....	61
EXLOADS Parameter	63
DATE Parameter	64
DOMAIN Parameter	65
VERSION Parameter.....	65
CHANGEONLY Parameter	66
CLISTNAME Parameter	66
CLISTVERSION Parameter	66
MOVESTAMP Parameter.....	67
EXCLTAB Parameter	67

EXTRACT Statement	67
Controlling Entity Extraction	68
Statement Requirements	69
EXTRACT Parameter	72
EXTRACT Statement Entity Types	72
EXTRACT Statement Options	76
Using Parameter Options in the EXTRACT Statement	76
DICTIONARY Parameter	77
NODE Parameter	77
IDSOURCE Parameter	78
LEVEL Parameter	78
EXSOURCE Parameter	80
EXLOADS Parameter	82
DATE Parameter	83
DOMAIN Parameter	84
VERSION Parameter	84
CHANGE Statement	84

Chapter 4: Reports **87**

Introduction	87
CA IDMS Dictionary Migrator Reports	88
Parameter Verification Report	89
Source Dictionary Verification Report	90
Extract Summary and Extract Detail Reports	91
Extract Summary Report	91
Extract Detail Report	93
Comparison Reports	95
Entity Discrepancy Report	95
Entity Cross-Reference Report	96
Syntax Production Report	98
Syntax Files Display Report	99
Catalog Navigation Report	99

Chapter 5: Operations **101**

Introduction	101
Operational Considerations	101
Local Mode Processing	102
Additional Tasks Performed by CA IDMS Dictionary Migrator	102
How CA IDMS Dictionary Migrator Operates	102
Two Methods of Operation	103
Allocating Space for Work Files and Syntax Files	104

Running as a Single Job Execution Under Central Version	108
Migration Activities - Single Job Execution	108
Step 1— USMVSAM	108
Step 2--USMSYNTAX--z/OS Only	108
Step 3— USMXTRCT	108
Step 4— USMLOAD1	109
Step 5— USMLOAD3 (Optional)	109
Step 6— USMSQLOD (Optional)	109
Using CREATSYNTAX Run Type	110
Running as a Two-Job Execution	110
Migration Activities - Two-Job Execution	111
Step 1— USMSYNTAX--z/OS Only	112
Step 2— USMEXPRT	112
Step 3— Utility Copy	112
Step 4— Utility Copy	112
Step 5— USMVSAM	112
Step 6— USMIMPRT	112
Step 7— USMLOAD1	113
Step 8— USMLOAD3 (Optional)	113
Step 9— USMSQLOD (Optional)	113

Chapter 6: Dictionary Migrator Assistant 115

Introduction	115
DMA Requirements	116
Installation	116
System Requirements	116
Storage Requirements	116
User Authority	117
Types of Messages	117
Screen Messages	117
Confirm Screens	118
DMA Commands	119
Enter a Command	120
System Commands Menu	121
Display PF Key Values	122
Access the Values Screen	122
Change a PF Key Value	122
Which PF Key Values Apply?	123
Questions and Answers About Using DMA	123
What are the PF Key Values?	124
What are the System Commands?	124

What Should I Enter On a DMA Screen?	124
Can I Perform DMA Activities?.....	124
How Will A Migration Affect The Target Dictionary?	124
How Do I Use the Dictionary?.....	124
First Steps in Using DMA.....	125
Access DMA.....	125
Select DMA Activity.....	125
Reset and Exit DMA	125
Reset DMA.....	126
Exit DMA	126
Common Fields on DMA Screens	128
DMA Activity Paths	129
Define a New Migration: Sequence of Screens	129
1. Specify Overrides of DMA-Supplied Defaults	131
2. Accept DMA-Supplied Defaults.....	131
Define a New Migration: Overview.....	131
Complete First Steps Using DMA.....	131
1. Describe the Migration Environment.....	132
Establish a Setup	132
Transfer to the Next Screen	132
2. Specify a New Starting Point.....	132
Transfer to the Next Screen	133
View a List of Starting Points.....	133
3. Accept or Override Defaults	133
Transfer to the Next Screen	134
4. Specify Starting Point Overrides.....	134
Transfer to the Next Screen	134
Parameter Reference	134
5. Include/Exclude Source Types (Current Starting Point).....	135
Transfer to the Next Screen	135
6. Include/Exclude Load Types (Current Starting Point)	135
Transfer to the Next Screen	135
Parameter Reference	135
7. Obtain CA IDMS Dictionary Migrator Reports	136
Specify Migration Output	136
Transfer to the Next Screen	136
Parameter Reference	136
8. Store Setup into a PARMFILE	137
What You Can Store.....	137
How to Store	137
Transfer to the Next Screen	137
9. Continue to Use DMA	138

Global Overrides.....	138
DMA Provides a Default Starting Point.....	139
Use a Series of Commands from the Menu	140
Specify Migration Overrides.....	140
Transfer to the Next Screen	140
Parameter Reference	141
Limit Entities for Overall Migration.....	141
Include/Exclude SOURCE Types (Overall Migration).....	141
Use the Command SWAP	141
Transfer to the Next Screen	141
Parameter Reference	141
Include/Exclude LOAD Types (Overall Migration).....	142
Use the Command SWAP	142
Transfer to the Next Screen	142
Parameter Reference	142
View the Parameter Statements	142
STORE Before Submitting	143
Select a DMA Parameter File	143
Select Multiple DMA Parameter Files.....	143
Specify a DMA Parameter File	143
Modify a DMA Parameter File	144
Steps to Modify a DMA Parameter File.....	144
Modify Dictionary/Starting Point Values	144
Modify Parameters	145
Modify JCL	145
Copy a DMA Parameter File.....	147
Submit Parameters and JCL Online.....	147
File Allocations.....	148
Submit Parameters and JCL from the Batch Environment	148
Copy CA IDMS/Dictionary Migrator JCL to DMA	148

Chapter 7: Messages **151**

Introduction.....	151
Messages Generated by CA IDMS Dictionary Migrator.....	152
Severity Codes	152
Conditions.....	153

Appendix A: Sample JCL **221**

Introduction.....	221
File Assignments	221
Allocate the Syntax Files —z/OS.....	223

Allocate the VSAM Work File—z/OS.....	225
Extract Information from the Source Dictionary—z/OS.....	226
Export Syntax Files —z/OS.....	229
Import the Extract Tape--z/OS	232
Upload Syntax Files --z/OS.....	235
Delete Migrated Source from Source Dictionary—z/OS.....	241
Allocate the VSAM Work File--z/VSE	243
Extract Information from the Source Dictionary - z/VSE	244
Export Syntax Files--z/VSE	248
Import the Extract Tape—z/VSE	252
Upload Syntax Files —z/VSE.....	256
Delete Migrated Source from Source Dictionary—z/VSE	267
Allocate the VSAM Work File--z/VM.....	271
Extract Information from the Source Dictionary—z/VM	272
Export Syntax Files--z/VM.....	275
Import the Extract Tape--z/VM.....	278
Upload Syntax Files —z/VM	281
Delete Migrated Source from Source Dictionary - z/VM	292

Appendix B: Extract Paths **297**

Introduction.....	297
Extract Path for DIALOG	298
EXTRACT, LEVEL = DIALOG, DIALOG = dialog-name.....	298
Extract Path for RECORD	301
EXTRACT, LEVEL = DIALOG, RECORD = record-name	302
Extract Path for RECORD when LEVEL = ENTITY	305
EXTRACT, RECORD = record-name, LEVEL = ENTITY	306
Extract Path for ATTRIBUTE when LEVEL = DIALOG	308
EXTRACT, ATTRIBUTE = attribute-name, LEVEL = DIALOG	308
EXTRACT Path for ATTRIBUTE when LEVEL = ENTITY	311
EXTRACT, ATTRIBUTE=attribute-name, LEVEL=ENTITY	312
Extract Path for ATTRIBUTE when LEVEL = ONLY	315
EXTRACT, ATTRIBUTE = attribute-name, LEVEL = ONLY	316
Extract Path for SUBSCHEMA when LEVEL = DIALOG	317
EXTRACT, SUBSCHEMA = subschema-name, LEVEL = DIALOG	318
Extract Path for SUBSCHEMA when LEVEL = ENTITY	321
EXTRACT, SUBSCHEMA=subschema-name, LEVEL=ENTITY	322
Extract Path for APPLICATION when LEVEL = DIALOG	324

Appendix C: Customization **327**

Introduction.....	327
-------------------	-----

Installation Parameters	327
Message Severity.....	327
Customization Instructions.....	328
Entity Exclusion Table.....	330
Customization Instructions.....	330
Manipulation of the Extract File.....	332
Extract Record.....	333
Field Definitions and Values.....	334

Glossary	339
-----------------	------------

Index	341
--------------	------------

Chapter 1: Introduction

The *CA IDMS Dictionary Migrator User Guide* is a reference tool that provides information on how to use the CA IDMS Dictionary Migrator and the CA IDMS Dictionary Migrator Assistant.

Chapter 2: General Information

This section contains the following topics:

[Introduction](#) (see page 15)

[Migration Steps](#) (see page 16)

[Definition of Migration](#) (see page 16)

[When to Use CA IDMS Dictionary Migrator](#) (see page 16)

[Migrate Systems Quickly and Accurately](#) (see page 17)

[Take the Guesswork Out of Migration](#) (see page 17)

[CA IDMS Dictionary Migrator Is Easy to Use](#) (see page 18)

[Flexible Processing Capabilities](#) (see page 19)

[Improve Development Productivity](#) (see page 20)

[Configure the Migration to Meet Your Needs](#) (see page 24)

[Comprehensive Reports](#) (see page 31)

[Online Aid for Selecting Parameters](#) (see page 32)

[Choose Parameters for Migration](#) (see page 32)

[Use DMA to Set up a Migration Easily](#) (see page 33)

[Menu-Driven Parameter Selection](#) (see page 34)

[Parameter Storage](#) (see page 35)

[Online Help](#) (see page 35)

[Obtain Reports](#) (see page 37)

[View Syntax for Migration](#) (see page 37)

[View JCL for Migration](#) (see page 37)

Introduction

CA IDMS Dictionary Migrator is the CA system migration tool for transferring systems from a testing environment to a production environment. CA IDMS Dictionary Migrator Assistant (DMA) is an online tool that assists you in selecting parameters to supply to CA IDMS Dictionary Migrator. This chapter introduces CA IDMS Dictionary Migrator and DMA's features and capabilities. It also explains how CA IDMS Dictionary Migrator can be used during system development and for dictionary maintenance.

Migration Steps

The steps for performing a migration are as follows:

1. Determine what needs to be migrated.
2. Determine where the information to be migrated is located.
3. Determine where the information to be migrated is placed.
4. Select CA IDMS Dictionary Migrator parameters manually, or select them using DMA.
5. Run CA IDMS Dictionary Migrator.
6. Review CA IDMS Dictionary Migrator (depending on how you installed and ran CA IDMS Dictionary Migrator).
7. Run the upload steps.
8. Review the output of the upload.
9. Activate (complete) using the new copy Clist.

Note: The information that is migrated becomes executable when the Clist is invoked or the central version is cycled.

Definition of Migration

Migration is the transfer of a set of related entities stored in a dictionary to another dictionary. A successful migration ensures that all migrated entities function in the new environment producing results similar to those in the old environment and that no entities in the new dictionary outside of the set of migrated entities are materially affected by the migration.

When to Use CA IDMS Dictionary Migrator

Use CA IDMS Dictionary Migrator whenever you need to transfer information from one dictionary to another. This chapter will further explain performing migration.

To configure the migration you can migrate by PROGRAM and DIALOG. PROGRAM indicates that the specified program is to be extracted with its related entities. DIALOG indicates that the specified dialog is to be extracted with its related entities.

Migrate Systems Quickly and Accurately

CA IDMS Dictionary Migrator is a powerful and comprehensive tool for transferring complete systems or portions of systems from one dictionary to another.

CA IDMS Dictionary Migrator eliminates the time-consuming tasks of researching and listing every component of a system and of manually producing extensive syntax to migrate a system. CA IDMS Dictionary Migrator does the work automatically. You can move systems into production quickly, as well as accurately reflect their testing environments.

CA IDMS Dictionary Migrator is especially effective in a CA ADS environment. CA IDMS Dictionary Migrator copies from the dictionary, the source, and load modules that store all CAADS entities records, elements, files, dialogs, messages, tables, schemas, subschemas, maps, and panels during development and transfers them to the production environment. See the first graphic.

Take the Guesswork Out of Migration

CA IDMS Dictionary Migrator takes the guesswork out of migration and puts you in complete control with its easy-to-use parameters and comprehensive reports. CA IDMS Dictionary Migrator gives you a thorough review of every component in a system, compares the testing environment to the production environment, and then migrates the data.

With CA IDMS Dictionary Migrator, moving systems from testing into production, you can accurately plan resources for the final stages of development. The efficiency of CA IDMS Dictionary Migrator and the accuracy of its results let you free resources for other projects by meeting deadlines on current projects.

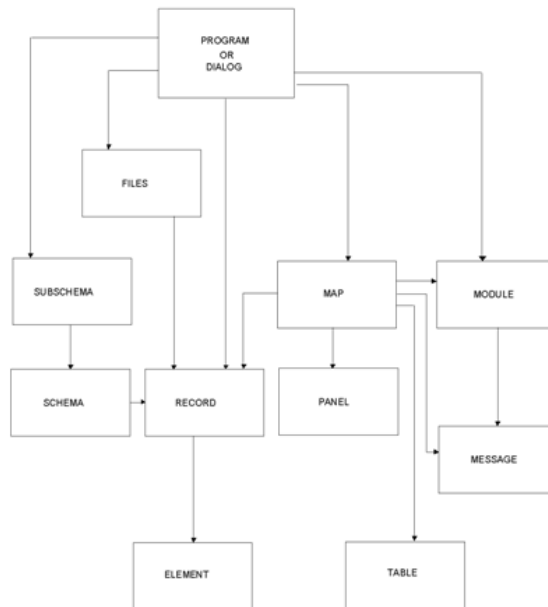
CA IDMS Dictionary Migrator also supports the transferring of dictionary entities to and from intermediate staging, or QA, or integration dictionaries.

CA IDMS Dictionary Migrator Is Easy to Use

Migrating a system requires extracting its components from source dictionaries and transferring them to an object dictionary. CAIDMS Dictionary Migrator performs the processing automatically. The second graphic illustrates CAIDMS Dictionary Migrator processes.

You can direct and control this process easily with CA IDMS Dictionary Migrator's parameters. Parameters give you the flexibility to perform simple or complex migration.

For a simple migration you need to specify the source and object dictionaries and the system that is going to be migrated. For a more complex migration, select from many optional parameters that set up criteria and tailor the process to meet your needs.



Flexible Processing Capabilities

CA IDMS Dictionary Migrator's processing capabilities are flexible. Parameters allow you to migrate:

- Entire systems by specifying organizational structure
- Single components with or without system relationships
- Source components from the DDLDDL area
- Load components from the DDLDCLOAD area
- Selected messages from the DDLDCMSG area
- Changed entities only.
- SQL catalog definitions from the DDLDCAT area

You can direct the migration process to:

- Access multiple machines and source dictionaries and/or catalogs
- Assign new or alternate version numbers to system components
- Qualify selection by version number and date criteria
- Make up to 50 changes in upload syntax
- Delete migrated entities from the source dictionary.

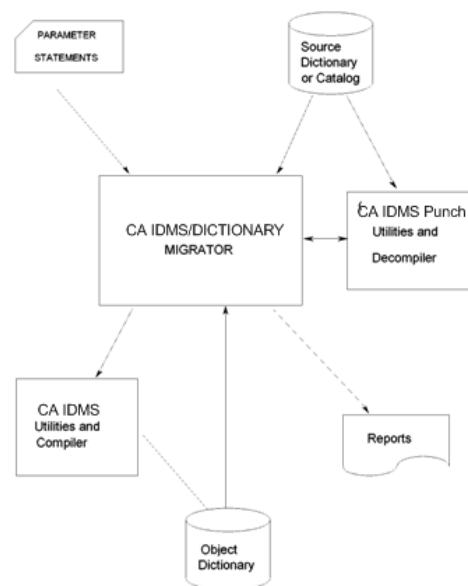
In addition, CA IDMS Dictionary Migrator allows you to review dictionary contents before and after a migration and to use the dictionary sign-on security.

Variations of the migration process are nearly unlimited with CA IDMS Dictionary Migrator. The parameters give you the freedom to configure a migration that is tailored to your system requirements and dictionary standards.

SQL Entity Migration

CA IDMS Dictionary Migrator allows you to extract definitions of logical SQL entities from a source catalog and create a syntax file containing these definitions for subsequent upload to a target catalog. Support for the following SQL entity types is included:

- Table(s)
- Views(s)
- Schema(s)
- Table Procedure(s)



Improve Development Productivity

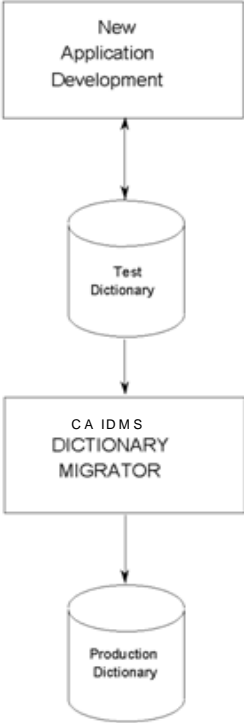
With CA IDMS Dictionary Migrator you can improve productivity in both new applications development and existing applications maintenance by moving systems quickly and accurately from dictionary to dictionary.

New Applications Environment

You can use CA IDMS Dictionary Migrator to copy a new application into production. The following graphic illustrates how easily this can be done.

Before the actual transfer to a production environment you can direct CA IDMS Dictionary Migrator to produce reports. From the reports you can review system components and relationships as they exist in the source dictionary and the syntax statements that CA IDMS Dictionary Migrator will provide for the actual migration.

When you have completed your review, you need only run the upload step of the migration you are performing. CA IDMS Dictionary Migrator generates all the syntax necessary for the migration.



Existing Applications Environment

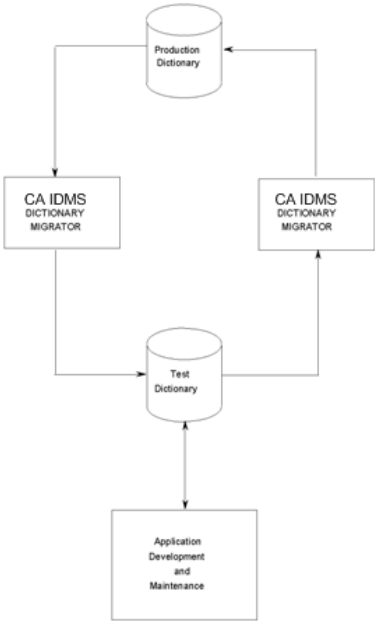
CA IDMS Dictionary Migrator is as useful in an existing applications environment as it is in the new applications environment. CA IDMS Dictionary Migrator allows you to perform testing and maintenance on a system without disturbing its production environment. The following graphic illustrates how you can use CA IDMS Dictionary Migrator for existing applications maintenance.

CA IDMS Dictionary Migrator produces a comprehensive listing of the production system and its components. From the listing you can select the necessary components--source and load modules that store dialogs, messages, tables, schemas, subschemas, maps, and panels, that are moved to the test dictionary.

The components can be assigned to a new version number to keep them distinct from current production entities. This makes it easy to test changes in entity definitions and make system modifications while keeping the production system intact.

When testing is complete, you can examine the reports to compare the system in maintenance to the original production system. Once you've noted the changes, you can migrate the changed portions back to the production version.

By selecting only the modified components, you can significantly reduce the time it takes to copy tested applications back into the production environment. CA IDMS Dictionary Migrator moves the modified system from its test dictionary to the production dictionary accurately and quickly.



Backup and Recovery

CA IDMS Dictionary Migrator can also be used to backup and recovery systems. Before moving a system from one dictionary to another you can back up the system to tape by using USMCPY. If you discover that the new version is not correct after you have migrated it, you can restore the original system.

Once a system is backed up to tape, you have an excellent way to access an individual system.

Configure the Migration to Meet Your Needs

CA IDMS Dictionary Migrator offers many options for tailoring a migration to meet your needs. To highlight a few possibilities, you can:

- Migrate an entire system by specifying its organizational structure or a related system component
- Migrate a single system component without relationships
- Migrate in a multiple machine environment
- Extract components from more than one source dictionary.

Migrate Entire Systems

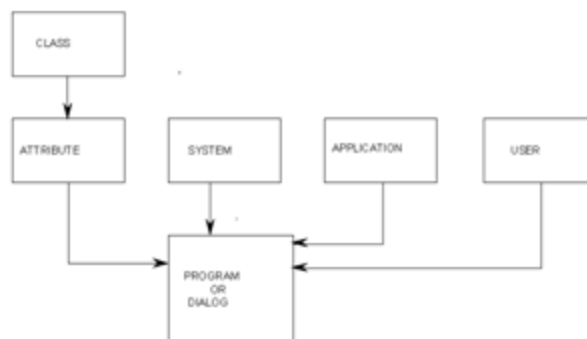
CA IDMS Dictionary Migrator provides you with methods of directing the migration of an entire system. You can direct CA IDMS Dictionary Migrator to trace either organizational structures or program/dialog connections, or migrate an entire ADSA application and all of the dialog programs and their components used in the application. Dictionary relationships are followed with either method to locate every component of the system.

Migrate by Organizational Structure

Migrating a system by its organizational structure is especially useful when you have linked all programs to an attribute, a system, a user, a program, or a dialog.

For example, if every dialog of your billing system is linked to the system structure named BILLING, you need only identify how it is organized for a migration. By specifying SYSTEM=BILLING, you can direct CA IDMS Dictionary Migrator to locate every program or dialog associated with the system. CA IDMS Dictionary Migrator will follow dictionary relationships to locate every component used by the billing system's programs or dialogs.

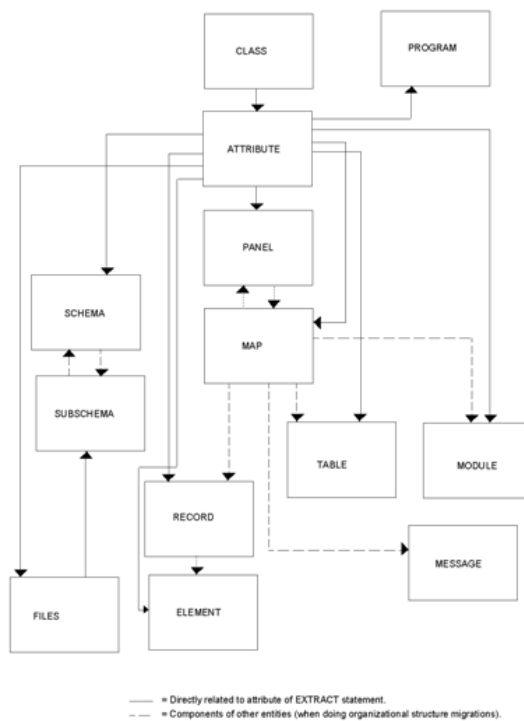
This method provides an efficient way to trace systems that are very large or that have been in development for a long time. For more information see the following flowchart.



Migrate by Organizational Structure Without Regard to Dialog

Migrating a system by its organizational structure is especially useful when you have linked an entire system to an attribute.

For example, if every component of your billing system is linked to the ATTRIBUTE structure named BILLING, you need only identify how it is organized for a migration. By specifying ATTRIBUTE=BILLING, and LEVEL=ENTITY, you can direct CA IDMS Dictionary Migrator to locate every entity occurrence associated with the system. CA IDMS Dictionary Migrator follows dictionary relationships to locate every component used by those entity occurrences. For more information see the following flowchart.



Migrate by Component Relationship

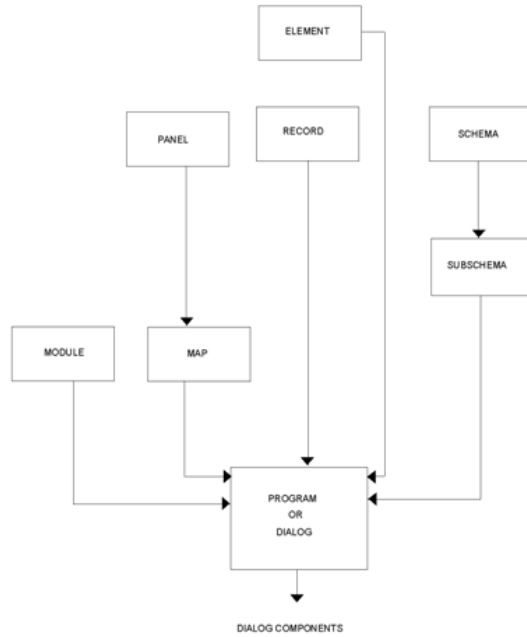
Migrating an entire system by component relationship is useful when you have not linked a system to an organizational structure and are not sure of an entire system's contents. If you specify one of the system components, CA IDMS Dictionary Migrator locates the programs or dialogs that use the component and traces dictionary relationships to extract the entire system.

For example, by specifying a schema name you can direct CA IDMS Dictionary Migrator to locate dialogs that use subschemas related to the specified schema. CA IDMS Dictionary Migrator then extracts the schema, subschemas, dialogs, and all dialog-related modules, maps, panels, tables, records, and elements.

This method can also be used when you change component definitions and need to research the effects of the change on other components within a system before making a complete migration. For more information see the following flowchart.

Migrate a Single Dictionary Entity

You can identify a single system component and direct CA IDMS Dictionary Migrator to bypass its program or dialog relationships. CA IDMS Dictionary Migrator locates the component and searches for direct relationships. For example, when you specify a single record, the record is migrated only with its elements.



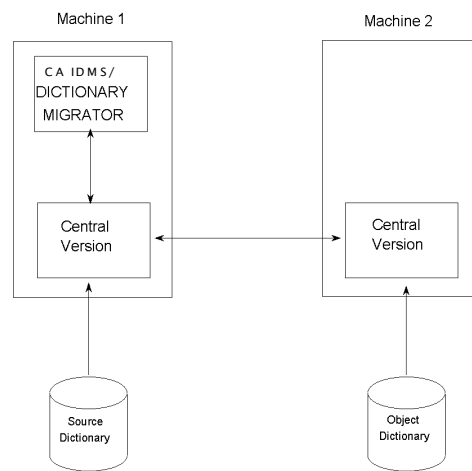
Migrate in a Multiple-Machine Environment

CA IDMS Dictionary Migrator can migrate when the object dictionary is not on the same machine as the source dictionary or Central Version (CV). To access another machine or CV you can use node communication or a two-job execution.

Single Job Execution--Using Node Communication

If you have node communication support, you can easily perform a migration in a multiple machine environment with a single-job execution of CA IDMS Dictionary Migrator. You enter parameters that allow you to identify the source and object central version nodes and the source and object dictionary.

The following graphic illustrates the single execution process.

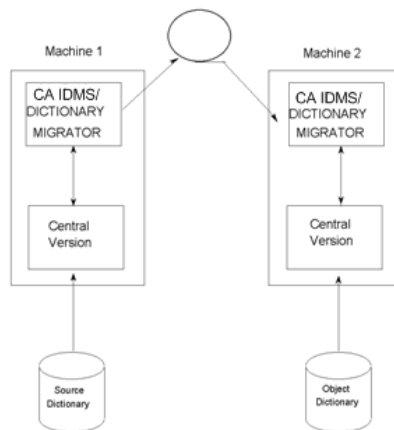


Two-Job Execution

If you do not have node communication support, you can easily process the migration in two jobs.

During the first job, the requested information from the source dictionary CPU is exported and stored on tape.

During the second job, the tape is read and source dictionary information is imported to another CPU and loaded in the object dictionary. Verification, discrepancy and cross-reference reports are also produced in this job. The following graphic illustrates the two-job process.



Advantages of the Two-Job Execution

The two-job execution allows you to communicate between different operating environments. A system developed in a z/VSE environment can be easily transferred to a z/OS environment, and vice versa.

There is an added advantage to using the two-job execution to perform a migration. The two-job execution provides an excellent method of logically backing up a system onto tape.

If you ever have to restore the system, CA IDMS Dictionary Migrator will produce verification reports about the system backed up on tape and the current system in the object dictionary before the system is actually restored.

Extract Entities from Multiple Source Dictionaries

CA IDMS Dictionary Migrator's flexible processing also allows you to extract system components from several source dictionaries in a single execution. If you are extracting from multiple source dictionaries, and node support is available, and dictionaries can be accessible on any CV, specify all dictionaries and the system component on the EXTRACT statement.

Comprehensive Reports

CA IDMS Dictionary Migrator produces nine comprehensive reports that illustrate each activity in its process:

- Parameter Verification Report
- Source Dictionary Verification Report
- Extract Summary Report
- Extract Detail Report
- Entity Discrepancy Report
- Entity Cross Reference Report
- Syntax Production Report
- Syntax Files Display Report
- Catalog Navigation Report

These reports serve as information resources before and after a migration. Each report is illustrated and described in the Reports chapter.

Before the actual migration is executed you can request reports that display the results of CA IDMS Dictionary Migrator's validity and comparative activities. These reports give informative and error messages about parameter statement formats and dictionary contents. The Extract Detail Report can be tailored to list a system and all of its relationships. This is helpful when you are moving large systems from development to production and want a thorough listing of the system.

You can use these reports to preview the contents of a migration and determine if you will be migrating the components you need. The reports can be used to give you the names and locations of additional entities that need to be migrated. You can also use the report information to verify that the system being migrated will not affect other systems.

After a migration CA IDMS Dictionary Migrator's reports are useful for analyzing dictionary contents, documenting entire systems and previous migrations, and planning future migrations. You can use the report information to make decisions about applying and maintaining dictionary standards.

The thoroughness of CA IDMS Dictionary Migrator's reports eliminates the guesswork of performing a migration and ensures that the object dictionary accurately reflects the source dictionary. You can also use CA IDMS Dictionary Migrator's reports to plan the capacity of the object dictionary and to help select the best time to perform a migration.

Online Aid for Selecting Parameters

CA IDMS Dictionary Migrator Assistant (DMA) is an online tool that assists you in selecting parameters to submit to CA IDMS Dictionary Migrator. You provide information about the source dictionary, the target (object) dictionary, and the starting point. DMA converts this information into parameter statements which you can store and submit to CA IDMS Dictionary Migrator. In addition, DMA stores dictionary and starting point values, stores JCL to run CA IDMS Dictionary Migrator, and stores any messages you insert into the JCL.

DMA: Your CA IDMS Dictionary Migrator Connection

With DMA, you can:

- Select and store parameters to submit to CA IDMS Dictionary Migrator
- Store JCL required to run CA IDMS Dictionary Migrator
- Submit parameters to CA IDMS Dictionary Migrator and JCL to run CA IDMS Dictionary Migrator.

Establish Several Migration Setups in One Session

In a DMA session, from entry to exit, you can define setups for one or more migrations.

A setup is established if you press the ENTER key on the Describe Migration Environment screen. DMA then has enough information to generate a set of parameter statements sufficient for a CA IDMS Dictionary Migrator run.

A setup is also established if you select a parameter file from the list of parmfiles.

Before submitting the parameter statements to CA IDMS Dictionary Migrator, you must store the setup.

Choose Parameters for Migration

DMA, CA IDMS Dictionary Migrator, and you each perform actions essential to the migration process. These distinct actions, described below, combine to permit quick and accurate migrations.

DMA Functions

DMA helps you choose parameters to submit to CA IDMS Dictionary Migrator. Optionally it can be used to store parameter statements and submit them to CA IDMS Dictionary Migrator. DMA can also store JCL to run CA IDMS Dictionary Migrator. When the job is submitted through DMA, the previously generated parameter statements are concatenated to the JCL.

User Responsibilities

As the DMA user, you:

- Identify the source and target dictionaries, the starting point or starting points for each migration, and the source types and load types you want to include for and exclude from each migration
- Type the source dictionary, target dictionary, and starting point values on DMA screens
- Modify the JCL to run CA IDMS Dictionary Migrator
- Select CA IDMS Dictionary Migrator output.

CA IDMS Dictionary Migrator Functions

When you use DMA to submit the CA IDMS Dictionary Migrator job, CA IDMS Dictionary Migrator will use the parameter statements generated by DMA to perform the following activities:

- Check parameter statements for errors
- Check the source dictionary for requested entities and components
- Compare the extracted entities in the source dictionary to the entities in the target (object) dictionary
- Generate syntax in the order necessary for successful uploading to CA IDMS utilities and compilers
- Display the generated syntax
- Provide reports telling you about the results of the specific migration activities.

Use DMA to Set up a Migration Easily

You avoid the necessity of learning the uses and syntax of CA IDMS Dictionary Migrator parameter statements by using DMA.

Alternative to Learning CA IDMS Dictionary Migrator Parameters

Instead of learning CA IDMS Dictionary Migrator parameters, you can simply provide DMA with the dictionary and starting point values necessary for performing a migration. The values you must enter in a DMA setup include:

- The source dictionary name
- The target (object) dictionary name
- One or more starting points for the migration.

Optionally you may specify:

- Limits and conditions on selection of the migration components
- Up to 50 changes to be made in upload syntax
- Inclusion of all source component types for migration
- Exclusion of specific source component types from migration
- Inclusion of all load module types for migration
- Exclusion of specific load module types from migration
- Your choice of output.

Menu-Driven Parameter Selection

Dictionary Migrator Assistant (DMA) is menu-driven. You can move easily through DMA, screen-by-screen, to:

- Store source dictionary, target dictionary, and starting point values into a parameter file
- Access parameter files for examination or modification
- Select output that describes migration activities, and thereby obtain up to eight CA IDMS Dictionary Migrator reports
- Submit parameters to run CA IDMS Dictionary Migrator and JCL to run CA IDMS Dictionary Migrator.

The DMA screen titles correspond to each of the DMA functions you can select.

Parameter Storage

A parameter file (PARMFILE) database is used to store:

- Parameters generated by DMA
- Dictionary and starting point values
- JCL to run CA IDMS Dictionary Migrator
- Any user-defined messages inserted in the JCL.

Storing this information in a database permits quick access to the parameters generated through DMA.

Online Help

Whenever you have a question while viewing any DMA screen, you can obtain online information about the DMA function you are using. Simply type HELP in the command line and press the ENTER key, or press PF1.

When to Review the Output

The reports produced by CA IDMS Dictionary Migrator allow you to verify that the migration will be complete and that you will be able to determine what effect the migration will have on the target dictionary.

Note: For any method, the target dictionary must be backed up before beginning the upload steps.

CA IDMS Dictionary Migrator provides three methods for reviewing the output, and completing the migration by updating the target dictionary.

Method One

The first method allows you to :

- Run CA IDMS Dictionary Migrator
- Review CA IDMS Dictionary Migrator reports
- Upload (update) the syntax files to the target dictionary (if no problems exist).

Method Two

The second method allows you to:

- Run CA IDMS Dictionary Migrator using RUN=VERIFY
- Review CA IDMS Dictionary Migrator reports
- Run CA IDMS Dictionary Migrator using CREATSYNTAX
- Upload (update) the syntax files to the target dictionary (if no problems exist).

Method Three

The third method requires advanced preparation and is usually done once during the installation. You can:

- Review the User Guide message section and determine which messages reflect critical problems in a migration and would want to start processing
- Code the USMSGSVT table (message severity table) listing all the messages that you have identified as critical
- Modify the USMTPARM module to STOP AFTER VALD ERROR (STOPVER=Y)
- Run CA IDMS Dictionary Migrator
- Upload the syntax files to the target dictionary using conditional processing on all upload steps that refer to a return code of less than 8
- Review the CA IDMS Dictionary Migrator reports (optional).

Note: Processing would have stopped if any errors occurred that you determine were critical.

The review of CA IDMS Dictionary Migrator reports is only necessary when the job does not run to completion. If all of the steps are processed, you can assume that no critical messages were written to the reports. You still would use the report to see what has been migrated. Method three is only available on systems where the return code checking is possible.

Obtain Reports

When you submit parameters to CA IDMS Dictionary Migrator through DMA, you are directing CA IDMS Dictionary Migrator to perform specific migration-related activities. Also, CA IDMS Dictionary Migrator produces reports with the results of each activity.

Depending on the run that you select from the list on the Specify Migration Output screen, you can obtain a combination of some or all of the following reports:

- Parameter Verification Report
- Source Dictionary Verification Report
- Extract Summary Report
- Extract Detail Report
- Entity Discrepancy Report
- Entity Cross-Reference Report
- Syntax Production Report
- Syntax Files Display Report
- Catalog Navigation Report

Note: For more information about the CA IDMS Dictionary Migrator reports, see the chapter "Reports."

View Syntax for Migration

To provide for easy examination of the parameters generated by DMA, DMA presents a Browse Syntax for Migration screen.

View JCL for Migration

To provide for easy examination and modification of the JCL accompanying the parameters, DMA presents an Edit JCL for Migration screen.

These screens and reports are illustrated in the chapter "Dictionary Migrator Assistant."

Chapter 3: Parameters

This section contains the following topics:

[Introduction](#) (see page 39)

[Selecting Parameters and Limiting Migration](#) (see page 40)

[CA IDMS Dictionary Migrator Parameter Statements](#) (see page 42)

[Notation Conventions and Syntax Rules](#) (see page 43)

[PROCESS Statement](#) (see page 49)

[EXTRACT Statement](#) (see page 67)

[CHANGE Statement](#) (see page 84)

Introduction

This chapter is a guide to understanding and using CA IDMS Dictionary Migrator parameters. It begins with a summary of all parameters, notation conventions, and syntax rules. The chapter then describes each parameter, shows its appropriate syntax, explains its rules for use, and describes its default. The parameters are presented in the order in which they are shown in the parameter summary. If a parameter option can be used in more than one statement, the option is presented and described in each statement.

If you need to move your dictionary entities from one dictionary to another, you should consider the following parameters and their explanation of how they relate to migration.

Item	Parameters
Parameters that describe environment. (Required)	IDSOURCE and IDOBJECT DICTIONARY and OBJDICTIONARY NODE and OBJNODE SYSIDMS and SYSCTL files are also needed to completely define the environment.
Parameters that describe the migration starting point and path. (Required)	EXTRACT entity name LEVEL

Item	Parameters
Parameters that describe output requested. (Optional)	RUN (required), NOPRINT, NOXREF, clistname, clistversion
Parameters that limit migration. (Optional)	EXLOAD EXSOURCE DISCONNECT ADDONLY CHANGEONLY DOMAIN DATE SQLONLY

Selecting Parameters and Limiting Migration

When certain conditions that pertain to migration are known, you can specify additional CA IDMS Dictionary Migrator parameters that will allow for even quicker and easier migrations.

In most cases, limiting what is migrated can help you in two ways:

- Fewer entities are examined to conserve resources.
- Entities that might cause conflicts when introduced to the target dictionary can be avoided.

The following is a table defining conditions and actions.

Condition: If you know...	Action: then use these parameters:
The schema definition has not changed.	EXSOURCE=DATABASE EXLOADS=SUBSCHEMA
Any of: 1. The message area is not used to store dialog messages. 2. The message area is shared between the source and target dictionary. 3. Variable names are used in the Display Message Code statement. 4. Display Message Code is not used in process code.	EXSOURCE=MESSAGE

Condition: If you know...	Action: then use these parameters:
All dialogs that need to be migrated are named in an extract statement.	EXSOURCE=SUBPROG
Migration method: 1. Move source and regenerate. 2. Move source and load modules (no regeneration). 3. Move load modules only.	EXLOADS=(SUBS, MAP, TABLE) Do not execute upload steps that use files TABLLOD, RHDCLD, or SUBSLOD. EXSOURCE=(ALL)
User registration is not used, or different users are registered in the target dictionary.	DISCONNECT=USER
Attributes in the source dictionary are not relevant to the target dictionary.	DISCONNECT=ATTRIBUTE
(virtually always)	DISCONNECT=SYNONYM
If the source and target dictionaries are in different Central Versions that are not accessible to each other through node communication.	RUN=EXPORT For accessing the target dictionary.
If you want to migrate the named entity with all of its related entities:	LEVEL=DIALOG or PROGRAM
If you want to limit the migration to the named entity and its subcomponents:	LEVEL=ENTITY
If you want to limit the migration to the named entity without any other entities:	LEVEL=ONLY
If you want to review the list of extracted entities and reports prior to creation of the syntax for the target dictionary population execute with: Assuming these meet with your satisfaction then execute with:	RUN=VERIFY RUN=CREATESYNTAX

Selection Criteria

Use these parameters to limit which entity occurrences are migrated:

- **CHANGEONLY**—Checks the target and source dates last updated. If the source is newer than the target entity, it will be migrated. Limitation updates to the target dictionary and can only be done by migration from a single source dictionary. You cannot migrate to a dictionary and directly update to the same dictionary.
- **DOMAIN=COMPONENT, DATE=(start-date, end-date)**—date range defines when changes occurred in the source dictionary and everything in the extract path that changed during that time range will be migrated.
- **ADDONLY**—Moves only what is new to the target dictionary.
- **SQLONLY**—Specifies that only those activities associated with the extraction of SQL entities will be carried out. This is valid only for Runtype='EXPORT'. The only reports that are created are Parameter Verification and Catalog Navigation reports.

CA IDMS Dictionary Migrator Parameter Statements

CA IDMS Dictionary Migrator has three parameter statements:

- PROCESS
- EXTRACT
- CHANGE.

There are many optional parameters available for each statement. The next three graphics display the requirements and options for the three statements.

PROCESS Statement

To initiate processing, you must enter one PROCESS statement for each execution of CA IDMS Dictionary Migrator. You must specify which type of migration run you want to use, and additional global limitations.

EXTRACT Statement

The EXTRACT statement identifies a source dictionary entity as the starting point of the migration. In a single run of CA IDMS Dictionary Migrator you can enter as many EXTRACT statements as you need, but each statement can identify only one entity type and name, or mask.

CHANGE Statement

The CHANGE statement is an optional statement which allows you to alter the contents of the upload syntax files.

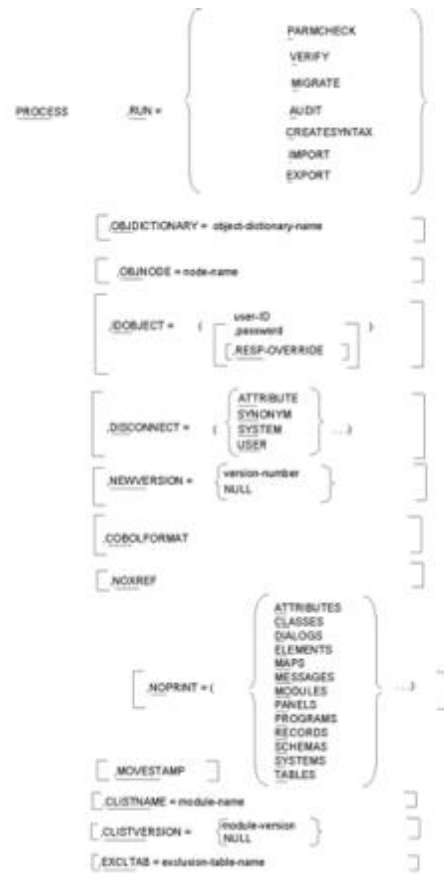
Notation Conventions and Syntax Rules

Be sure to review the following table for CA notation conventions and the next table for parameter syntax rules.

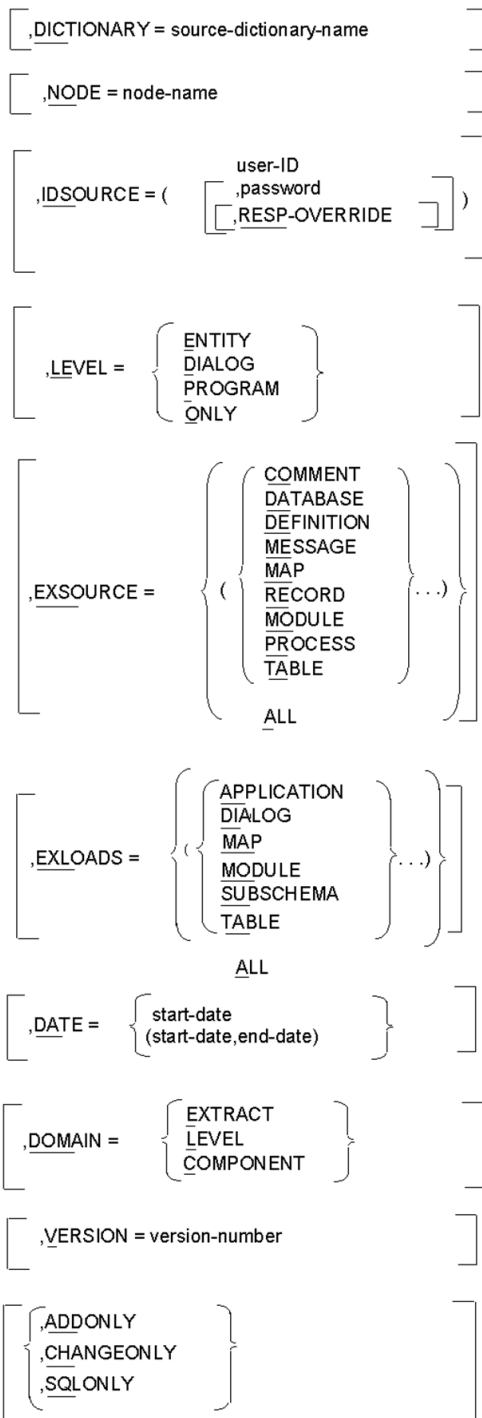
Example	Rule
PROCESS	Keywords appear in UPPERCASE. The required portion of each keyword is underlined. You can omit the portion of a keyword that is not underlined without altering the meaning of the statement.
DIALOG=dialog-name	Variables appear in lowercase. You must substitute an appropriate value for each variable.
// APPLICATION \\ DIALOG (< < MAP > > ...) SUBSCHEMA \\ TABLE //	An ellipsis (...) indicates that one or more options that are separated by commas can be selected and entered.
< version > \ NULL /	Braces enclose two or more options. You must select only one of the options.
[,DATE=start-date]	Square brackets indicate optional clauses.

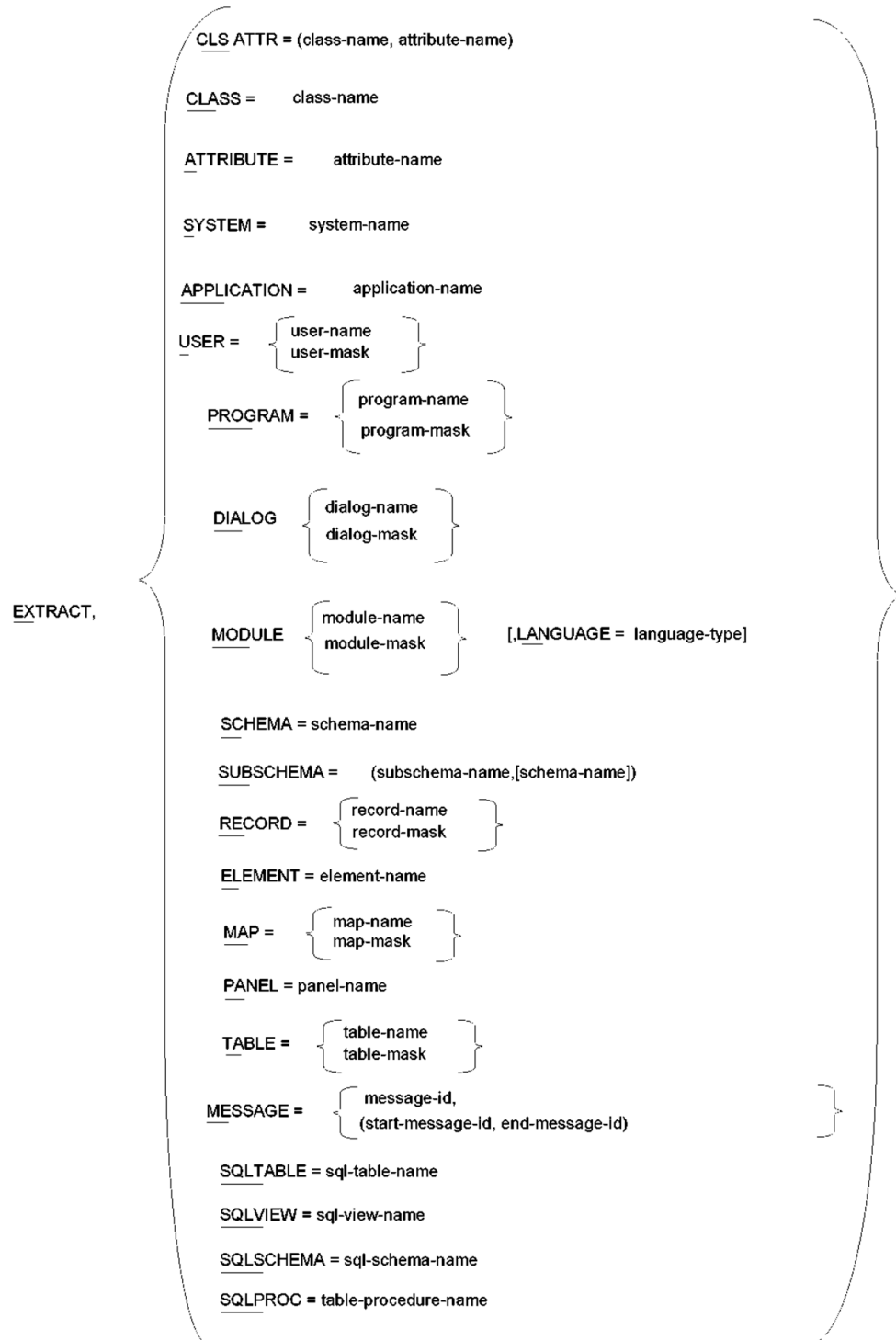
Item	Rule
Order of Parameter Statements	Enter the PROCESS statement first. You can enter the other parameter statements in any order.
Required Parameter Statements	You must enter all parameters included in this chapter, except the Environment parameter.
Continuing a Parameter Statement	To continue a parameter statement onto the next record, key in a trailing comma. Do not split a keyword between two records.

Item	Rule
Entering Blanks in Parameter Statements	You can enter blanks (character spaces) to separate keywords and improve readability in a parameter statement without affecting processing. When you include blanks in a value field, you must enclose the entire field in single quotes.
Entering Quotes in Value Fields	You may need to enter quotes within a value field that is delimited by quotes. To do this, enter two single quotes wherever a quote is needed within the value field.
Entering Parameter Statements	On an 80-character input record, enter all parameter syntax between columns 1 and 72 (inclusive).
Comments	Enter an asterisk (*) in column 1 to indicate a comment.
Synonyms	The PROGRAM and DIALOG options can be interchanged in parameter statements. Use the option to which you are most accustomed.
Series of Variables	When entering values for a series of variables, separate the values with commas and enclose the entire series with parentheses.

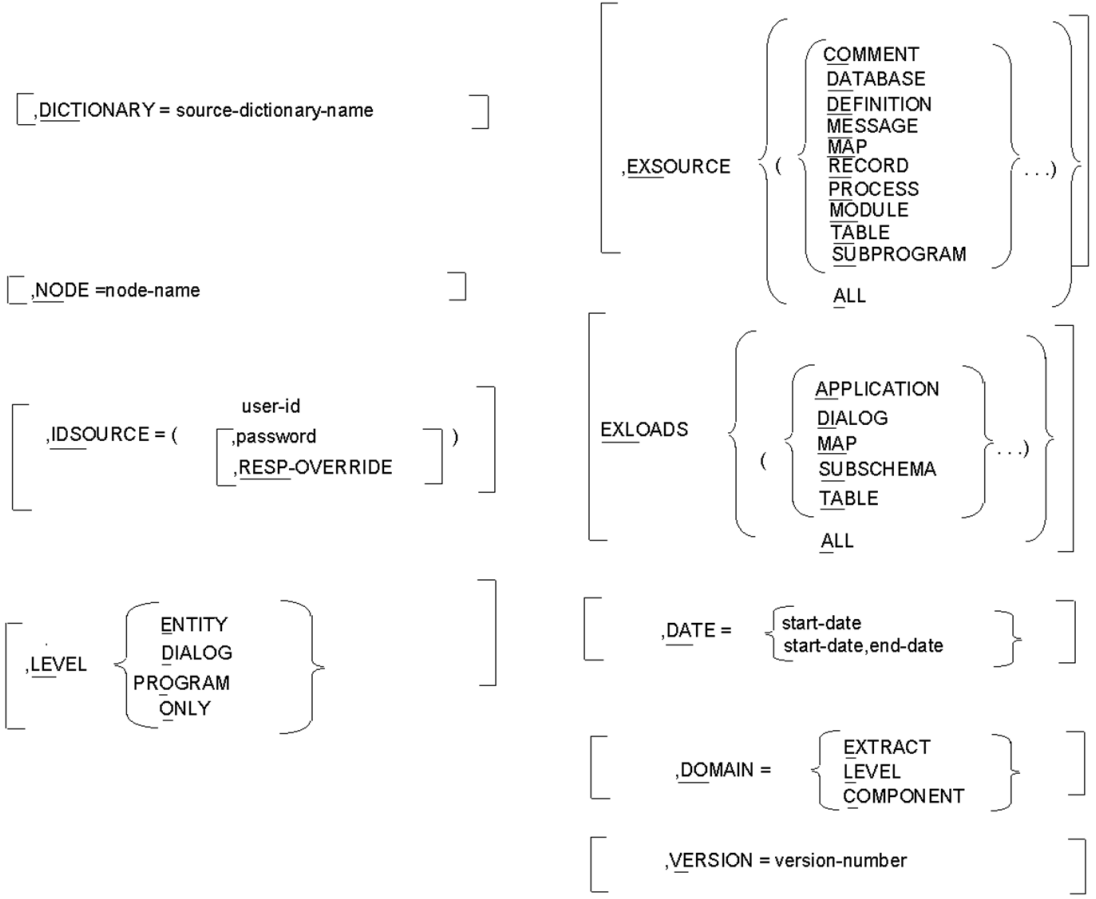


Process continued...





EXTRACT continued...



PROCESS Statement

A single PROCESS statement is required for each run of CA IDMS Dictionary Migrator. The PROCESS statement must be the first statement entered. It must also specify the type of run to be performed as well as identify the CA IDMS environments of the source and object dictionaries.

Requirements

The two required parameters of the PROCESS statement are:

- PROCESS
- RUN.

In addition to PROCESS statement requirements, you can also include parameter options that globally control processing. The first of the previous four parameter summaries displays parameter requirements and options for the PROCESS statement.

PROCESS Parameter

The *PROCESS* parameter initiates execution of CA IDMS Dictionary Migrator and indicates that processing requirements and options follow.

RUN Parameter

```

      / PARMCHECK \
      | VERIFY   |
, RUN < MIGRATE >
      | AUDIT    |
      | EXPORT   |
      \ IMPRT    /

```

The RUN parameter is a required parameter that controls CA IDMS Dictionary Migrator processing and separates its activities. Each run type directs CA IDMS Dictionary Migrator to perform specific activities, stop processing, and produce reports with the results of each activity. The selection of run types allows you to control processing for both single job and two-job executions.

RUN Type Activities

You can specify run types to perform these activities:

- Check parameter statements
- Verify use of source dictionary and retain extracted entities
- Compare extracted entities to object dictionary
- Generate syntax for CA IDMS utilities
- Display syntax for CA IDMS utilities.

The following report shows the output from each run type. The functions and activities of each run type are also presented on the following pages.

Run = Report	Parmcheck	Verify	Migrate	Audit	Export	Import	Create Syntax
Parameter Verification	X	X	X	X	X	X	X
Source Dictionary Verification		X	X	X	X		
Extract Summary		X	X	X		X	
Extract Detail		X	X	X		X	
Entity Discrepancy		X	X	X		X	
Entity Cross Reference		X	X	X		X	
Syntax Production Report			X	X		X	X
Syntax Files Display				X			X
Syntax Files			X	X		X	X
Catalog Navigation Report					X		

RUN Types for a Single Job Execution

CA IDMS Dictionary Migrator provides five run types to be used when you are performing a migration that transfers entities from one dictionary to another within the same Central Version (CV). These run types are also used when you are performing a migration between two CVs that share node communication. Node communication is discussed in the *CA IDMS System Operations Guide*. The first two run types, PARMCHECK and VERIFY, ensure that the migration is performed correctly. The actual upload to the object dictionary is performed after either a MIGRATE or an AUDIT run, or a CREATESYNTAX run providing a verify run that has been executed previously. The Operations Section contains additional details, a system flow, and a description of migration in a single job execution of CA IDMS Dictionary Migrator.

,RUN=

Indicates that a run type follows.

PARMCHECK

Indicates that CA IDMS Dictionary Migrator is to check parameter statements for errors and produce a report of the results.

VERIFY

Indicates that CA IDMS Dictionary Migrator is to check parameter statements for errors, check the source dictionary for requested entities, check extract and sort entities, compare them to the object dictionary, and produce reports of the results.

MIGRATE

Indicates that CA IDMS Dictionary Migrator is to perform parameter checking, verification activities, and produce reports of the results. MIGRATE also directs CA IDMS Dictionary Migrator to generate syntax for CA IDMS utilities. The syntax is *not* displayed on reports produced from this run type.

AUDIT

Indicates that CA IDMS Dictionary Migrator is to perform parameter checking and verification activities as well as produce detailed reports of the results. AUDIT also directs CA IDMS Dictionary Migrator to generate syntax for CA IDMS utilities that populate the object dictionary. The syntax is displayed on the Syntax Files Display Report.

CREATE SYNTAX

Indicates that the extract file created in a verify run is used to generate syntax for the CA IDMS utilities that populate the object dictionary. Create syntax is used after running the verify run and not finding any errors that require correction.

Rule: An EXTRACT statement is not allowed when RUN=CREATESYNTAX.

RUN Types for a Two-Job Execution

Two options of the RUN parameter--EXPORT and IMPORT--allow you to perform a migration between dictionaries that exist on different Central Versions (CVs) without sharing node communication. This type of migration requires at least two runs of CA IDMS Dictionary Migrator. The chapter "Operations" illustrates the system flow for the two-job execution and contains additional details and model JCL to perform EXPORT/IMPORT runs.

The previous report lists output from the EXPORT/IMPORT run types.

,RUN=

Indicates that a run type follows.

EXPORT

Indicates that CA IDMS Dictionary Migrator is to perform only the activities that access the source dictionary.

EXPORT directs CA IDMS Dictionary Migrator to create an extract file from the source dictionary and to generate syntax files for CA IDMS utilities. The extract file is used for entity comparison in the IMPORT run. The syntax files are used by the CA IDMS utilities to populate the object dictionary.

IMPORT

Indicates that CA IDMS Dictionary Migrator is to perform only the activities that compare the extracted entities to the object dictionary and produce reports with the results.

Note: No comparison activities are carried out for extracted SQL entities from the source catalog.

Rule: An EXTRACT statement is not allowed when RUN=IMPORT.

SQL Entity Migration

Extraction of SQL entity definitions can be performed as a stand-alone run by specifying the SQLONLY parameter on the PROCESS statement. You can also perform SQL entity extraction in conjunction with regular network extractions.

A separate file, ddname=BCFUPD, containing SQL CREATE syntax will be created. The primary use of this enhancement is for transferring SQL definitions from a source catalog to a target catalog where these definitions do not already exist.

The SQL syntax file (BCFUPD) is used by the IDMSBCF facility to populate a target catalog.

PROCESS Statement Options

The parameter options available in the PROCESS statement provide two functions:

- Tailoring the output to the object dictionary and reports
- Tailoring the selection of entities from the source dictionary.

The graphic PROCESS Statement Parameter Summary shows the parameter options and requirements available in the PROCESS statement.

By including options that tailor both the output to the object dictionary and reports, you can direct CA IDMS Dictionary Migrator to:

- Transfer entities to alternate dictionaries, versions, and machines in a single execution
- Transfer entities without related systems and attributes, users, or synonyms
- Limit the amount of information on reports
- Apply the standard dictionary sign-on security
- Generate COBOL-like syntax instead of DDDL syntax for records.

The options used for tailoring the selection of information from the source dictionary direct CA IDMS Dictionary Migrator to:

- Access an alternate source dictionary
- Extract entities on a downward path, bypassing dialog and program relationships
- Exclude specific source and load component types
- Limit the entry point of entity extraction to specific date and version number
- Specify a version number for the starting point

- Limit starting point to those updated during a specified and/or time period
- Limit migration of entity occurrences to those updated during a specified time period
- Specify a table of entity occurrences which should not be migrated.

Using Parameter Options in the PROCESS Statement

When the parameter options are specified in the PROCESS statement they will provide global control. If component types are excluded in the PROCESS statement, you cannot include them in the EXTRACT statement. If you have individual parameter requirements, you should indicate them in the EXTRACT statement. Some parameters can be specified in both the PROCESS and EXTRACT statements.

For example, if you specify :

EXSOURCE = MESSAGE

on the PROCESS statement and:

EXSOURCE = RECORD

on the EXTRACT statement, both MESSAGES and RECORDS would be excluded.

If the same parameters are used in both the EXTRACT and PROCESS statements, the values are combined for EXSOURCE, EXLOAD, and DOMAIN. For options with a single value the EXTRACT statement value, if any, will override the PROCESS statement.

OBJDICTIONARY Parameter

,OBJDICTIONARY=dictionary-name

is an optional parameter that identifies the object dictionary that will be populated with the extracted entities. This parameter is useful when a central version or local mode can access more than one dictionary and you do not want the default dictionary to be populated.

Default: The default is the default dictionary of the central version or the dictionary specified in the JCL.

OBJNODE Parameter

,OBJNODE=node-name

Is an optional parameter that identifies a node as the communication link between central versions and CPUs. By identifying the node you can populate an object dictionary that exists in a CV other than the one in which CA IDMS Dictionary Migrator is processing. You must specify a node name when running a single job where the object dictionary is in a CV other than the one pointed to by the SYSCTL file.

IDOBJECT Parameter

,IDOBJECT=(user-id, [password, [RESP-OVERRIDE]])

Is an optional parameter that provides standard sign-on security to control access to the object dictionary. ID object must be specified if the dictionary signon security is on.

user-id

Indicates the standard identification of the user.

password

Indicates a valid password that identifies the user of the object dictionary.

Default: There is no default value. If the dictionary sign-on security is required, you must enter a valid-ID and password.

RESP-OVERRIDE

Releases dictionary security by generating a REGISTRATION OVERRIDE in the IDD SET OPTIONS statement. This option requires a user-ID that has full authorization.

DISCONNECT Parameter

```

,DISCONNECT = ( / ATTRIBUTE \
                < SYNONYM > ... )
                | SYSTEM |
                \ USER /

```

DISCONNECT is an optional parameter that prevents the transfer of attributes, systems, and users connected to entities in the source dictionary and prevents syntax from being formulated with synonym specification. The source dictionary retains the original attribute and system connections. You can use any of the options of this parameter in the same run of CA IDMS Dictionary Migrator.

ATTRIBUTE

Indicates that attributes connected to entities are not moved when the entities are transferred to the object dictionary. If **ATTRIBUTE** is selected, class associations connected to entities also are not moved when the entities are transferred to the object dictionary.

SYNONYM

Indicates that all syntax is formulated *without* synonym specification. If the root-element name cannot be located, an extended extraction search is attempted against the **ELEMENT SYNONYM** structure. If **SYNONYM** is not selected, this feature is disabled.

SYSTEM

Indicates that systems connected to entities are not moved when the entities are transferred to the object dictionary.

USER

Indicates that user security structures are not moved when the entities are transferred to the object dictionary.

Default: All system, attribute, synonym, and user associations in the source dictionary will be transferred to the object dictionary.

NEWVERSION Parameter

```
          / version-number \  
,NEWVERSION = < NULL      >
```

NEWVERSION is an optional parameter that allows you to assign a different version number to entities as they are transferred from the source dictionary to the object dictionary. The version numbers of records, subschemas, and maps embedded in load modules are also changed.

version-number

The new version number keeps the entities distinct from the production environment. For example, you can transfer entities from production environment dictionaries to a test dictionary, assign a new version number to the entities, and use them for testing. The version numbers of the entities as they exist in the production environment will not be changed.

Rule: The version number must be an integer from 0 through 9999. If a **NULL** is used, the dictionary must have a default of an absolute number instead of a relative value.

NULL

Indicates that all version qualifications are removed from the derived syntax in order to facilitate dictionary-level default version specification.

Default: The default value is the version number of each entity in the source dictionary.

COBOLFORMAT Parameter

,COBOLFORMAT

Is an optional parameter that directs CA IDMS Dictionary Migrator to generate COBOL-like syntax, instead of DDDL syntax, to transfer records with elements.

NOXREF

NOXREF is an optional parameter that allows you to prevent the creation of the Entity Cross Reference Report.

Default: The Entity Cross Reference Report is produced.

NOPRINT Parameter

,NOPRINT=(entity-type ...)

NOPRINT is an optional parameter that allows you to prevent specific entities from appearing on the Extract Detail Report and the Entity Cross-Reference Report.

,NOPRINT=

indicates that one or more entity types follow.

entity-type

Indicates that extracted entities of the specified type are not to appear on the reports. You can enter one or more of these entity types for the NOPRINT parameter:

- ATTRIBUTES
- CLASSES
- DIALOGS
- ELEMENTS
- MAPS
- MESSAGES
- MODULES
- PANELS
- PROGRAMS
- RECORDS
- SCHEMAS
- SUBSCHEMAS
- SYSTEMS
- TABLES

Default: Every extracted entity appears on the Extract Detail Report and the Entity Cross-Reference Report.

DICTIONARY Parameter

*,DICTIONARY=***dictionary-name** or **source-catalog-name**

Is an optional parameter that allows you to access an alternate source dictionary instead of the default dictionary as specified in the central version. Identifying an alternate source dictionary in the PROCESS statement is useful when more than one dictionary is accessible through the central version in which CA IDMS Dictionary Migrator is processing. For SQL entities, this parameter allows you to access a catalog other than the default system catalog.

Default: The default dictionary as specified in the central version. For SQL entities, the default is the system catalog.

NODE Parameter

,NODE=node-name

Is an optional parameter that identifies a node as the communication link between central versions and CPUs. By identifying the node you can access a central version that exists in a CPU or region other than the one in which CA IDMS Dictionary Migrator is processing.

Default: You access the dictionaries that exist in the central version in which CA IDMS Dictionary Migrator is processing (as defined by the sysctl file).

IDSOURCE Parameter

,IDSOURCE=(user-id , [password[,RESP-OVERRIDE]])

Is an optional parameter that provides standard sign-on security to control access to the source dictionary.

,IDSOURCE=

Indicates that valid user identification follows.

user-id

Indicates the standard identification of the user in the source dictionary.

password

Indicates a valid password that identifies the user of the source dictionary.

Default: There is no default value. If the dictionary sign-on security is required, you must enter a valid-ID and password.

RESP-OVERRIDE

Allows access to entity occurrences that are registered to a user other than the one named on IDSOURCE

Default: This option requires a user-ID that has full authorization, there is no default value.

LEVEL Parameter

LEVEL is an optional parameter that allows you to specify the path that CA IDMS Dictionary Migrator follows to extract an entity from the source dictionary.

CA IDMS Dictionary Migrator can extract an entity by using one of three methods. First, CA IDMS Dictionary Migrator can locate the programs or dialogs that use the entity specified in the EXTRACT statement and extract all existing relationships. Second, CA IDMS Dictionary Migrator can bypass program or dialog relationships for a specific entity so that only the entity specified in the EXTRACT statement and its lower-level entities will be extracted. Finally, only the entity in the EXTRACT statement will be extracted.

You can direct CA IDMS Dictionary Migrator to bypass program or dialog relationships in EXTRACT statements that specify these entity types:

- Schema
- Subschema
- Record
- Module
- Element
- Map
- Panel
- Table
- Class
- Attribute
- Class/Attribute
- User

For example, if you specify a record in an EXTRACT statement and include LEVEL=ENTITY, CA IDMS Dictionary Migrator extracts only the specified record and the elements connected to the record.

```
      / DIALOG \
,LEVEL = < PROGRAM >
      | ENTITY |
      \ ONLY /

,LEVEL=
```

Indicates that a starting point option follows. The DIALOG and PROGRAM options indicate the same level.

DIALOG

CA IDMS Dictionary Migrator traces an entity specified in the EXTRACT statement to its dialog level. All the dialogs, programs, and the components of the dialogs and programs where the entity participates are extracted.

PROGRAM

CA IDMS Dictionary Migrator traces an entity specified in the EXTRACT statement to its program level. All the programs and the components of the programs in which the entity participates are then extracted.

ENTITY

Indicates that an entity is to be extracted with its lower-level relationships only. This option applies to schemas, subschemas, records, elements, modules, maps, panels, tables, classes, and attributes.

ONLY

Indicates that the entity is to be extracted by itself.

Default: All entities except messages are extracted with their dialog or program relationships. Messages are extracted with any lower-level relationships.

EXSOURCE Parameter

```

      / / COMMENT \ \
      | | DATABASE | |
      | | DEFINITION | |
      | | MESSAGE | |
      | | MAP | |
,EXSOURCE = < (< RECORD >...)>
      | | MODULE | |
      | | PROCESS | |
      | | TABLE | |
      | \ SUBPROGRAM / |
      \ ALL /

```

The EXSOURCE parameter (exclude source) is an optional parameter that prevents CA IDMS Dictionary Migrator from extracting specific source components. You can specify as many as nine component types to be bypassed in a single execution.

,EXSOURCE=

indicates source component types follow.

COMMENT

Indicates that all comments associated with entities are not to be extracted.

DATABASE

Indicates that all schema-related entities that exist in the source dictionary are not to be extracted. These include all schemas, schema records and elements, subschemas, and subschema records and elements.

DEFINITION

Indicates that definitions associated with entities are not to be extracted.

MESSAGE

Indicates messages are referenced by maps and dialog module source are not to be extracted.

MAP

Indicates that all map-related entities are not to be extracted. These include maps and panels, tables and messages used by the map, and records and elements owned by the map.

RECORD

Indicates that all records and elements are not to be extracted.

PROCESS

Indicates that a process code is not to be extracted.

MODULE

Indicates that the modules (other than process code) are not to be extracted.

TABLE

Indicates that table source code is not to be extracted from the source dictionary.

SUBPROGRAM

Indicates that programs or dialogs referenced by dialogs found from the EXTRACT statement will not be extracted.

ALL

Indicates that all EXSOURCE options are not to be extracted from the source dictionary.

Default: All source component types are extracted.

EXLOADS Parameter

EXLOADS (exclude loads) is an optional parameter that limits the load modules that CA IDMS Dictionary Migrator extracts from the DDLDCLOD (load) area of the source dictionary. You can enter up to four load module types in the same run.

```

      / / APPLICATION \ \
      | | DIALOG      | |
      | | MAP         | |
,EXLOADS = < (< MODULE >...)>
      | | SUBSCHEMA  | |
      | \ TABLE     / |
      \ ALL          /

```

,EXLOADS=

Indicates that a load module type follows.

APPLICATION

Indicates that load application modules are not extracted from source dictionary. The ADSA application cannot be updated unless the load module is migrated.

DIALOG

Indicates that dialog load modules are not to be extracted from the source dictionary. Consequently, you cannot regenerate the dialog if it does not already exist in the object dictionary.

MAP

Indicates that map load modules are not to be extracted from the source dictionary. Map load modules should not be extracted if map source is being extracted and the map will be regenerated.

MODULE

Indicates that map help load modules are not to be extracted from the source dictionary. Map help load modules should not be extracted if map source is being extracted and the map will be generated.

SUBSCHEMA

Indicates that subschema load modules are not to be extracted from the source dictionary. Subschema load modules should not be extracted if subschema source is being extracted and the subschema will be regenerated or if no database components are to be migrated.

TABLE

Indicates that table load modules are not to be extracted from the source dictionary. Table load modules should not be extracted if table source is being extracted and the table will be regenerated.

ALL

Indicates that no load module types are to be extracted.

Default: All load modules are extracted.

DATE Parameter

,DATE=start-date

**,DATE = < start-date >
 \ (start-date,end-date) /**

Is an optional parameter that allows you to specify a time period in which an entity was updated. The DOMAIN parameter determines which entity occurrences will be tested.

,DATE=

Indicates that a date or date range follows.

start-date

Indicates the date, specified as mmddyy, which CA IDMS Dictionary Migrator to search for in the dictionary. Once the entity with the indicated date or a more recent date is located, the entities which will be extracted are determined by the domain parameter.

start-date,end-date

Indicates a date range with a starting date and ending date, specified as (mmddyy,mmddyy), which CA IDMS Dictionary Migrator is to search for in the source dictionary.

Default: All entities are extracted without regard to date updated.

DOMAIN Parameter

```

      / EXTRACT    \
DOMAIN = < LEVEL  >
      \ COMPONENT /

```

DOMAIN specifies where the date selection criteria will be applied.

EXTRACT

Indicates that the date range in the date parameter is tested on each entity named in an Extract Statement. If a test fails, the entity is not used as the starting point for extraction.

LEVEL

Indicates that the date range in the date parameter is tested on each entity serving as the top level of an extract path. This is dependent on the Level parameter. If the test fails neither the entity nor any of its components will be migrated.

COMPONENT

Indicates that the data range in the date parameter is tested on each entity occurrence found in the extract path. If the test fails the entity is not extracted.

Default: Date range is tested on each entity named in an EXTRACT statement.

VERSION Parameter

```
, VERSION=version-number
```

Is an optional parameter that allows you to locate an entity occurrence based on a specific version number. This parameter is used when an entity has multiple versions in the source dictionary. Entities often have more than one version when they are being used for application development.

The specified version number applies only to the entry point. You can specify values 0-9999.

Default: The default version number is 1.

CHANGEONLY Parameter

CHANGEONLY

CHANGEONLY is an optional parameter that allows you to limit migration to changed entities.

ADDONLY

Is an optional parameter which allows you to limit migration to entities that are new to the target dictionary.

SQLONLY

Is an optional parameter which you use to limit the migration to SQL entities only. Only Parameter Verification and Catalog Navigation reports are produced for *SQLONLY* runs.

CLISTNAME Parameter

CLISTNAME=module-name

The *CLISTNAME* parameter is an optional parameter which allows you to assign a 32-character module name for the module used to contain the CLIST for new-copying, migration-related load modules.

module-name

Indicates the module name.

Default: If *CLISTNAME* is not specified, a module name based on the date and time will be specified. The default module name will be *DBMSYYDDDDHHMMSSTT*. *YYDDD* indicates the current Julian date. *HHMMSSTT* indicates the current time to thousandths of a second

CLISTVERSION Parameter

```
                / module-version \  
CLISTVERSION = < NULL           >
```

The *CLISTVERSION* parameter is an optional parameter used in conjunction with the *CLISTNAME* parameter to specify a version number for the module specified in the *CLISTNAME* parameter.

module-version

Indicates the module version number.

NULL

Indicates that the dictionary-level default version will be used.

Default: If no CLIST version is specified, the default version is 1.

MOVESTAMP Parameter

MOVESTAMP

MOVESTAMP is an optional parameter which indicates that the map is to be decompiled with the DATETIME=YES parameter specified. Specifying MOVESTAMP ensures that the generation of the map in the object dictionary will leave the date/time stamp from the source dictionary unchanged.

EXCLTAB Parameter

,EXCLTAB = exclusion-table-name

EXCLTAB is an optional parameter which allows you to reference an entity exclusion table other than the default.

exclusion-table-name

Is a user defined list of entity occurrences which will always be excluded from migration to avoid needless migration or complications arising from changing particular entities. When you specify alternate tables, you can alter the exclusion list as needed for special migrations.

Default: The exclusion table USMEXCLT will be used.

EXTRACT Statement

The EXTRACT statement identifies a source dictionary entity type and name as the starting point of the migration. This statement is required to access the source dictionary every time CA IDMS Dictionary Migrator is run except RUN = IMPORT, or RUN = CREATSYNTAX in a two-job execution.

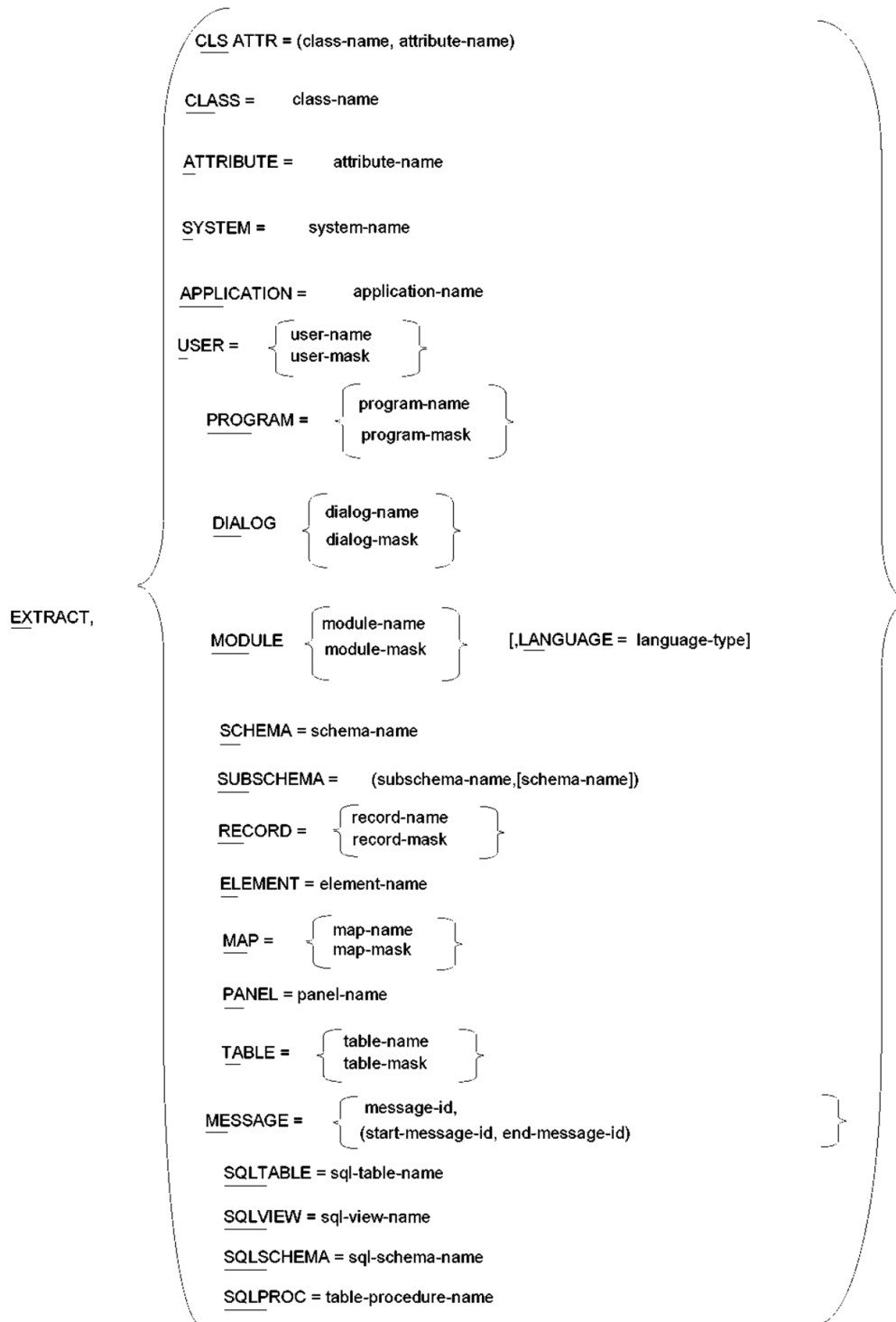
Controlling Entity Extraction

CA IDMS Dictionary Migrator normally locates a dialog or program and extracts all of its components and their relationships. By including the `LEVEL=ENTITY` parameter you can bypass program and dialog relationships and extract specific entity types with their lower-level components only. See the `LEVEL` Parameter section in this chapter for details. By including `LEVEL=ONLY` parameter you can bypass all relationship and extract specific entities by themselves.

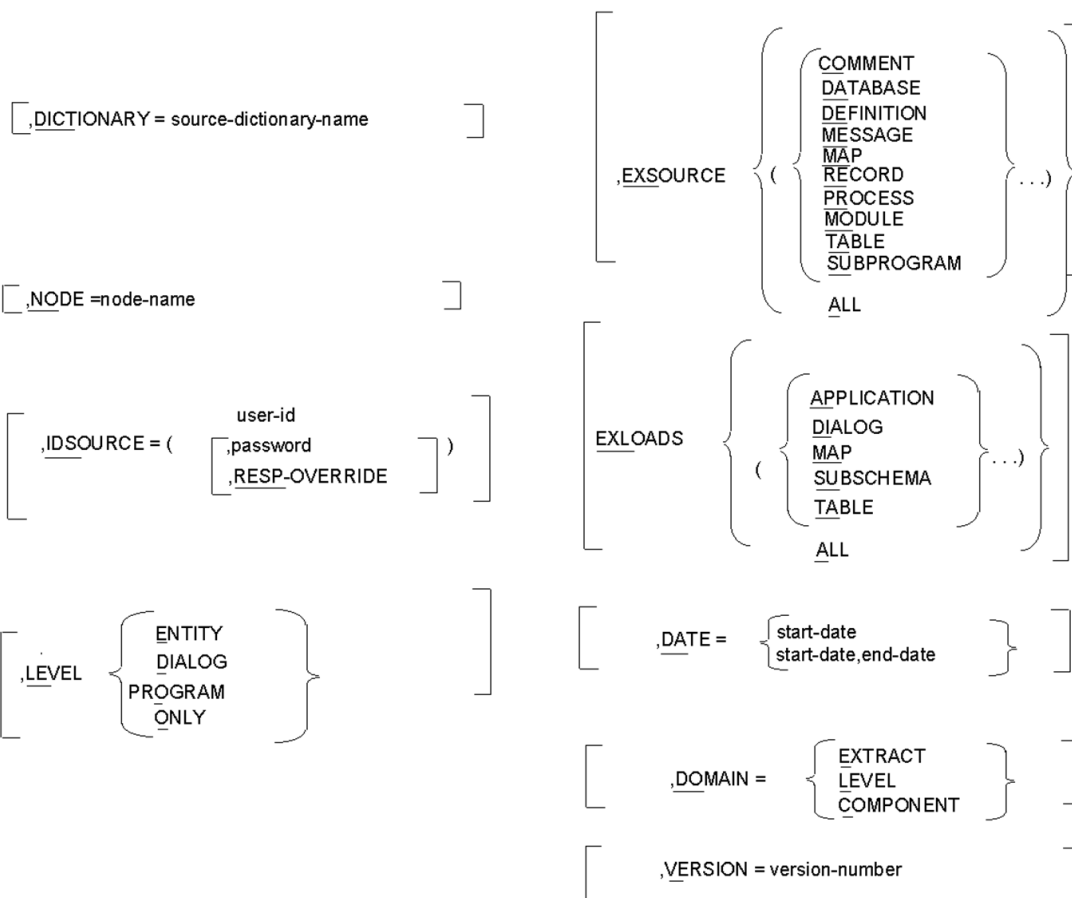
Statement Requirements

Each EXTRACT statement must specify one, and only one, entity type and name. You can enter as many EXTRACT statements as you need in the same run of CA IDMS Dictionary Migrator. More than one EXTRACT statement is required to access entities that exist in different version numbers, in alternate dictionaries, or on another machine. Even though a single entity may appear in multiple relationships, it will be extracted only once.

The following two graphics display the requirements and options of the EXTRACT statement.



EXTRACT continued...



EXTRACT Parameter

EXTRACT,

Indicates that an entity type to be extracted from the source dictionary follows. The EXTRACT statement must contain an entity-type and a name.

Note: Selection of entities related to the starting point named in the EXTRACT statement is controlled by the LEVEL parameter.

EXTRACT Statement Entity Types

One and only one of the following entity types must be selected.

CLSATTR=(class-name), (attribute-name)

Indicates that the specified attribute of the specified class is the starting point.

CLASS=class-name

Indicates that the specified class is the starting point.

ATTRIBUTE=attribute-name

Indicates that the specified attribute is the starting point. If more than one attribute with the same name exists, all attributes will be used as starting points.

SYSTEM=system-name

Indicates that the specified system is the starting point and that dialogs or programs associated with it are to be extracted.

APPLICATION=application-name

Indicates that the specified ADSA application is the starting point and that the application and dialogs or programs associated with the application are to be extracted.

USER=user-name

Indicates that the specified user is the starting point.

user-mask

Indicates that every user with a user name that matches a position-dependent identifier is extracted with its related entities.

Asterisks are used to replace characters in the user's thirty-two position name. For example, HLP**032 specifies that every user that includes HLP and 032 in the name in the indicated positions will be extracted. If blanks occur at the end of the user-mask, then blanks must occur at the end of the user-name.

```
      / program-name \  
PROGRAM = < program-mask >
```

program-name

Indicates that the specified program is to be extracted with its related entities.

program-mask

Indicates that every program with a program name that matches a position-dependent identifier is extracted with its related entities.

Asterisks are used to replace characters in the program's eight-position name. For example, HLP**010 specifies that every program that includes HLP and 010 in its name in the indicated positions is to be extracted.

```
      / dialog-name \  
DIALOG = < dialog-mask >
```

dialog-name

Indicates that the specified dialog is to be extracted with its related entities.

dialog-mask

Indicates that every dialog with a dialog name that matches a position-dependent identifier is extracted with its related entities.

Asterisks are used to replace characters in the dialog's eight-position name. For example, HLP**010 specifies that every dialog that includes HLP and 010 in its name in the indicated positions is to be extracted.

```
      / module-name \  
MODULE = < module-mask >
```

module-name

Indicates that the specified module is to be extracted with its related entities.

module-mask

Indicates that every module with a module name that matches a position-dependent identifier is extracted with its related entities.

Asterisks are used to replace characters in the module's thirty-two position name. For example:

HLP***010**

Indicates that every module that includes HLP and 010 in its name in the indicated positions is to be extracted. If blanks occur at the end of the module-mask, then blanks must occur at the end of the module-name.

LANGUAGE=language-type

A language type may be specified to select modules of a particular language or to differentiate between like named modules. Language can only be specified with entity type of module.

SCHEMA=schema-name

Indicates that every dialog owning a subschema in the specified schema is to be extracted with its related entities.

SUBSCHEMA=(subschema-name , [schema-name])

Indicates that every dialog owning the specified subschema is to be extracted with its related entities. You must also specify the schema name if the subschema is defined in more than one schema. This uniquely identifies the subschema used as an entry point.

RECORD=record-name

Indicates that every dialog owning the specified record is to be extracted with the dialog's related entities.

record-mask

Indicates that all dialogs owning every record with a record name that matches a position-dependent identifier is extracted with its related entities.

Asterisks are used to replace characters in the record's thirty-two position name. For example, HLP**32 specifies that every record that includes HLP and 032 in the record in the indicated positions is extracted. If blanks occur at the end of the record-mask, then the blanks must occur at the end of the record-name.

ELEMENT=element-name

Indicates that every dialog owning records that use the specified element is to be extracted with its related entities.

MAP=map-name

Indicates that every dialog owning the specified map is to be extracted with its related entities.

map-mask

Indicates that all dialogs owning every map with a map name that matches a position-dependent identifier is extracted with its related entities.

Asterisks are used to replace characters in the map's eight position name. For example, HLP**008 specifies that every map that includes HLP and 008 in the map in the indicated positions is extracted.

PANEL=panel-name

Indicates that every dialog owning a map that uses the specified panel is to be extracted with its related entities.

/ table-name \

TABLE = < table-mask >

table-name

Indicates that the table with the specified name is to be extracted with its related load modules.

table-mask

Indicates that every table with a table name that matches a position-dependent identifier are extracted with their related load modules.

Asterisks are used to replace characters in the table's eight-character name. For example, XXX***** indicates that every table whose name includes XXX in positions 1, 2 and 3 is to be extracted together with its related load modules.

MESSAGE=message-id

/ message-id \

MESSAGE = < (start-message-id,end-message-id) >

message-id

Indicates that a message with the specified ID is to be extracted from the source dictionary.

start-message-id,end-message-id

Indicates messages with ids that fall in the specified range are to be extracted from the source dictionary.

Table Procedure Name

specified as 'schema-name'. 'proc-name' where both 'schema-name' and 'proc-name' are 1-18 character SQL identifiers.

Indicates that the table procedure, along with all of its associated keys, is to be extracted from the source catalog specified.

SQL-table-name

specified as 'schema-name'. 'table-name' where both 'schema-name' and 'table-name' are 1-18 character SQL identifiers.

Indicates that the SQL table, along with all of its related SQL tables and entities, is to be extracted from the source catalog specified.

SQL-view-name

specified as 'schema-name'. 'view-name' where both 'schema-name' and 'view-name' are 1-18 character SQL identifiers.

indicates that the SQL view, along with all of its related SQL tables and entities, is to be extracted from the source catalog specified.

SQL-schema-name

specified as 'schema-name' where 'schema-name' is a 1-18 character SQL identifier.

Indicates that the SQL schema is to be extracted from the source catalog specified.

EXTRACT Statement Options

The parameter options available in the EXTRACT statement allow you to tailor the selection of each entity from the source dictionary.

Using Parameter Options in the EXTRACT Statement

When the parameter options are specified in the PROCESS statement, they will provide global control. If component types are excluded in the PROCESS statement, you cannot include them in the EXTRACT statement. If you have individual parameter requirements, you should indicate them in the EXTRACT statement. Some parameters may be specified in both the PROCESS and EXTRACT statements.

For example, if you specify:

EXSOURCE = MESSAGE

on the PROCESS statement and:

EXSOURCE = RECORD

on the EXTRACT statement, both MESSAGES and RECORDS would be excluded.

If the same parameters are used in both the EXTRACT and PROCESS statements, the values are combined for EXSOURCE, EXLOAD, and DOMAIN. For options with a single value, the EXTRACT statement value, if any, will override the PROCESS statement.

DICTIONARY Parameter

,DICTIONARY=source-dictionary-name or
Source-catalog-name

Is an optional parameter that allows you to access a dictionary other than the dictionary specified in the PROCESS statement or the central version's default dictionary.

For SQL entities, this parameter allows you to access a catalog other than the default system catalog.

Identifying an alternate source dictionary in the EXTRACT statement is useful when more than one dictionary resides in the central version in which CA IDMS Dictionary Migrator is processing.

Default: The dictionary as specified in the PROCESS statement. If no source dictionary is specified in the PROCESS statement, the default is the central version's default dictionary. For SQL entities, the default is the system catalog.

NODE Parameter

,NODE=node-name

Is an optional parameter that identifies a node as the communication link between central versions and CPUs. By identifying the node you can access a central version that exists in a CPU or region other than the one in which CA IDMS Dictionary Migrator is processing.

Default: You access the dictionaries that exist in the central version in which CA IDMS Dictionary Migrator is processing.

IDSOURCE Parameter

,IDSOURCE=(user-ID[,password ,[RESP-OVERRIDE]])]

Is an optional parameter that provides standard sign-on security to control access to the source dictionary.

,IDSOURCE=

indicates that valid user identification follows.

user-id

Indicates the standard identification of the user.

password

Indicates a valid password that identifies the user of the source dictionary.

Default: There is no default value. If the dictionary sign-on is required, you must enter a valid ID and authorized password with this parameter.

RESP-OVERRIDE

Releases dictionary security by generating a REGISTRATION OVERRIDE in the IDD SET OPTIONS statement. This option requires a user-ID that has full authorization.

LEVEL Parameter

LEVEL is an optional parameter that allows you to specify the path that CA IDMS Dictionary Migrator follows to extract an entity from the source dictionary.

CA IDMS Dictionary Migrator can extract an entity by using one of three methods. First, CA IDMS Dictionary Migrator can locate the programs or dialogs that use the entity specified in the EXTRACT statement and extract all existing relationships. Second, CA IDMS Dictionary Migrator can bypass program or dialog relationships for a specific entity so that only the entity specified in the EXTRACT statement and its lower-level entities will be extracted. Finally, only the entity in the Extract statement will be extracted.

You can direct CA IDMS Dictionary Migrator to bypass program or dialog relationships in EXTRACT statements that specify these entity types:

- Schema
- Subschema
- Record
- Module

- Element
- Map
- Panel
- Table
- Class
- Attribute
- Class/Attribute
- User

For example, if you specify a record in an EXTRACT statement and include LEVEL=ENTITY, CA IDMS Dictionary Migrator extracts only the specified record and the elements connected to the record.

```

      / DIALOG \
,LEVEL = < PROGRAM >
      | ENTITY |
      \ ONLY /

```

,LEVEL=

indicates that a starting point option follows. The DIALOG and PROGRAM options indicate the same level.

DIALOG

CA IDMS Dictionary Migrator traces an entity specified in the EXTRACT statement to its dialog level. All the dialogs, programs, and the components of the dialogs and programs in which the entity participates are then extracted.

PROGRAM

CA IDMS Dictionary Migrator traces an entity specified in the EXTRACT statement to its program level. All the programs and the components of the programs in which the entity participates are then extracted.

ENTITY

Indicates that an entity is to be extracted with its lower-level relationships only. This option applies to schemas, subschemas, records, elements, modules, maps, panels, tables, classes, and attributes.

ONLY

Indicates that the entity is to be extracted by itself.

Default: All entities except messages and tables are extracted with their dialog or program relationships. Messages and tables are extracted with any lower-level relationships.

EXSOURCE Parameter

```

      / / COMMENT  \ \
      | | DATABASE | |
      | | DEFINITION | |
      | | MESSAGE  | |
      | | MAP      | |
,EXSOURCE = < (< RECORD  >...)>
      | | MODULE   | |
      | | PROCESS  | |
      | | TABLE   | |
      | \ SUBPROGRAM / |
      \ \ ALL      / \
    
```

EXSOURCE (exclude source) is an optional parameter that prevents CA IDMS Dictionary Migrator from extracting specific source components. You can specify as many as nine component types to be bypassed in a single execution.

,EXSOURCE=

Indicates source component types follow.

COMMENT

Indicates that all comments associated with entities are not to be extracted.

DATABASE

Indicates that all schema-related entities that exist in the source dictionary are not to be extracted. These include all schemas, schema records and elements, subschemas, and subschema records and elements.

DEFINITION

Indicates that definitions associated with entities are not to be extracted.

MESSAGE

Indicates messages that have codes embedded in maps and dialog module source are not to be extracted.

MAP

Indicates that all map-related entities are not to be extracted. These include maps and panels, tables and messages used by the map, and records and elements owned by the map.

RECORD

Indicates that all records and elements are not to be extracted.

MODULE

Indicates that the modules (other than process code) are not to be extracted.

PROCESS

Indicates that a process code is not to be extracted.

TABLE

Indicates that table source code is not to be extracted from the source dictionary.

SUBPROGRAM

Indicates that programs or dialogs referenced by dialogs found from the EXTRACT statement are not to be extracted.

ALL

Indicates that none of the source component types are to be extracted.

Default: All source component types are extracted.

EXLOADS Parameter

EXLOADS (exclude loads) is an optional parameter that limits the load modules that CA IDMS Dictionary Migrator extracts from the DDLDCLOD (load) area of the source dictionary. You can enter up to four load module types in the same run.

```
      / / APPLICATION \ \
      | | DIALOG      | |
      | | MAP          | |
,EXLOADS = < (< MODULE >...)>
      | | SUBSCHEMA  | |
      | \ TABLE     / |
      \ ALL          /
```

,EXLOADS=

Indicates that a load module type follows.

APPLICATION

Indicates that load application modules are not extracted from source dictionary. The ADSA application cannot be updated unless the load module is migrated.

DIALOG

Indicates that dialog load modules are not to be extracted from the source dictionary. Consequently, you cannot regenerate the dialog if it does not already exist in the object dictionary.

MAP

Indicates that map load modules are not to be extracted from the source dictionary. Map load modules should not be extracted if map source is being extracted and the map will be regenerated.

MODULE

Indicates that the modules (other than process code) are not to be extracted.

SUBSCHEMA

Indicates that subschema load modules are not to be extracted from the source dictionary. Subschema load modules should not be extracted if subschema source is being extracted and the subschema will be regenerated on it if no database components are to be migrated

TABLE

Indicates that table load modules are not to be extracted from the source dictionary. Table load modules should not be extracted if table source is being extracted and the table will be regenerated.

ALL

Indicates that no load module types are to be extracted.

Default: All load modules are extracted.

DATE Parameter

,DATE=start-date

**/ start-date \
,DATE = < (start-date,end date) >**

Is an optional parameter that allows you to specify a time period in which an entity was updated. The DOMAIN parameter determines which entity occurrences will be tested.

,DATE=

Indicates that a date or date range follows.

start-date

Indicates the date, specified as mmddyy, which CA IDMS Dictionary Migrator is to search for in the dictionary. Once the entity with the indicated date or a more recent date is located, the entities which will be extracted are determined by the domain parameter.

start-date,end-date

Indicates a date range with a starting date and ending date, specified as (mmddyy,mmddyy), which CA IDMS Dictionary Migrator is to search for in the source dictionary.

Default: All entities are extracted without regard to a date updated.

DOMAIN Parameter

DOMAIN specifies where the date selection criteria will be applied.

EXTRACT

Indicates that the date range in the date parameter is tested on each entity name in an Extract Statement. If a test fails, the entity is not used as the starting point for extraction.

LEVEL

Indicates that the date range in the date parameter is tested on each entity serving as the top level of an extract path. This is dependent on the Level parameter. If the test fails neither the entity or any of its components will be migrated.

COMPONENT

Indicates that the date range in the date parameter is tested on each entity occurrence found in the extract path. If the test fails the entity is not extracted.

Default: Date range is tested on each entity named in an EXTRACT statement.

VERSION Parameter

, **VERSION=version-number**

Is an optional parameter that allows you to locate an entity occurrence based on a specific version number. This parameter is used when an entity has multiple versions in the source dictionary. Entities often have more than one version when they are being used for application development.

The specified version number applies only to the entry point. You can specify values 0-9999.

Default: The default version number is 1.

CHANGE Statement

CHANGE=(a,b)

The CHANGE statement is an optional statement which allows you to change the contents of the upload syntax files. Up to 50 change statements may be used. Each change is applied against every record in each of the upload syntax files. The changes are processed in the order in which they are entered.

a

Indicates the value which is to be changed and can have up to forty characters.

b

Indicates the value which is to replace value a and can be any string up to forty characters.

Chapter 4: Reports

This section contains the following topics:

[Introduction](#) (see page 87)

[CA IDMS Dictionary Migrator Reports](#) (see page 88)

[Extract Summary and Extract Detail Reports](#) (see page 91)

[Comparison Reports](#) (see page 95)

Introduction

This section describes and illustrates the nine reports produced by CA IDMS Dictionary Migrator. Each report tracks a different activity performed by CA IDMS Dictionary Migrator. You receive reports with detail and summary results of an extraction, comparisons of the extractions to the object dictionary, parameter input and dictionary verifications, a summary of syntax production difficulties, and a listing of syntax statements produced by CA IDMS Dictionary Migrator.

When reviewing the reports produced by CA IDMS Dictionary Migrator, you should remember the attributes of a successful migration:

Everything needed to recreate an entity in another environment is migrated or accessible.

Introduction of the migrated entities will not adversely affect anything already in the target dictionary.

The CA IDMS Dictionary Migrator reports are descriptions of what is produced after a migration. Each report tracks a different activity. Reports contain descriptions of errors, problems, or anything adversely affected in the target dictionary as well. The following is a table of the CA IDMS Dictionary Migrator reports and their summaries:

Report	Report Description
Parameter Verification Report	Displays the parameter statements entered for an execution of CA IDMS Dictionary Migrator after the syntax of the statements has been checked.
Source Dictionary Verification Report	Provides information about the source dictionary after it has been checked for entities.
Extract Summary Report	Provides information about the total number of transactions that will take place for the upload to the target dictionary.

Report	Report Description
Extract Detail Report	Provides information about each entity that will be extracted from the source dictionary and migrated to the target dictionary.
Entity Discrepancy Report	Displays information about possible conflicts or inconsistencies that could occur as result of migration.
Entity Cross-Reference Report	Displays the extracted entities that are currently used in the object dictionary.
Syntax Production Report	Displays messages reflecting problems that have occurred in the creation of syntax.
Syntax Files Display Report	Shows the contents of each syntax file that will be used by the CA IDMS utilities and compilers or the CA IDMS utilities to populate the object dictionary.
Catalog Navigation Report	Provides information about the SQL entities that will be extracted from the source catalog specified.

CA IDMS Dictionary Migrator Reports

CA IDMS Dictionary Migrator produces nine comprehensive reports:

- Parameter Verification Report
- Source Dictionary Verification Report
- Extract Summary Report
- Extract Detail Report
- Entity Discrepancy Report
- Entity Cross-Reference Report
- Syntax Production Report
- Syntax Files Display Report
- Catalog Navigation Report

From the information in the reports you can preview the migration to check parameter statements, the total number of transactions that will occur during the migration, and the detail of each transaction. You can also use the reports to compare the extracted entities to the object dictionary and review any discrepancies. If any difficulties were encountered in the production of syntax, you can use the reports to identify corrective measures. The reports give you complete control of the migration.

The following pages describe the functions and show samples of CA IDMS Dictionary Migrator reports.

Parameter Verification Report

The Parameter Verification Report displays the parameter statements entered for an execution of CA IDMS Dictionary Migrator after the syntax of the statements has been checked. The statement, as you entered it, is displayed in the right column. If there are errors, they will be marked. Informative, warning, and error messages appear in the left column of the report.

The following report shows a sample Parameter Verification Report.

The Parameter Verification Report contains:

- **release number**—the version of CA IDMS Dictionary Migrator that was installed, where nn.nn represents the release number and the subrelease number.
- **message code**—the eight-character code of an informative or error message.
- **message text**—information that explains the message code.
- **parameter statement**—display of parameter statement as you have entered it.

CA TOOLS	RELEASE nn.nn	CA IDMS/Dictionary Migrator PARAMETER VERIFICATION REPORT	DATE mm/dd/yy	TIME hh:mm:ss	PAGE nnnn
		V-----1-----2-----3-----4-----5-----6-----7-V			
NKWP091I	INPUT	PARAMETER STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP091I	CONTINUATION	STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP092I	CONTINUATION	STATEMENT			
NKWP091I	INPUT	PARAMETER STATEMENT			
EL=DIALOG					
NKWP094I	END OF	PARAMETER INPUT			

Source Dictionary Verification Report

The Source Dictionary Verification Report provides information about the source dictionary after it has been checked for entities. If the entities you requested for an extraction do not appear in the source dictionary, this report shows informative messages about those entities. This report also provides information about referenced entities within processes or maps.

For example, The following report shows a sample Source Dictionary Verification Report that provides messages about components related to other components that are being migrated. The indicated subschema does not have a corresponding load modules in the load area of the dictionary.

The Source Dictionary Verification Report contains:

- **entity type**—the type of entity referenced within a process or map.
- **entity name**—the name of the entity referenced within a process or map.
- **version number**—the version number of the entity referenced within a process or map.
- **message code**—the eight-character code of an informative or error message.
- **message text**—information that explains the message code. Detailed information about the message--reasons and appropriate actions--is provided in the chapter "Messages".

The Source Verification report will contain messages related to item 1, finding all of the necessary pieces. If components of an entity cannot be found, or if they appear discrepant with the entity, a message will be produced. The Source Verification Report is formatted as an exception report, only the deficiencies are listed.

Some of the likeliest conditions are:

1. A message is not found. See the chapter "Parameters" for conditions where you might not want to migrate messages. Take into account that by not following the suggestions listed might result in a large number of warning messages produced.
2. A source component has been changed since it was last compiled. This can occur anytime a change is made without a subsequent regeneration. However, it may indicate that a noteworthy change has been made, but not yet compiled.

- An entity is not found. Depending on the message, this may be an entity named in the parameter's EXTRACT statement or some component of another entity. In either case, it indicates that something that is needed for the migration was not found and will not be migrated, which any ca use the migration to be incomplete.

CA TOOLS	RELEASE nn.nn	CA IDMS/Dictionary Migrator SOURCE DICTIONARY VERIFICATION REPORT		DATE mm/dd/yy	TIME hh:mm:ss	PAGE nnnn
ENTITY - TYPE	ENTITY - NAME	VERSION	ERROR - MESSAGE			
LOAD MODULE	STSLRF5	1	ES00505E - INDICATED ENTITY HAD NO RELATED LOAD MODULE			

Extract Summary and Extract Detail Reports

These reports provide information about the extent of what will be migrated. The Extract Detail Report lists every entity which is related to the entity(s) named in the Extract statement(s). This is known as the "Extract Set." For each entity, its status for the migration listed:

- New (additions)
- Changed (modified)
- Dropped (not migrated)

An entity is dropped when it is presumed to be the same in the source dictionary.

The Extract Summary Report gives totals by entity type and utility.

The Extract Detail Report provides information about each entity that will be extracted from the source dictionary and entities for which delete syntax is created.

Extract Summary Report

The Extract Summary Report provides information about the total number of transactions that will take place for the upload to the CA IDMS utilities and compilers. You can use this report before an actual migration to plan for dictionary space.

The following report shows a sample Extract Summary Report. The report contains:

- **CA IDMS utility or compiler name**
- **transaction type**—the type of transaction that occurred for the utility or compiler.
- **transaction total**—the number of transactions for each utility or compiler.
- **transaction type total**—the number of transactions for each transaction type.
- **change summary**—indicates the number of add, change (modify), and drop (not migrated) transactions.

Extract Summary and Extract Detail Reports

CA TOOLS	RELEASE nn.nn	CA IDMS/Dictionary Migrator EXTRACT SUMMARY REPORT	DATE mm/dd/yy ---- CHANGE	TIME hh:mm:ss SUMMARY	PAGE nnnn -----
		TOTAL	ADDS	CHANGES	DROPS
EXTRACTION CATEGORIES					
IDMSDDL		522	0	0	0
MESSAGE	DELETES	0	0	0	0
DICTIONARY	LOAD DELETES	7	0	0	0
DIALOG	DELETES	1	0	0	0
PROGRAM	DELETES	0	0	0	0
SCHEMA	DELETES	1	1	0	0
TABLE	DELETES	0	0	0	0
QFILE	DELETES	0	0	0	0
PROCESS	DELETES	5	0	0	0
MODULE	DELETES	0	0	0	0
PGROUP	DELETES	0	0	0	0
RECORD	DELETES	20	0	0	0
REPORT	DELETES	0	0	0	0
TRANSACTION	DELETES	0	0	0	0
FILE	DELETES	0	0	0	0
ELEMENT	DELETES	228	0	0	0
SYSTEM	DELETES	0	0	0	0
ATTRIBUTE	DELETES	0	0	0	0
USER	DELETES	0	0	0	0
CLASS	DELETES	0	0	0	0
CLASS	UPDATES	0	0	1	0
USER	UPDATES	0	0	0	0
ATTRIBUTE	UPDATES	0	0	0	0
SYSTEM	UPDATES	0	0	0	0
ELEMENT	UPDATES	228	228	0	109
MODULE	UPDATES	0	0	0	0
PROCESS	UPDATES	5	5	0	0
QFILE	UPDATES	0	0	0	0
TABLE	UPDATES	0	0	0	0
FILE	UPDATES	0	0	0	0
RECORD	UPDATES	20	20	0	0
REPORT	UPDATES	0	0	0	0
TRANSACTION	UPDATES	0	0	0	0
PROGRAM	UPDATES	0	0	0	0
DIALOG	UPDATES	1	1	0	0
APPLICATION	UPDATES	0	0	0	0
DCLOAD	MODULE UPDATES	7	7	0	0
MESSAGE	UPDATES	0	0	0	0
IDMSCHEM	TRANSACTIONS	2	0	0	0
SCHEMA	UPDATES	1	1	0	0
IDMSUBSC	TRANSACTIONS	0	0	0	0
SUBSCHEMA	DELETES	5	5	0	0
SUBSCHEMA	UPDATES	5	5	0	0
RHDCMPUT	TRANSACTIONS	4	0	0	0
MAP	DELETES	1	1	0	0
PANEL	DELETES	1	1	0	0
PANEL	UPDATES	1	1	0	0
MAP	ADDS/MODIFIES	1	1	0	0
ADSOBTAT	TRANSACTIONS	0	0	0	0

Extract Detail Report

The Extract Detail Report provides information about each entity that will be extracted from the source dictionary and entities for which delete syntax is created. The report displays the entity's type, name, version number, source dictionary, and dds node.

You can use this report to review every component of a system. The review is very helpful when you have large systems in development and need to know all of their components before a migration. You can review the Extract Detail Report to determine which components should be moved from testing into production.

You can also limit the information on this report by using the NOPRINT parameter. See the chapter "Parameters" for complete details.

The following report shows a sample Extract Detail Report. The report contains:

- **entity type**—the type of entity that will be extracted.
- **entity name**—the name of the entity that will be extracted.
- **version number**—the version from which the entity will be extracted.
- **dictionary name**—the name of the source dictionary from which the entity will be extracted.
- **change-status**—indicates which entities will be added, changed (modified), or dropped (not migrated).

CA TOOLS	RELEASE	CA IDMS/Dictionary Migrator	DATE	TIME	PAGE
nn.nn	EXTRACT DETAIL REPORT	mm/dd/yy	hh:mm:ss	nnnn	
ECHO LIST OF ENTITIES TO BE EXTRACTED					
ENTITY-TYPE	ENTITY-NAME	VERSION	CHANGEONLY-STATUS	FROM DICTIONARY/NODE	
LOAD-DELETE	AAMAP1	1			
LOAD-DELETE	AAX	1			
LOAD-DELETE	EMPSLR	1			
LOAD-DELETE	EMPS01	1			
LOAD-DELETE	QALR04SS	1			
LOAD-DELETE	QALR07SS	1			
LOAD-DELETE	QALR17SS	1			
DIALOG-DELETE	AAX	1			
SUBSCHEMA-DELETE	EMPSLR	100			
SUBSCHEMA-DELETE	EMPS01	100			
SUBSCHEMA-DELETE	QALR04SS	100			
SUBSCHEMA-DELETE	QALR07SS	100			
SUBSCHEMA-DELETE	QALR17SS	100			
SCHEMA-DELETE	EMPSCHM	100			
MAP-DELETE	AAMAP1	1			
PROCESS-DELETE	AAX-PREMAP-1	1			
PROCESS-DELETE	AAX-RESPONSE-1	1			
PROCESS-DELETE	AAX-RESPONSE-4	1			
PROCESS-DELETE	AAX-RESPONSE-8	1			
PANEL-DELETE	AAMAP1-OLMPANEL	1			
RECORD-DELETE	AA-RECORD-1	1			
RECORD-DELETE	AA-RECORD-2	1			
RECORD-DELETE	AAX-RECORD-2	1			
RECORD-DELETE	COVERAGER	100			
RECORD-DELETE	DENTAL-CLAIM	100			
RECORD-DELETE	DEPARTMENT	100			
RECORD-DELETE	ELLIE-RECORD1	1			
RECORD-DELETE	ELLIE-RECORD2	1			
RECORD-DELETE	ELLIE-RECORD3	1			
RECORD-DELETE	EMPLOYEE	100			
RECORD-DELETE	EMPOSITION	100			
RECORD-DELETE	EXPERTISE	100			
RECORD-DELETE	HOSPITAL-CLAIM	100			
RECORD-DELETE	INSURANCE-PLAN	100			
RECORD-DELETE	JOB	100			
RECORD-DELETE	NON-HOSP-CLAIM	100			
RECORD-DELETE	OFFICE	100			
RECORD-DELETE	SKILL	100			
RECORD-DELETE	SQLCA	1			
RECORD-DELETE	STRUCTURE	100			
ELEMENT-DELETE	AA-ELEMENT-X	1			
ELEMENT-DELETE	AA-ELEMENT-1	1			

Comparison Reports

CA IDMS Dictionary Migrator produces two reports--Entity Discrepancy Report and Entity Cross-Reference Report—that are the results of comparing source dictionary extractions to the object dictionary. The reports show where an extracted entity is used in the object dictionary.

When CA IDMS Dictionary Migrator extracts an entity from the source dictionary, it checks the object dictionary for that entity. Once the entity is located, it is checked for use by other entities in the object dictionary. If these entities appear in the list of extracted entities, the results of the search and comparison are provided on the Entity Cross-Reference Report. If the entity is used in the object dictionary by entities that do not appear in the list of extracted entities, their inconsistencies and the impact of migrating that entity are provided on the Entity Discrepancy Report.

Entity Discrepancy Report

The Extract Discrepancy Report addresses the second part of a successful migration, avoiding detrimental impact on other entities in the target dictionary. The messages in this report show the name of the migrated entity and the name of another entity that may be affected if the migration is done.

The sample reports shown in The following two reports illustrate how comparisons and inconsistencies are reported by CA IDMS Dictionary Migrator.

The Entity Discrepancy Report contains:

- **entity type**—the type of entity that will be extracted.
- **entity name**—the name of the entity that will be extracted.
- **version number**—the version from which the entity will be extracted.
- **error message**—the eight-character code and message text. Detailed information about the message (reasons and appropriate actions) is provided in the chapter "Messages."

CA TOOLS	RELEASE	CA IDMS/Dictionary Migrator	DATE	TIME	PAGE
	nn.nn	ENTITY DISCREPANCY REPORT	mm/dd/yy	hh:mm:ss	nnnn
ENTITY DISCREPANCIES IN THE OBJECT DICTIONARY					
ENTITY-TYPE: RECORD					
ENTITY-NAME	VERSION	ERROR-MESSAGE			
TEST-REC	1	VALD219C-EXTRACT FORCED FOR SUBSCHEMA VALD219C-EXTRACT FORCED FOR SCHEMA	STSYNSUB	VERSION	1
			STATSCHM	VERSION	1

Entity Cross-Reference Report

The Entity Cross-Reference Report displays the extracted entities that are currently used in the object dictionary. The relationships of the extracted entities are reported in the sample Entity Cross-Reference Report in the following report.

If an extracted entity does not appear in either the Entity Cross-Reference Report or the Entity Discrepancy Report, the entity is not currently used by entities in the object dictionary.

The report can be omitted using the NOXREF parameter, or you can limit the amount of information on the Entity Cross-Reference Report by using the NOPRINT parameter. See the chapter "Parameters" for complete details.

The Entity Cross-Reference Report contains:

- **extract entity**—the type of entity extracted from the source dictionary
- **entity name**—the name of the entity in the source and object dictionaries
- **version**—the version number of the extracted entity
- **related entity**—the type of entity that uses the extracted entity in the object and source dictionaries
- **related entity name**—the name of the entity that uses the extracted entity in the object and in the source dictionaries
- **version**—the version of the related entity

nn.nn ENTITY CROSS-REFERENCE REPORT mm/dd/yy hh:mm:ss nnnn					
ENTITY RELATIONSHIPS IN OBJECT DICTIONARY					
EXTRACT - ENTITY	ENTITY-NAME	VERSION	RELATED-ENTITY	RELATED-ENTITY-NAME	VERSION
ELEMENT	AT-SCR-BEGIN-REPLAY	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-DIALOG-NAME	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-DIALOG-NAME-ALT	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-DIALOG-VER	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-ERROR-LIT	1	RELATES TO ELEMENT	AT-SCR-HOLD	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-ERROR-X	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-ERROR-9	1	RELATES TO ELEMENT	AT-SCR-ERROR-X	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-HOLD	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-HOLD-STATUS	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-LIMIT	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-MAX-LINES	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-QUE-TO-SCR	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-REPLAY-DATE	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-REPLAY-NBR	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-REPLAY-TIME	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-REST	1	RELATES TO ELEMENT	AT-SCR-HOLD	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-STAT-VALUE	1	RELATES TO ELEMENT	AT-SCR-HOLD	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-SCR-TRACE-INFO	1	RELATES TO ELEMENT	AT-WORK-RECORD-BEGIN	1
			RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-WORK-RECORD-BEGIN	1	RELATES TO RECORD	AT-LINK-RECORD	1
ELEMENT	AT-WORK-RECORD-END	1	RELATES TO RECORD	AT-LINK-RECORD	1

Syntax Production Report

The Syntax Production Report displays messages reflecting difficulties that have occurred in the creation of syntax. The messages will alert you to conditions which result in discontinuation of the migration process, and also indicate where the unsatisfactory conditions originate.

If messages do not appear in the Syntax Production Report, then syntax was successfully created for all entities that were extracted.

The following report shows a sample Syntax Production Report. The report contains:

- **message code**—the eight-character code of an error message.
- **message text**—information that explains the message code. Detailed information about the message--reasons and appropriate actions--is provided in the chapter "Messages".

CA TOOLS	RELEASE nn.nn	CA IDMS/Dictionary Migrator	DATE SYNTAX PRODUCTION REPORTmm/dd/yy	TIME hh:mm:ss	PAGE nnnn
PS10608I	NEW-COPY	MODULE IS	IDMS8735517033048	VERSION IS	1

Syntax Files Display Report

The Syntax Files Display Report shows the contents of each syntax file that will be used by the CA IDMS utilities and compilers or the Computer Associates utility to populate the object dictionary. This report is a hard copy of the syntax files produced by CA IDMS Dictionary Migrator. The Syntax Files Display Report is useful for verifying and documenting migrated entities.

The following report shows pages from a sample Syntax Files Display Report. The report contains:

- **CA IDMS utility or compiler or CA IDMS Dictionary Migrator utility name**—the name of the utility or compiler that uploads the file to the object dictionary
- **syntax type**—the type of syntax produced by CA IDMS Dictionary Migrator
- **syntax**—the syntax statements produced by CA IDMS Dictionary Migrator

CA TOOLS	RELEASE	CA IDMS/Dictionary Migrator	DATE	TIME	PAGE
	nn.nn	SYNTAX FILES DISPLAY REPORT	mm/dd/yy	hh:mm:ss	nnnn
IDMSDDL - ENTITY DELETION SYNTAX					
SIGNON					
USER 'DICST01'					
PASSWORD=????????					
USAGE MODE IS PROTECTED UPDATE					
.					
SET OPTIONS FOR SESSION					
INPUT COLUMNS ARE 1 THRU 80					
DELETE IS ON					
LIST					
.					
ADD					
CLASS NAME IS MIGRATOR					
ATTRIBUTES ARE MANUAL PLURAL					
.					
ADD					
CLASS NAME IS 'KENN SYS'					
ATTRIBUTES ARE MANUAL PLURAL					
.					
ADD					
ATTRIBUTE NAME IS UM299 WITHIN CLASS MIGRATOR					
.					
ADD					
ATTRIBUTE NAME IS 'TEST REC' WITHIN CLASS 'KENN SYS'					

Catalog Navigation Report

The Catalog Navigation Report provides information about each SQL table, view, schema, or table procedure that will be extracted from the source catalog and for which SQL CREATE syntax statements are generated. This report also highlights any error conditions encountered during the SQL catalog extraction process.

The report contains:

- **message-code**—the eight-character code of an informative or error message.
- **message text**—information that explains the message code. Detailed information about the message (reasons and

Note: For more information about the message(reasons and appropriate actions), see the chapter "Messages."

CA T00LS MV9507	RELEASE Rnn.n	CA IDMS/Dictionary Migrator CATALOG NAVIGATION REPORT	DATE mm/dd/yy	TIME 16:03:44	PAGE 0001
USMS004I	EXTRACT SCHEMA - SCHEMA: DEMO				
USMS002I	EXTRACT TABLE - SCHEMA: DEMO , TABLE: EMPL				
USMS010I	EXTRACT CONSTRAINT - SCHEMA: DEMO , CONSTRAINT: EMP_POS				
USMS002I	EXTRACT TABLE - SCHEMA: DEMO , TABLE: POSITION				
USMS010I	EXTRACT CONSTRAINT - SCHEMA: DEMO , CONSTRAINT: MGR_EMPL				
USMS010I	EXTRACT CONSTRAINT - SCHEMA: DEMO , CONSTRAINT: EMP_DEPS				
USMS002I	EXTRACT TABLE - SCHEMA: DEMO , TABLE: DEPENDENTS				
USMS010I	EXTRACT CONSTRAINT - SCHEMA: DEMO , CONSTRAINT: DEPT_EMPL				
USMS002I	EXTRACT TABLE - SCHEMA: DEMO , TABLE: DEPT				
USMS010I	EXTRACT CONSTRAINT - SCHEMA: DEMO , CONSTRAINT: JOB_POS				
USMS002I	EXTRACT TABLE - SCHEMA: DEMO , TABLE: JOB				
USMS010I	EXTRACT CONSTRAINT - SCHEMA: DEMO , CONSTRAINT: DEPT_BUDGET				
USMS002I	EXTRACT TABLE - SCHEMA: DEMO , TABLE: BUDGET				
USMS011I	EXTRACT TABLE PROCEDURE - SCHEMA: DEMO , PROCEDURE: TESTPROC_1				
USMS012I	EXTRACT KEY - KEYNAME: PROCL_PRIME_KEY , ON SCHEMA: DEMO , PROCEDURE: TESTPROC_1				
USMS012I	EXTRACT KEY - KEYNAME: PROCL_UNIQ_KEY , ON SCHEMA: DEMO , PROCEDURE: TESTPROC_1				
USMS012I	EXTRACT KEY - KEYNAME: PROCL_STD_KEY , ON SCHEMA: DEMO , PROCEDURE: TESTPROC_1				
USMS001I	CATALOG NAVIGATION COMPLETED				

Chapter 5: Operations

This section contains the following topics:

[Introduction](#) (see page 101)

[Operational Considerations](#) (see page 101)

[Additional Tasks Performed by CA IDMS Dictionary Migrator](#) (see page 102)

[How CA IDMS Dictionary Migrator Operates](#) (see page 102)

[Running as a Single Job Execution Under Central Version](#) (see page 108)

[Running as a Two-Job Execution](#) (see page 110)

Introduction

This chapter discusses CA IDMS Dictionary Migrator operations. It provides operational considerations, information about allocating space for the work and syntax files, how the files are used, system flow, and additional features.

Operational Considerations

Before and during the execution of CA IDMS Dictionary Migrator there are some important operational factors to consider:

- Before running CA IDMS Dictionary Migrator, the object dictionary should be backed up.
- All CA IDMS conventions regarding locking the source and object dictionaries against access are used while CA IDMS Dictionary Migrator is executing. To ensure accurate reporting prevent central version (CV) or local CA IDMS update jobs from accessing the dictionaries while CA IDMS Dictionary Migrator is executing.
- The network schema, subschemas, and their entities cannot be extracted by CA IDMS Dictionary Migrator.
- When the source dictionaries defined in the PROCESS and EXTRACT statements are at different CA IDMS levels, the EXTRACT statement is bypassed.
- Both dictionaries must be accessed in the same mode, local or CV. If SYSCTL file is used and names the Gateway CV, both dictionaries must be accessible to the CV directly, or through node communication. Node communication is discussed in the *CA IDMS System Operations Guide*.
- The object dictionary is not updated by extracting and comparing entities, or producing syntax. Updating the object dictionary occurs during the upload step, using the CA IDMS utilities.
- An CA IDMS environment is always started in the CA IDMS Dictionary Migrator region (partition).

Local Mode Processing

The following considerations must be taken into account for local mode processing:

- Additional JCL and EXEC statements may be needed to access files for: dictionaries, journals, UPSI, DDLDCMSG area, and the dictionary load area depending on the contents of the DMCL.
- Both dictionaries areas must be defined in the same DMCL module.
- The SYSIDMS file contents control if the job runs in local mode.

Note: For more information about the necessary parameters, see the *CA IDMS Common Facilities Guide*. Do not, however, use Dictname, Nodename, Dbname or Dictnode parameter in the SYSIDMS file.

Additional Tasks Performed by CA IDMS Dictionary Migrator

Although CA IDMS Dictionary Migrator's primary function is to transfer systems from testing to production, it can also be used for these dictionary analysis and maintenance tasks:

- Thoroughly and accurately analyze a dictionary's contents. By identifying the same dictionary in both the DICTIONARY and OBJDICTIONARY parameters and specifying RUN=VERIFY or MIGRATE or AUDIT, you can direct CA IDMS Dictionary Migrator to compare the source dictionary to itself. The reports produced from this type of run show all existing relationships and entities within the source dictionary.
- Redefine schema records and elements as separate entities. When CA IDMS Dictionary Migrator decompiles a schema, it produces syntax to add the records and elements to the object dictionary as separate entities. It also produces record COPY statements for the schema definition. This enables fields to be modified without disrupting a schema structure.
- Archive a system or dictionary to a permanent data set. By specifying RUN=EXPORT, you can direct CA IDMS Dictionary Migrator to create an extract file from the source dictionary that can then be saved on tape or disk.

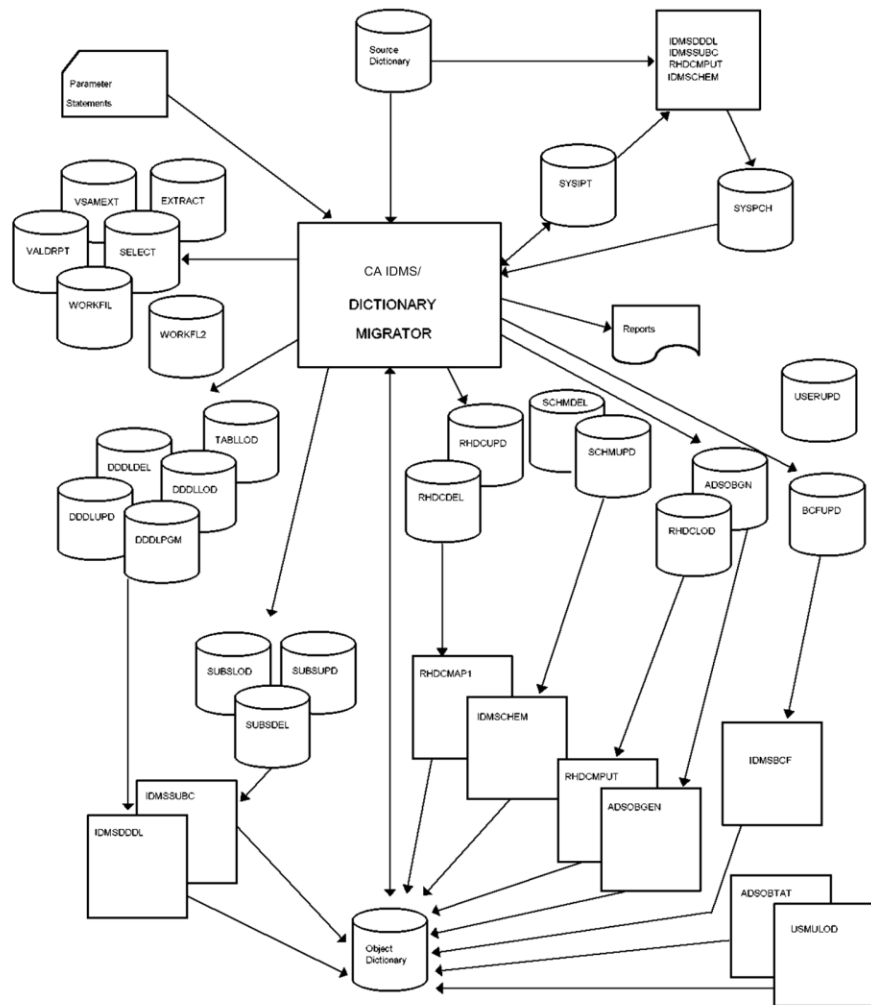
How CA IDMS Dictionary Migrator Operates

CA IDMS Dictionary Migrator is a parameter-driven process that uses work and syntax files. The work files are used as temporary storage for dictionary entities as they are extracted from the source dictionary, sorted, compared to the object dictionary, and retained for reports and syntax generation. CA IDMS Dictionary Migrator then produces the syntax statements for entities that are to be transferred to the object dictionary and stores these statements on syntax files. The syntax files are processed by CA IDMS utilities and compilers and the CA IDMS utilities provided with CA IDMS Dictionary Migrator to populate the object dictionary.

Extracting and comparing entities, and producing syntax does not update the object dictionary. Updating the object dictionary occurs during the final upload step, using the CA IDMS utilities and compilers and the Computer Associates utility provided with CA IDMS Dictionary Migrator.

Two Methods of Operation

There are two methods of operating CA IDMS Dictionary Migrator--as a single job execution or as a two-job execution. A single execution (see the following diagram) is used when the source and object dictionaries exist in the same CPU or when the dictionaries exist in CPUs that can be accessed by node communication. A two-job execution (EXPORT/IMPORT) is used when CA IDMS Dictionary Migrator is processed in one CPU and the object dictionary exists in another CPU without node communication (see the next diagram).



Allocating Space for Work Files and Syntax Files

CA IDMS Dictionary Migrator requires space allocations for as many as eight work files and sixteen syntax files. One of the work files--VSAMEXT--must be a VSAM file. In addition to allocating space for the work and syntax files, the actual run of CA IDMS Dictionary Migrator requires you to allocate adequate space for sort-work files.

The first of the following tables describes each work file and suggests space allocations.

The second table provides the following information about syntax files:

- The function of each file
- The order in which the files are used by CA IDMS utilities and compilers during the upload phase
- The region size needed for each utility or compiler
- The suggested space allocation for each file.

The actual number of tracks for the work and syntax files varies depending on the size of the system being migrated, and upon the device types assigned. You may need to adjust the numbers listed in the following two tables.

Changing Block Size of Work Files and Syntax Files in z/OS

The block size of work files and syntax files can be changed by using the DCB parameter of the DD statement. The LRECL (logical record length) of each work file is listed in the first of the following tables. The LRECL of each syntax file used by the CA IDMS utility provided with CA IDMS Dictionary Migrator or CA IDMS utility or compiler is 80.

z/VSE File Assignments

An ASSIGN statement is required for every work and syntax file processed by CA IDMS Dictionary Migrator. This assignment is necessary because CA IDMS Dictionary Migrator has its own device-independent support that dynamically builds a DTF based on the device type indicated by the assignment of the logical unit. The logical unit required for each work and syntax file is provided in the following two tables.

z/VSE File Allocation Alternate Method

Occasionally, you may receive the message "FILE ASSIGNED to SYSnnn IS NOT VSAM." This means that the data set is processed SAM instead of VSAM because CA IDMS Dictionary Migrator was unable to find the data set in the VSAM catalog. SYS016, however, must be a VSAM file.

File Name (z/VSE Logical Unit)	Required	Type	LRECL	Suggested Space	Description
MIGPARM	yes	disk tape card	80	1,1	z/OS and z/VM--Input file for CA IDMS Dictionary Migrator parameter statement
SYS011	yes	disk tape card	80	1,1	z/VSE only--Input file for CA IDMS Dictionary Migrator parameter statements
SYSIPT (SYS012)	yes	disk	80	30,60	Input to CA IDMS IDMSDDDL, IDMSSUBC, RHDCMPUT, and IDMSCHEM utilities
SYSDPC (SYS018)	yes	disk	80	150,60	Output from IDMSDDDL, IDMSSUBC, RHDCMPUT, and IDMSCHEM utilities
SELECT (SYS014)	**	disk	176	2,1	Contains encoded extract statements
EXTRACT	yes	disk	228	150,60	Contains entities from source dictionary as they are extracted
VSAMEXT	**	VSAM KSDS	228	60,30	Contains extracted entities for dictionary comparisons
VALDRPT	**	disk	138	5,2	Sorts and controls information from dictionary comparisons for reports
WORKFIL	**	disk	80	60,30	Temporary holding file for syntax-creation or modification

File Name (z/VSE Logical Unit)	Required	Type	LRECL	Suggested Space	Description
WORKFL2 (SYS034)	**	disk	228	60,30	Required to construct combination file for entities extracted from the object dictionary
SYSCTL	***	disk	20	pre- allocated share	Defines access to the central version which will be the gateway CV if a node is specified.
SYSIDMS	***	card disk	80	pre- allocated share	See the DBA Reference. Never specify DBNAME, NODENAME, DICTNODE, or DICTNAME. Use DMCL=dmclname, LOCAL=ON, and JOURNAL=OFF to run local (minicv) mode.

*Track Allocation expressed for 3350 disk device (30 tracks = 1 cylinder).

**File requirement depends on the run type.

***File requirement depends on environment.

Exhibit 5.2: Work Files Table

Step Order	Syntax File (z/VSE logical unit)	Suggested Space	CA IDMS Utility or Compiler	Region Size	Step Function
1	RHDCDEL (SYS019)	5,2	RHDCMAP1	4000K	Panel and map deletion.
2	SCHMDEL (SYS020)	5,2	IDMSCHEM	3000K	Schema and subschema source deletion.
3	SUBSDEL (SYS021)	5,2	IDMSSUBSC	2500K	Subschema source deletion. This step is not necessary if you deleted the schema.
4	DDDLDEL (SYS022)	5,2	IDMSDDDL	3000K	Element, record, table deletion.
5	DDDLUPD (SYS023)	60,30	IDMSDDDL	3000K	ADD/MODIFY for elements, records, messages, modules, class, tables, attribute, and system.

Step Order	Syntax File (z/VSE logical unit)	Suggested Space	CA IDMS Utility or Compiler	Region Size	Step Function
6	BCFUPD (SYS040)	5,2	IDMSBCF	1024K	Create statements for tables, views, and SQL schemas.
7	TABLLOD (SYS024)	5,2	IDMSDDDL	3000K	MODIFY for tables. Generates table load modules in the DDLCOD area.
8	DDDLLOD (SYS025)	5,2	IDMSDDDL	3000K	ADD/Modify of load modules for subschemas, maps, map editing tables, and dialog processes.
9	SCHMUPD (SYS026)	5,2	IDMSCHEM	2500K	Upload of schema modifications — source statements written in schema DDL and any data copied from the dictionary to add schemas.
10	SUBSUPD (SYS028)	5,2	IDMSSUBSC	2500K	Upload of subschemas.
11	SUBSLOD (SYS029)	5,2	IDMSSUBSC	2500K	Generates subschema load modules in DDLDCLOD area.
12	RHDCUPD (SYS030)	5,2	RHDCMAP	4000K	ADD/MODIFY of panels and maps.
13	RHDCLOD (SYS031)	60,30	RHDCMPUT	4000K	Compiles maps and stores map load modules in DDLDCLOD area.
14	ADSOBGN (SYS032)	1,2	ADSOBCOM	5000K	Generates dialogs. This step can only be executed if dialog load modules were migrated.
15	DDDLPGM (SYS033)	5,2	IDMSDDDL	3000K	ADD/MODIFY of program statements.
16	ADSBTAT (SYS037)	1,1	ADSOBTAT	3000K	Updates Task Application Table (revised ADSA applications only).
17	USERUPD (SYS038)	1,1	USMULOD	3000K	Updates passwords on migrated user records to reflect source database.

*Track allocation expressed for 3350 disk device (30 tracks = 1 cylinder).

Running as a Single Job Execution Under Central Version

Executing CA IDMS Dictionary Migrator as a single job means that you are performing a migration between two dictionaries that exist on the same machine or two machines that have node communication. You use the PARMCHECK, VERIFY, MIGRATE, or AUDIT option in the RUN parameter for a single execution.

Migration Activities - Single Job Execution

The following pages present a description of the activities for a single job execution of CA IDMS Dictionary Migrator. The complete JCL and key appear in the JCL members that were downloaded from your installation media. The JCL for a single execution perform the following migration activities:

Step 1—USMVSAM

The JCL model contained in the distribution SAMPJCL library member USMVSAM (z/OS), TOOLJCL library member USMVSAM.S (z/VSE), or the USMVSAM EXEC (z/VM), allocates the VSAM work file. The key to the JCL is also contained in USMVSAM.

Step 2--USMSYNTAX—z/OS Only

The JCL model contained in the distribution SAMPJCL library member USMSYNTAX (z/OS only) allocates the syntax files. The key to the model JCL is also contained in USMSYNTAX. In a z/VSE or z/VM environment, the syntax files are allocated as part of the USMXTRCT step.

Step 3—USMXTRCT

The JCL model contained in the distribution SAMPJCL library member USMXTRCT (z/OS), TOOLJCL library member USMXTRCT.S (z/VSE), or the USMXTRCT EXEC (z/VM), allocates the work and syntax files (z/VSE only), extracts information from the source dictionary, compares the extract to the object dictionary, reports on the comparison, and produces the syntax files. The key to the model JCL is also contained in USMXTRCT.

Step 4—USMLOAD1

The JCL model contained in the distribution SAMPJCL library member USMLOAD1 (z/OS) TOOLJCL library member USMLOAD1.S (z/VSE), or the USMLOAD1 EXEC (z/VM), each show a step for uploading a syntax file to the object dictionary through an CA IDMS utility supplied by Computer Associates. The key to the JCL is also contained in USMLOAD1. There can be from 1 to 16 steps involved in the upload, depending on your site and the type of migration being performed. See the second of the previous tables to determine:

- The steps you need to perform
- The function of each step
- The order in which the syntax files must be processed
- The utility or compiler to be used for each file.

For each step, you must supply values for all the variables shown in the step.

Note: Some utilities are executed more than once. Do not combine syntax files for the same utility. Process the syntax files in the order outlined in the second of the previous tables to correctly populate the object dictionary.

Step 5—USMLOAD3 (Optional)

The JCL model contained in the distribution SAMPJCL library member USMLOAD3 (z/OS), TOOLJCL library member USMLOAD3.S (z/VSE), or the UMLOAD3 EXEC (z/VM), is an optional step that, when executed, deletes the migrated source from the source dictionary after migration. The key to the model JCL is also contained in USMLOAD3.

If you want to delete entities from the source dictionary after migration, make a second non-CHANGEONLY run using the same parameters as the real migration, but specifying the same dictionary for both source and object.

Step 6—USMSQLOD (Optional)

If you are using CA IDMS Dictionary Migrator to extract SQL entity definitions from a source catalog, the JCL model contained in the distribution SAMPJCL library member USMSQLOD (z/OS), TOOLJCL library member USMSQLOD.S (z/VSE), or the USMSQLOD EXEC (z/VM), can be used to update your target catalog with the syntax for the extracted SQL entities.

Using CREATESYNTAX Run Type

You can use the files created in a VERIFY run as input to CREATESYNTAX run. As a result, you can create syntax files using the data that already has produced the migration reports.

The JCL model contained in the distribution SAMPJCL library member USMIMPRT (z/OS), executes the CREATESYNTAX step.

CREATESYNTAX is useful when the VERIFY run does not show errors that require correcting. Therefore, the CREATESYNTAX run begins where the VERIFY run stops.

Running as a Two-Job Execution

To perform a migration between two machines that do not share node communication, you must run CA IDMS Dictionary Migrator twice (in two jobs) using the EXPORT/IMPORT options in the RUN parameter. The following flowchart illustrates the system flow of a two-job execution for z/OS and z/VSE.

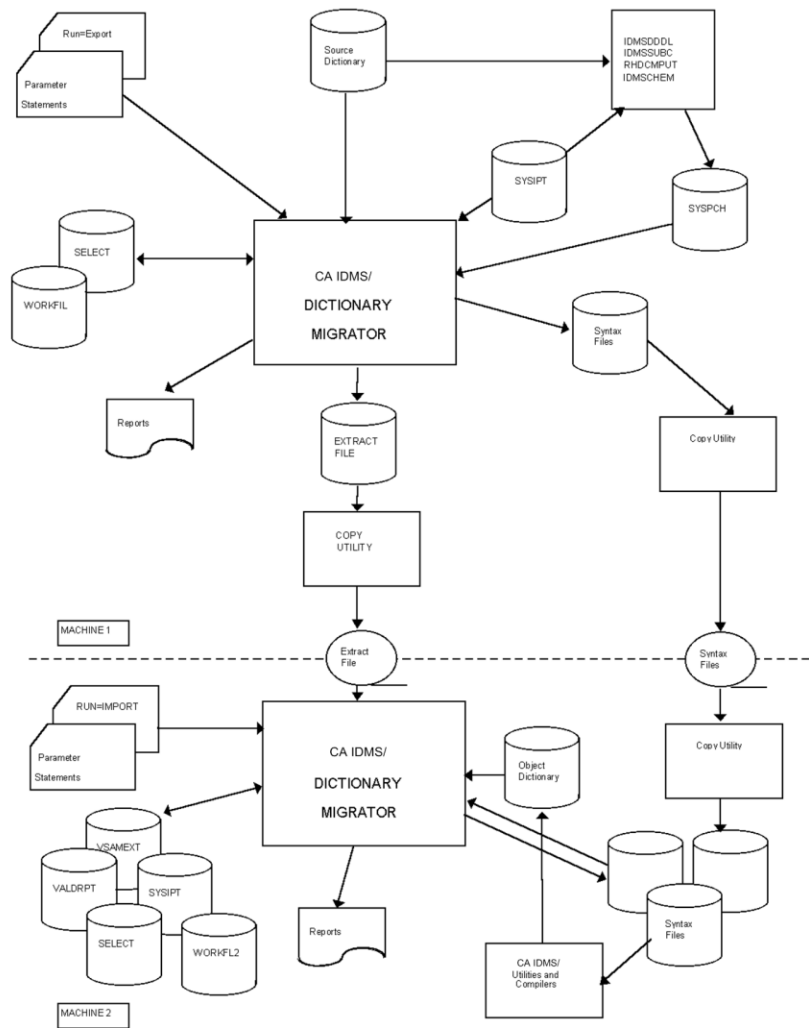
During the first job -- EXPORT -- CA IDMS Dictionary Migrator extracts information from a source dictionary in one machine and produces syntax, and USMCOPY stores the information and syntax on tape.

During the second job -- IMPORT -- CA IDMS Dictionary Migrator compares the source dictionary information on tape to the object dictionary in another machine and produces detailed reports of the analysis.

The CA IDMS utilities and the CA IDMS utilities supplied with CA IDMS Dictionary Migrator then upload the syntax files to the object dictionary.

Migration Activities - Two-Job Execution

The following pages present a description of the activities for a two-job execution of CA IDMS Dictionary Migrator. The complete JCL and keys are contained in the JCL members that were downloaded from your installation media. A two-job execution performs the migration activities listed below.



Step 1—USMSYNTAX--z/OS Only

The JCL model contained in the distribution SAMPJCL library member USMSYNTAX (z/OS) allocates the syntax files. The key to the model JCL is also contained in USMSYNTAX. In a z/VSE environment, the syntax files are allocated as part of the USMEXPRT step.

Step 2—USMEXPRT

The JCL model contained in the distribution SAMPJCL library member USMEXPRT (z/OS), TOOLJCL library member USMEXPRT.S (z/VSE), or the USMEXPRT EXEC (z/VM), allocates the syntax files (z/VSE only), allocates the work files, extracts information from the source dictionary and puts it onto disk, and produces syntax files on disk. The key to the model JCL is also contained in USMEXPRT.

Step 3—Utility Copy

Use a copy utility to copy the extract file and all syntax files onto a medium that can be transported to the target machine. Model z/OS JCL that performs this action and places all files on one tape can be found in source library member USMCOPY (z/OS). The key to the model JCL is contained in USMCOPY.

Step 4—Utility Copy

Use the same copy utility used in Step 3 to copy the transported files onto disk on the target machine.

Step 5—USMVSAM

The JCL model contained in the distribution SAMPJCL library member USMVSAM (z/OS), TOOLJCL library member USMVSAM.S (z/VSE), or the USMVSAM EXEC (z/VM), allocates the VSAM work file. The key to the model JCL is also contained in USMVSAM.

Step 6—USMIMPRT

The JCL model contained in the distribution SAMPJCL library member USMIMPRT (z/OS), TOOLJCL library member USMIMPRT.S (z/VSE), or the UMIMPRT EXEC (z/VM), imports the extract tape, compares extract to object dictionary, and reports on the comparison. The key to the model JCL is also contained in USMIMPRT.

Step 7—USMLOAD1

The JCL model contained in the distribution SAMPJCL library member USMLOAD1 (z/OS), TOOLJCL library member USMLOAD1.S (z/VSE), or the USMLOAD1 EXEC (z/VM), is a step for uploading a syntax file to the object dictionary through an CA IDMS utility or through the CA ADSA application upload utility supplied by Computer Associates with CA IDMS Dictionary Migrator. The key to the model JCL is also contained in USMLOAD1. There can be from 1 to 16 steps involved in the upload, depending on your site, the type of migration being performed, and whether you want to delete the migrated syntax from the source dictionary.

See the previous table to determine:

- The steps you need to perform.
- The function of each step.
- The order in which the syntax files must be processed.
- The utility or compiler to be used for each file.

For each step, you must supply values for all the variables shown in the step.

Note: Some utilities are executed more than once. Do not combine syntax files for the same utility. Process the syntax files in the order outlined in Exhibit 5.3 to correctly populate the object dictionary.

Step 8—USMLOAD3 (Optional)

The JCL model contained in the distribution SAMPJCL library member USMLOAD3 (z/OS), TOOLJCL library member USMLOAD3.S (z/VSE), or the USMLOAD3 EXEC (z/VM), deletes the migrated entities from the source dictionary after migration. The key to the model JCL is also contained in USMLOAD3. This step is optional. If you want to delete entities from the source dictionary after migration, make a second non-CHANGEONLY run using the same parameters as the real migration, but specifying the same dictionary for both source and object.

Step 9—USMSQLOD (Optional)

If you are using CA IDMS Dictionary Migrator to extract SQL entity definitions from a source catalog, the JCL model contained in the distribution SAMPJCL library member USMSQLOD (z/OS), TOOLJCL library member USMSQLOD.S (z/VSE), or the USMSQLOD EXEC (z/VM), can be used to update your target catalog with the syntax for the extracted SQL entities.

Chapter 6: Dictionary Migrator Assistant

This section contains the following topics:

- [Introduction](#) (see page 115)
- [DMA Requirements](#) (see page 116)
- [User Authority](#) (see page 117)
- [Types of Messages](#) (see page 117)
- [DMA Commands](#) (see page 119)
- [Display PF Key Values](#) (see page 122)
- [Questions and Answers About Using DMA](#) (see page 123)
- [First Steps in Using DMA](#) (see page 125)
- [Reset and Exit DMA](#) (see page 125)
- [Common Fields on DMA Screens](#) (see page 128)
- [DMA Activity Paths](#) (see page 129)
- [Define a New Migration: Sequence of Screens](#) (see page 129)
- [Define a New Migration: Overview](#) (see page 131)
- [Complete First Steps Using DMA](#) (see page 131)
- [1. Describe the Migration Environment](#) (see page 132)
- [2. Specify a New Starting Point](#) (see page 132)
- [3. Accept or Override Defaults](#) (see page 133)
- [4. Specify Starting Point Overrides](#) (see page 134)
- [5. Include/Exclude Source Types \(Current Starting Point\)](#) (see page 135)
- [6. Include/Exclude Load Types \(Current Starting Point\)](#) (see page 135)
- [7. Obtain CA IDMS Dictionary Migrator Reports](#) (see page 136)
- [8. Store Setup into a PARMFILE](#) (see page 137)
- [9. Continue to Use DMA](#) (see page 138)
- [Global Overrides](#) (see page 138)
- [Use a Series of Commands from the Menu](#) (see page 140)
- [Specify Migration Overrides](#) (see page 140)
- [Limit Entities for Overall Migration](#) (see page 141)
- [Select a DMA Parameter File](#) (see page 143)
- [Specify a DMA Parameter File](#) (see page 143)
- [Modify a DMA Parameter File](#) (see page 144)
- [Copy a DMA Parameter File](#) (see page 147)
- [Submit Parameters and JCL Online](#) (see page 147)
- [Submit Parameters and JCL from the Batch Environment](#) (see page 148)
- [Copy CA IDMS/Dictionary Migrator JCL to DMA](#) (see page 148)

Introduction

CA IDMS Dictionary Migrator Assistant (DMA) is an online tool that assists you in selecting parameters to supply to CA IDMS Dictionary Migrator. This section provides an overview of DMA requirements and user authority, and tells you how to use it.

DMA Requirements

The system and storage requirements for operating DMA are listed below.

Installation

You must upload CA IDMS Dictionary Migrator JCL to DMA before you can use DMA. For information on installing DMA and uploading CA IDMS Dictionary Migrator JCL to DMA, see the *Installation Guide*.

System Requirements

System requirements include the operating environment, terminal type, and security.

Terminal Type

DMA supports 3270 or 3270-compatible terminals, models 2, 3, 4, or 5.

Security

You can limit access to Dictionary Migrator Assistant by setting security on the task code DMA, or on the modified task code specified in your sysgen tables.

Within a DMA session, you can delete or modify your own parameter files. You cannot delete or modify another user's parameter files. You may have authority to copy another user's parameter files. Consult your DBA.

Storage Requirements

DMA requires program storage, working storage, scratch storage, and database storage.

Program Storage

Because all DMA modules are reentrant, only one copy of DMA is required to support multiple users. When DMA is in use, 498K is the most program storage required at any one time.

Working Storage

Each user who is currently using DMA may require up to 25K of working storage.

Scratch Storage

Scratch records are used during each DMA session. For each user, 0.5K of scratch storage is required unless the SHOW or JCL function is in use. If the SHOW and/or the JCL function is in use, 5K is required.

Database Storage

For each parameter file stored in the PARMFILE database, 4K of database storage is required.

User Authority

Security may be in effect for your company's dictionaries. If so, you need authority to read the source dictionary and modify the target (object) dictionary. See your DBA concerning any questions you may have about dictionary authority.

In addition, security may be in effect in your company for using CA IDMS Dictionary Migrator. If so, you need authority to use CA IDMS Dictionary Migrator. Again, see your DBA for any questions you may have concerning authority to use CA IDMS Dictionary Migrator.

You may also need authority to perform DMA activities. Depending upon your company policy, you may need authority for all or selected DMA activities, including:

- Defining a new migration
- Modifying a previously-generated parameter file and JCL
- Copying another user's parameter files
- Submitting parameters (previously generated through DMA) and JCL to CA IDMS Dictionary Migrator

Types of Messages

DMA displays messages either in the message areas on DMA screens or on confirm screens.

Screen Messages

DMA displays informative, warning, and error messages in the message area on the third line of every screen. These messages, along with reasons for occurrence and suggested actions, are listed in the chapter "Messages".

Confirm Screens

DMA displays confirmation messages on separate screens. A confirm screen appears when you enter:

- A RESET command
- An exit command (X)
- A D on the List of Starting Points Screen
- A D on the List of Parmfiles Screen

Sample confirm screens are shown in the next screen.

The confirm screen reminds you of the consequences of executing the command. It also gives you a choice of either executing the command or returning to the current setup without executing the command.

To execute the command:

- Press the ENTER key

To return to the current setup without executing the command:

- Type PREVIOUS in the command line and press the ENTER key, or press PF3

```
CA IDMS/DMA Rnn.nn --- Confirm Setup RESET -----hh:mm mm/dd/yy
COMMAND ==> XDMACFM
*****
*
* WARNING ----- WARNING ----- WARNING *
*
* A RESET command has been entered. If the RESET is executed: *
* 1) The current setup will be erased. *
* 2) You will go to the 'Select DMA Activity' to restart the setup *
* Press "ENTER" to execute the RESET. *
* Use the "PREVIOUS" command to continue the setup. *
*
*****
```

```

CA IDMS/DMA Rnn.nn --- Confirm Session Exit -----hh:mm mm/dd/yy
COMMAND ==> XDMAXFM
*****
*
*   WARNING ----- WARNING ----- WARNING   *
*
*   An EXIT command has been entered. If the EXIT is executed: *
*
*       1) The current setup will be erased. *
*
*       2) You will Exit CA-IDMS/Dictionary Migrator Assistant. *
*
*   Press "ENTER" to execute the EXIT. *
*
*   Use the "PREVIOUS" command to continue the session. *
*
*****

```

DMA Commands

DMA commands allow you to execute DMA functions, transfer to other functions, or view more data within the same function.

In the next screen, all of the valid DMA commands are listed. For each command that implements a transfer to another screen, that screen is listed. Otherwise the function is described. The table also lists a default PF key for each command.

The screen currently displayed determines which commands are active. For example, when you are viewing the ConfirmSession EXIT screen, the only active commands are HELP, KEYS, ENTER, and PREVIOUS.

Enter a Command

To enter a command press the PF key associated with that command or:

- In the command line, type the whole command or only the first characters unique to the command. The optional portion of each command is enclosed in square brackets. See the next screen for a list of commands.
- Press the ENTER key.

```
CA IDMS/DMA Rnn.nn --- Display PF Key Values -----hh:mm mm/dd/yy
COMMAND ==>
PF KEY VALUES: PRIMARY KEYS
PF1 ==> HELP
PF2 ==> NEXT
PF3 ==> PREVIOUS
PF4 ==> START
PF5 ==> SPOVER
PF6 ==> FILES
PF7 ==> UP
PF8 ==> DOWN
PF9 ==> OUTPUT
PF10 ==> ENVIRON
PF11 ==> JCL
PF12 ==> MENU

INSTRUCTIONS: PRESS "ENTER" KEY TO PROCESS CHANGES AND DISPLAY ALTERNATE KEYS
              Enter "PREVIOUS" command to process changes and RETURN
```

```
CA IDMS/DMA Rnn.nn --- Display PF Key Values -----hh:mm mm/dd/yy
COMMAND ==>
PF KEY VALUES: ALTERNATE KEYS
PF13 ==> RESET
PF14 ==> OVERRIDE
PF15 ==> SWAP
PF16 ==> KEYS
PF17 ==> STORE
PF18 ==> SHOW
PF19 ==> WITHSRC
PF20 ==> SUBMIT
PF21 ==> SLISTT
PF22 ==> WITHLOAD
PF23 ==> ENTER
PF24 ==> =X

INSTRUCTIONS: PRESS "ENTER" KEY TO PROCESS CHANGES AND DISPLAY ALTERNATE KEYS
              Enter "PREVIOUS" command to process changes and RETURN
```


System Commands Menu

DMA commands can be selected from the System Commands menu. At any time after you have selected a DMA activity, you can access the System Commands Menu by typing *MENU* in the command line and pressing the ENTER key, or by pressing PF12.

The System Commands Menu is shown in the next screen.

To select a command from the menu:

1. To the left of the command, type S
2. Press the ENTER key.

You can also enter a command in the command line on the System Commands screen.

```

CA IDMS/DMA Rnn.nn --- System Commands -----hh:mm mm/dd/yy
COMMAND ==>                                         XDMAMNU
(S)elect one or more of the following functions OR enter one COMMAND:
  FILES   - LIST of PARMFILES
  STORE   - Store Setup into PARMFILE
  ENVIRON - Describe the Migration Environment
  OVERRIDE - Specify Migration Overrides
  OUTPUT  - Specify Migration Output
  START   - Specify a new STARTING POINT
  SPOVER  - Specify STARTING POINT Overrides
  SLIST   - LIST of STARTING POINTS
  WITHSRC - Include/Exclude SOURCE Types
  WITHLOAD - Include/Exclude LOAD Types
  JCL     - Edit JCL for Migration
  SHOW    - BROWSE Parameters for Migration
  RESET   - ERASE setup and RESTART
  SUBMIT  - SUBMIT current setup to CA-IDMS/Dictionary Migrator
  KEYS    - DIsply Key Values
  HELP    - Display HELP Screen
  X       - EXIT from DMA

```

Multiple Commands

To specify more than one command on the System Commands menu:

1. Type S to the left of each selected command.
2. Press the ENTER key.

The first (nearest the top) command is executed.

When you have finished with the first screen, **Press PF3**.

The next command on the System Commands menu is then executed.

Display PF Key Values

To provide quick access to DMA functions, all of the system commands are assigned to PF keys. Values assigned to the primary and alternate PF keys are displayed on the Display PF Key Values screen. The default values of the PF keys are shown in the next screen.

Access the Values Screen

To access the Display PF Key Values screen:

1. Type KEYS in the command line on any screen.
2. Press the ENTER key.
3. Alternate method: press PF16.

Change a PF Key Value

To change the value of a PF key:

1. On the Display PF Key Values screen, type the new value over the old value.

For example, to change PF5 to OUTPUT, place the cursor at the beginning of SPOVER after PF5, and type OUTPUT. If you then press the ENTER key without changing PF9, both PF5 and PF9 will have the value OUTPUT.

Which PF Key Values Apply?

If you change any PF key values, the PF keys retain the new values throughout your current DMA session, until you exit DMA.

The screen currently displayed determines which commands are active. For example, when you are at the Confirm Session EXIT screen the only active commands are HELP (PF1), KEYS (PF16), ENTER (PF23), and PREVIOUS (PF3).

```

CA-IDMS/DMA Rnn.nn --- Display PF Key Values -----hh:mm mm/dd/yy
COMMAND ==>                                         XDMAKEY
PF KEY VALUES:  PRIMARY KEYS
PF1  ==> HELP
PF2  ==> NEXT
PF3  ==> PREVIOUS
PF4  ==> START
PF5  ==> SPOVER
PF6  ==> FILES
PF7  ==> UP
PF8  ==> DOWN
PF9  ==> OUTPUT
PF10 ==> ENVIRON
PF11 ==> JCL
PF12 ==> MENU

INSTRUCTIONS:  PRESS "ENTER" KEY TO PROCESS CHANGES AND DISPLAY ALTERNATE KEYS
               Enter "PREVIOUS" command to process changes and RETURN
    
```

```

CA-IDMS/DMA Rnn.nn --- Display PF Key Values -----hh:mm mm/dd/yy
COMMAND ==>                                         XDMAKEY
PF KEY VALUES:  ALTERNATE KEYS
PF13 ==> RESET
PF14 ==> OVERRIDE
PF15 ==> SWAP
PF16 ==> KEYS
PF17 ==> STORE
PF18 ==> SHOW
PF19 ==> WITHSRC
PF20 ==> SUBMIT
PF21 ==> SLISTT
PF22 ==> WITHLOAD
PF23 ==> ENTER
PF24 ==> =X

INSTRUCTIONS:  PRESS "ENTER" KEY TO PROCESS CHANGES AND DISPLAY ALTERNATE KEYS
               Enter "PREVIOUS" command to process changes and RETURN
    
```

Questions and Answers About Using DMA

Here is a list of some questions that may arise in your use of DMA, and some recommended actions to take.

What are the PF Key Values?

If you are unsure of the values assigned to PF keys, type KEYS in the command line of any screen and press the ENTER key, or press PF16.

What are the System Commands?

If you are unsure of what the system commands are, type MENU in the command line and press the ENTER key, or press PF12.

What Should I Enter On a DMA Screen?

If you are unsure of what to enter on a DMA screen, type HELP in the command line and press the ENTER key, or press PF1.

Can I Perform DMA Activities?

Contact your DBA if you have questions about your authority to perform the following activities:

- Defining a new migration
- Modifying a previously-generated parameter file
- Submitting parameters and JCL to CA IDMS Dictionary Migrator
- Copying and using another user's DMA parameter file

How Will A Migration Affect The Target Dictionary?

You may be unsure what entities are to be extracted from the source dictionary. Or, you may be unsure how a migration would affect the target dictionary. To find out, on the Specify Migration Output screen select VERIFY, MIGRATE, or AUDIT. Then, submit the parameters to CA IDMS Dictionary Migrator. By selecting VERIFY, MIGRATE, or AUDIT, you can expect CA IDMS Dictionary Migrator to perform the migration-related activities and then produce reports that display the information you need.

How Do I Use the Dictionary?

If you have questions about using the dictionary, contact your DBA and/or see the appropriate CA IDMS manuals.

First Steps in Using DMA

You must complete the steps described below any time you begin using DMA.

Access DMA

To access DMA, type the taskcode DMA on the CA IDMS/DC system prompt screen. Once you have done this, DMA displays the Select DMA Activity Screen. To return to the CA IDMS/DC system prompt screen from the DMA entry screen, press the CLEAR key.

Select DMA Activity

From the Select DMA Activity screen, you can begin to perform any of the major DMA activities. Choose *one*:

- **Define a NEW Migration?**--Type Y for yes, N for no. You can start a new migration setup by making this choice
- **See a LIST of Parameter Files?**--Type Y for yes, N for no. You can select a parameter file to modify or submit a stored parameter file or copy a parameter file from another user.
- **Start from Parameter File Name?**--Specify a parameter file--a PARMFILE--stored under your user ID.
- **Get HELP?**--Type Y to access a help screen that tells you how to answer the questions on the Select DMA Activity screen.

After you have made one choice on the screen, press the ENTER key.

```

CA IDMS/DMA Rnn.nn --- Select CA-IDMS/DMA Activity -----hh:mm mm/dd/yy
COMMAND ==>                                         XDMAWHA
      Do You Want to:
        Define a NEW Migration? ==> N (Y)es or (N)o
        See a LIST of Parameter Files? ==> N (Y)es or (N)o
        Start from Parameter File Name? ==>
          Get HELP? ==> N (Y)es or (N)o

Please Make a Choice and Press the "ENTER" key

```

Reset and Exit DMA

If you want to begin a new DMA migration setup without exiting DMA, use the RESET command; if you want to exit DMA, use the exit command (X). A migration setup terminates when you execute a RESET or an exit command.

Reset DMA

Resetting DMA permits you to:

- Erase all values entered since the last time you used the STORE function
- Continue to use DMA
- Return to the Select DMA Activity screen

To RESET DMA, type RESET in the command line and press the ENTER key, or press PF13. After using the RESET command, DMA displays the Confirm Setup RESET screen (see the next screen). This screen reminds you of the consequences of executing the RESET.

To execute the RESET function, press the ENTER key.

If you enter the RESET command by mistake, type PREVIOUS in the command line and press the ENTER key, or press PF3. DMA displays the System Commands screen.

```

CA IDMS/DMA Rnn.nn --- Confirm Setup RESET -----hh:mm mm/dd/yy
COMMAND ==>                                         XDMACFM
*****
*
*      WARNING ----- WARNING ----- WARNING      *
*
*      A RESET command has been entered. If the RESET is executed:
*
*          1) The current setup will be erased.
*
*          2) You will go to the 'Select DMA Activity' to restart the setup
*
*      Press "ENTER" to execute the RESET.
*
*      Use the "PREVIOUS" command to continue the setup.
*
*****
    
```

Exit DMA

Exiting DMA permits you to:

- Erase all values entered since the last time you used the STORE function
- Return to the CA IDMS/DC system prompt screen

To exit DMA, type X in the command line and press the ENTER key, or press PF24. When you enter the exit command, DMA displays the Confirm Session EXIT screen (see next screen). This screen reminds you of the consequences of executing the EXIT.

To execute the exit function, press the ENTER key.

If you enter the exit command by mistake, type *PREVIOUS* in the command line and press the ENTER key, or press PF3. DMA returns to the previous screen.

```
CA IDMS/DMA Rnn.nn --- Confirm Session Exit -----hh:mm mm/dd/yy
COMMAND ==> XDMAXFM
*****
*
*   WARNING ----- WARNING ----- WARNING   *
*
*
*   An EXIT command has been entered. If the EXIT is executed:   *
*
*       1) The current setup will be erased.                       *
*
*       2) You will Exit CA IDMS/Dictionary Migrator Assistant.   *
*
*   Press "ENTER" to execute the EXIT.                             *
*
*   Use the "PREVIOUS" command to continue the session.           *
*
*****
```

Common Fields on DMA Screens

Several fields are common to all DMA screens. Here are descriptions of the fields, designated in the next screen.

- **DMA Release number**--in the format Rnn.s, where nn represents the release number and s represents the subrelease number.
- **Screen title**--identifies the DMA function or identifies a list of parameter files, starting points, or system commands (System Commands).
- **Current time**--in the format hh:mm, representing the time at which the screen is displayed, in hours and minutes on a 24-hour clock.
- **Current date**--in the format mm/dd/yy, representing the current month, day, and year.
- **Command line**--the field where you can type System Commands. On some screens this field is labeled OPTION; in an OPTION field type the number of an option listed on the screen.
- **Message area**--the third line of every screen displays DMA messages (DMA00601). For a complete explanation of DMA messages, reasons for occurrence, and suggested actions, see the chapter "Messages".
- **Screen name**--the top right of the screen (XMAMNU). Unique identifier for screen.

```
CA IDMS/DMA Rnn.nn --- System Commands -----hh:mm mm/dd/yy
COMMAND ==>                                         XDMAMNU
DMA0060I MIGRATION JOB SUBMITTED
(S)elect one or more of the following functions OR enter one COMMAND:
  FILES   - LIST of PARMFILES
  STORE   - Store Setup into PARMFILE
  ENVIRON - Describe the Migration Environment
  OVERRIDE - Specify Migration Overrides
  OUTPUT  - Specify Migration Output
  START   - Specify a new STARTING POINT
  SPOVER  - Specify STARTING POINT Overrides
  SLIST   - LIST of STARTING POINTS
  WITHSRC - Include/Exclude SOURCE Types
  WITHLOAD - Include/Exclude LOAD Types
  JCL     - Edit JCL for Migration
  SHOW    - BROWSE Parameters for Migration
  RESET   - ERASE setup and RESTART
  SUBMIT  - SUBMIT current setup to CA-IDMS/Dictionary Migrator
  KEYS    - DIspay Key Values
  HELP    - Display HELP Screen
  X       - EXIT from DMA
```


DMA Activity Paths

Once you have accessed DMA, the Select DMA Activity screen is displayed. From this screen, four activities are available:

- Defining a new migration
- Seeing a list of parameter files
- Starting with a parameter file that you name
- Getting help--for the Select DMA Activity Screen

```

CA IDMS/DMA Rnn.nn --- Select CA IDMS/DMA Activity -----hh:mm mm/dd/yy
COMMAND ==>                                         XDMAWHA
      Do You Want to:
        Define a NEW Migration? ==> N (Y)es or (N)o
        See a LIST of Parameter Files? ==> N (Y)es or (N)o
        Start from Parameter File Name? ==>
          Get HELP? ==> N (Y)es or (N)o

Please Make a Choice and Press the "ENTER" key

```

Define a New Migration: Sequence of Screens

The sequence of screens for defining a new migration is displayed in the next flowchart. When you have finished entering information on a screen, to access the next screen in the sequence enter the NEXT command (or press PF2).

To begin defining a new migration, enter information on the following two screens:

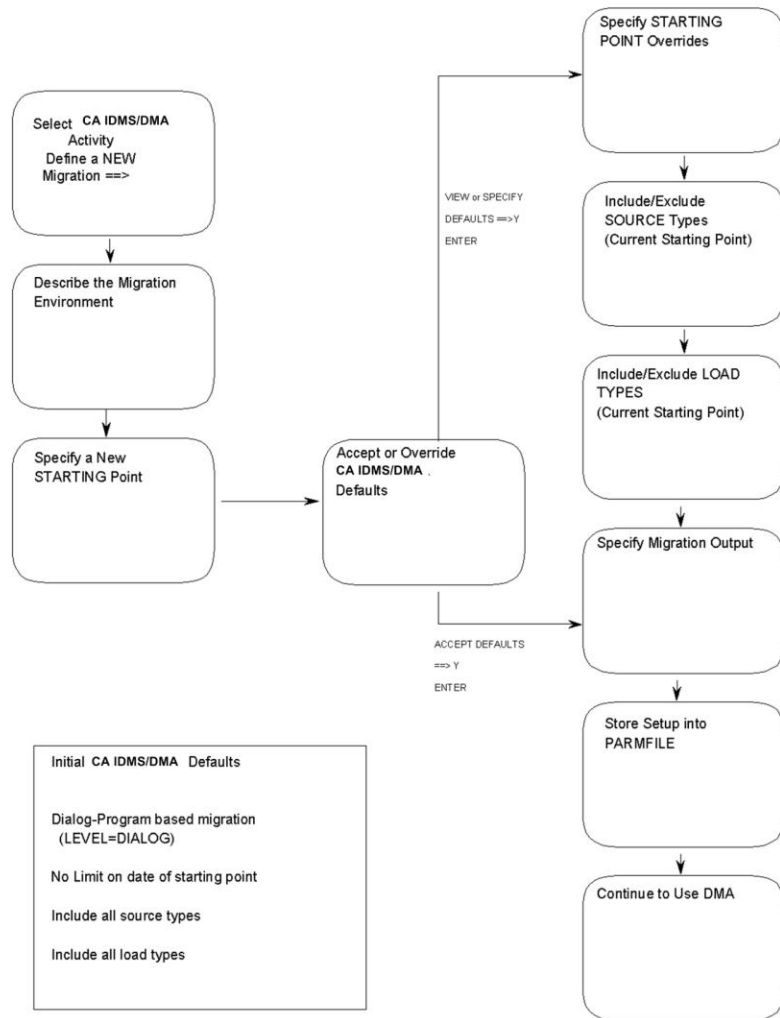
1. Describe the Migration Environment.
2. Specify a New STARTING POINT.

DMA supplies defaults governing the overall migration. After you provide source dictionary, target dictionary, and starting point values on the two screens listed above, DMA displays another screen:

3. Accept or Override DMA Defaults.

This screen lists two choices:

- Accept DMA-supplied defaults?
- View or Specify overrides of the defaults?



1. Specify Overrides of DMA-Supplied Defaults

If you choose to specify overrides to the DMA-supplied defaults, then DMA presents the following sequence of screens:

1. Specify STARTING POINT Overrides.
2. Include/Exclude SOURCE Types (for the Current STARTING POINT).
3. Include/Exclude LOAD Types (for the Current STARTING POINT).
4. Specify Migration Output.
5. Store Setup into a PARMFILE.
6. Continue to Use DMA.

2. Accept DMA-Supplied Defaults

If you accept the DMA-supplied defaults, then DMA presents the last three screens in the sequence:

1. Specify Migration Output
2. Store Setup into a PARMFILE
3. Continue to Use DMA

Define a New Migration: Overview

To define a new migration, use the sequence of screens as they are presented by DMA. Follow the instructions given below and on the following pages.

Complete First Steps Using DMA

Sign on to DMA, completing the steps described under First Steps Using DMA. At the Select DMA Activity screen, type Y to the right of the question:

Define a NEW Migration? Y

1. Describe the Migration Environment

Provide values for the source dictionary and the target dictionary on the Describe the Migration Environment screen. Other sections in this user guide refer to the *target* dictionary as the *object* dictionary.

Type the values for the source dictionary and the target dictionary in the appropriate positions.

Type `HELP` in the command line and press the `ENTER` key or press the `PF1` key to access information on the fields on the Describe the Migration Environment screen.

Establish a Setup

Even if you do not change the values presented on Describe the Migration Environment screen, if you press the `ENTER` key, you have established a migration setup. DMA has collected enough information to generate parameter statements for CA IDMS Dictionary Migrator. To use the setup, you must store it in a parmfile.

Transfer to the Next Screen

To access the next screen in defining a new migration, type `NEXT` in the command line and press the `ENTER` key, or press `PF2`.

2. Specify a New Starting Point

On the Specify a New `STARTING POINT` screen you can:

- Replace an existing starting point
- Add a new starting point
- Specify additional starting points.

Type `HELP` in the command line and press the `ENTER` key or press the `PF1` key to access information on the fields on the Specify a New `STARTING POINT` screen.

You can specify as many starting points as you need for a migration. To specify more than one starting point:

- Specify any necessary overrides for the current starting point (See the following pages)
- *STORE* (PF17) the starting point (See Part 11 for details)
- Use the *START* command (PF4) to re-access the Specify a New *STARTING POINT* screen
- Repeat the process as many times as necessary.

Transfer to the Next Screen

To access the next screen in defining a new migration, type *NEXT* in the command line and press the *ENTER* key, or press PF2.

View a List of Starting Points

To view a list of starting points:

- Type *SLIST* in the command line and press the *ENTER* key, or press PF21.

An arrow points to the current (active) starting point.

You can select another current (active) starting point or delete non-current starting points from this screen by typing *S* or *D* to the left of a starting point and pressing the *ENTER* key.

3. Accept or Override Defaults

By this time you have provided values for the source dictionary, the target dictionary, and the starting point; or, you have accepted the dictionary and starting point values collected and supplied by DMA. DMA supplies additional defaults for the overall migration. The defaults DMA supplies depend on whether:

- You are defining a new migration directly, without accessing a parameter file first.
- You selected a parameter file.
- You stored the current setup.

1. If you are familiar with the defaults supplied by DMA, and decide to accept them as they are, then type Y to the right of:
Accept DMA-supplied defaults?
2. If you want to see which specific defaults have been supplied for your current setup, or if you want to specify overrides of the DMA-supplied defaults, type Y to the right of:
View or Specify overrides of defaults?
3. If you want a general description of the types of defaults supplied for your current setup, type Y to the right of:
Obtain more information about your options?

Transfer to the Next Screen

After typing Y for *one* choice, press the ENTER key to go to the next screen in defining a new migration.

4. Specify Starting Point Overrides

Source dictionary values and three starting point overrides, which govern the extraction of entities from the source dictionary, are displayed on the Specify STARTING POINT Overrides screen.

Consider the source dictionary values and each override separately. You can specify information for one, any, or all of the overrides displayed on this screen.

Type HELP in the command line and press the ENTER key or press the PF1 key to access information on the Specify a New STARTING POINT screen.

Transfer to the Next Screen

To access the next screen in defining a new migration, type NEXT in the command line and press the ENTER key, or press PF2.

Parameter Reference

EXTRACT--IDSOURCE, LEVEL, DATE, DOMAIN, DICTIONARY

5. Include/Exclude Source Types (Current Starting Point)

When using DMA, you can limit the entities for migration by excluding some or all source types associated with the current starting point. You can also choose to retain all source types associated with the current starting point.

Type `HELP` in the command line and press the `ENTER` key or press the `PF1` key to access information on the Include/Exclude `SOURCE` Types screen.

Transfer to the Next Screen

To access the next screen in defining a new migration, type `NEXT` in the command line and press the `ENTER` key, or press `PF2`.

Parameter Reference

`EXTRACT--EXSOURCE`

6. Include/Exclude Load Types (Current Starting Point)

When using DMA, you can limit the entities for migration by excluding some or all load types associated with the current starting point. You can also choose to retain all load types associated with the current starting point.

Type `HELP` in the command line and press the `ENTER` key or press the `PF1` key to access information on the Include/Exclude `LOAD` Types screen.

Transfer to the Next Screen

To access the next screen in defining a new migration, type `NEXT` in the command line and press the `ENTER` key, or press `PF2`.

Parameter Reference

`EXTRACT--EXLOADS`

7. Obtain CA IDMS Dictionary Migrator Reports

When you submit parameters to CA IDMS Dictionary Migrator, you are *not* updating the target dictionary. Instead, CA IDMS Dictionary Migrator performs migration activities and produces reports that show the results of each activity. You obtain information about the requested entities located in the source and target dictionaries.

If you provided information about the target dictionary only on the Describe the Migration Environment screen, you can select the output for the IMPORT run. If you provided information about the intended source dictionary only, you can select output for the EXPORT run.

Note: For more information about CA IDMS Dictionary Migrator reports, see the chapter "Reports."

Specify Migration Output

Type `HELP` in the command line and press the `ENTER` key or press the `PF1` key to access information on specifying the migration output and detailed information about each type of report that CA-IDMS/Dictionary Migrator produces.

Transfer to the Next Screen

To access the next screen in defining a new migration, type `NEXT` in the command line and press the `ENTER` key, or press `PF2`.

Parameter Reference

`PROCESS--RUN`

`NOXREF`

`CLIST CLISTVERSION`

8. Store Setup into a PARMFILE

At any time after you have established a setup by:

- Selecting a DMA activity
- Accessing any of the possible subsequent screens
- Pressing the ENTER key,

you can store your current DMA setup.

A user would usually store a current setup after supplying to DMA all source dictionary, target dictionary, and starting point values for a migration.

What You Can Store

From the Store Setup into PARMFILE screen you can use DMA to store:

- The source and target dictionary and starting point values
- Parameters for CA IDMS Dictionary Migrator that reflect the source and target dictionary and starting point values
- JCL to run CA IDMS Dictionary Migrator
- User-defined messages inserted into the JCL.

How to Store

To store the current DMA setup into a PARMFILE (parameter file), type appropriate responses in the fields identified in on the Store Setup into PARMFILE screen.

Type `HELP` in the command line and press the ENTER key or press the PF1 key to access information on the Store Setup into PARMFILE screen.

Transfer to the Next Screen

To access the next screen in defining a new migration, type `NEXT` in the command line and press the ENTER key, or press PF2.

9. Continue to Use DMA

The Continue to Use DMA screen gives control over what happens after a new migration is defined. At this point, you can continue to use DMA by selecting one of the major DMA functions, or you can exit.

Type Y to the right of the *one* function you want to use and press the ENTER key.

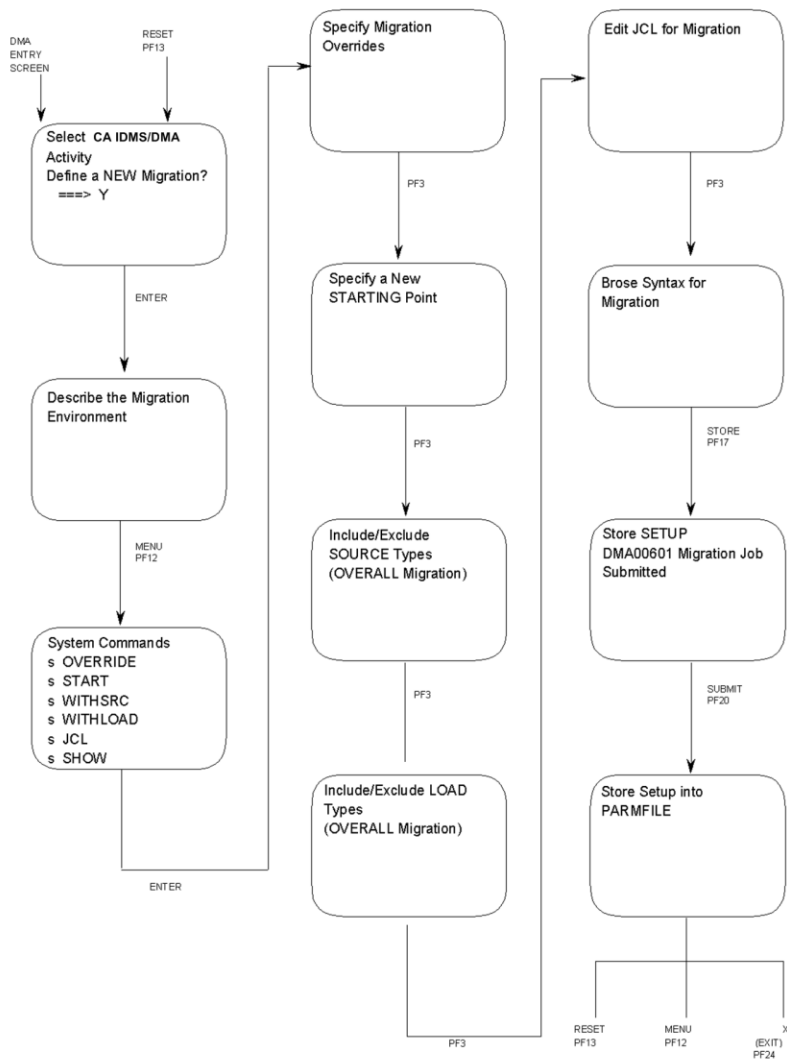
Type HELP in the command line and press the ENTER key or press the PF1 key to access information about the DMA processing options presented on the Continue to Use DMA screen.

Global Overrides

You may want to set up global conditions for most or all of the entities in the migration. The screens for global overrides differ from the screens for starting point overrides in that the global override screens all have the words "OVERALL migration" or simply "migration" either in the title or in the first line of text.

An example of a typical sequence of screens and commands for setting up global conditions is shown in the next flowchart. Many of the screens in the sequence are described on the following pages. You can find descriptions of the other screens in the section Define a New Migration: Sequence of Screens.

In the sequence shown, the user has specified most of the sequence on the System Commands screen, by typing S to the left of each command in order. Then PF3 automatically takes the display to the next screen in the sequence. Each screen can be accessed individually if the user enters a command when ready for a screen.



DMA Provides a Default Starting Point

If you select CA IDMS Dictionary Migrator output without specifying a starting point, DMA supplies a default value for the starting point: user ID entered when you accessed DMA.

Use a Series of Commands from the Menu

The System Commands menu allows you to select one command or several commands in a series.

In the example shown in the next screen, the user wants a series of six screens. Each screen will be displayed in order, from top to bottom, when the user presses PF3 after finishing with the current screen.

```
CA IDMS/DMA Rnn.nn --- System Commands -----hh:mm mm/dd/yy
COMMAND ==>                                         XDMAMNU
(S)elect one or more of the following functions OR enter one COMMAND:
  FILES      - LIST of PARMFILES
  STORE      - Store Setup into PARMFILE
  ENVIRON    - Describe the Migration Environment
  S OVERRIDE - Specify Migration Overrides
  OUTPUT     - Specify Migration Output
  S START    - Specify a new STARTING POINT
  SPOVER     - Specify STARTING POINT Overrides
  SLIST      - LIST of STARTING POINTS
  S WITHSRC  - Include/Exclude SOURCE Types
  S WITHLOAD - Include/Exclude LOAD Types
  S JCL      - Edit JCL for Migration
  S SHOW     - BROWSE Parameters for Migration
  RESET      - ERASE setup and RESTART
  SUBMIT     - SUBMIT current setup to CA-IDMS/Dictionary Migrator
  KEYS      - Display Key Values
  HELP      - Display HELP Screen
  X         - EXIT from DMA
```

Specify Migration Overrides

Several migration overrides, which govern the extraction of entities from the source dictionary, are displayed on the Specify Migration Overrides screen. Consider each override separately. You can specify values for one, any, or all of the overrides displayed on this screen.

The values you enter on the Specify Migration Overrides screen are retained as default values unless you enter other values on the Specify STARTING POINT Overrides screen.

Type **HELP** in the command line and press the **ENTER** key or press the **PF1** key to access information about the Specify Migration Overrides screen.

Transfer to the Next Screen

To access the next screen in the command sequence, type **N[EXT]** in the command line and press the **ENTER** key, or press **PF2**.

Parameter Reference

PROCESS--LEVEL, DATE, VERSION, NEWVERSION, COBOLFORMAT, DISCONNECT

Limit Entities for Overall Migration

Using DMA, you can limit the entities for migration by excluding some or all source types and load types. You can also retain all source and load types associated with the overall migration.

Include/Exclude SOURCE Types (Overall Migration)

The values entered on the Include/ExcludeSOURCE Types screen for the *OVERALL Migration* are retained as default values unless you enter other values on the Include/ExcludeSOURCE Types screen for the *Current STARTING POINT*.

This permits you to specify just one time the starting point information that does not change for the migration, and to enter only the entity values that change for each of the starting points you are using.

Type **HELP** in the command line and press the **ENTER** key or press the **PF1** key to access information about the Include/ExcludeSOURCE Types Screen.

Use the Command SWAP

To switch back and forth between the values specified for the Overall Migration and the values specified for the *Current* Starting Point, type **SWAP** in the command line and press the **ENTER** key, or press **PF15**.

Transfer to the Next Screen

To access the next screen in defining a new migration, type **N[EXT]** in the command line and press the **ENTER** key, or press **PF2**.

Parameter Reference

PROCESS—EXSOURCE

Include/Exclude LOAD Types (Overall Migration)

By including some or all load types, and by excluding all source Types, you can set up a run-time environment without a dictionary. To update this type of environment you need to again perform a similar migration.

The values entered on the Include/Exclude LOAD Types screen for *OVERALL Migration* are become default values unless you enter values on the Include/Exclude LOAD Types screen for the *Current STARTING POINT*.

This permits you to specify just one time the starting point information that does not change for the migration, and to enter only the entity values that change for each of the starting points you are using.

Type `HELP` in the command line and press the `ENTER` key or press `PF1` to access information about the Include/Exclude LOAD Types screen.

Use the Command SWAP

To switch back and forth between the values specified for the Overall Migration and the values specified for the *Current Starting Point*, type `SWAP` in the command line and press the `ENTER` key, or press `PF15`.

Transfer to the Next Screen

To access the next screen in defining a new migration, type `NEXT` in the command line and press the `ENTER` key, or press `PF2`.

Parameter Reference

PROCESS—EXLOADS

View the Parameter Statements

The parameter statements generated by the typical sequence of screens shown on the preceding pages are available for viewing on the Browse - Syntax for Migration screen.

An example of JCL is shown later in the DMA Parameters section.

To change the parameter statements, follow the procedure described in [Modifying a DMA Parameter File](#) (see page 144).

STORE Before Submitting

Be sure to *STORE* (PF17) the setup into a parameter file before trying to submit it.

Select a DMA Parameter File

A DMA parameter file contains parameter statements and JCL.

To *select* a DMA parameter file:

1. Complete FIRST STEPS IN USING DMA.
2. Type Y to the right of
See a List of Parameter Files?
and press the ENTER key.

Alternate action: Enter the *FILES* command from any screen.

DMA displays the list of parameter files stored under your user ID. Use the UP and DOWN commands if the list of files is longer than one screen.
3. Type S to the left of the parameter file with the parameters and JCL you want to examine. Make sure that the correct user ID is displayed to the right of For USER ID
4. Press the ENTER key.

Select Multiple DMA Parameter Files

To select more than one DMA parameter file, follow the steps above, but type S to the left of *each* of the parameter files you want to examine.

After you have completed your examination of a parameter file, type *PREVIOUS* on the command line and press the ENTER key, or press PF3. DMA displays the next parameter file you selected.

Specify a DMA Parameter File

To *specify* a DMA parameter file:

1. Complete FIRST STEPS IN USING DMA.
2. Type a parameter file name to the right of:
Start from Parameter FileName?
3. Press the ENTER key.

Modify a DMA Parameter File

You can *modify* all parameter files stored under your DMA user ID. You can *examine* all parameter files stored under other users' IDs. You cannot modify or delete other users' parameter files, but, as explained on following pages, if you have authority, you can copy them, modify the copies, and store them as parmfiles under your user ID.

Steps to Modify a DMA Parameter File

To modify a DMA parameter file, first select or specify a parameter file, as described on the previous pages. (If you are already in a DMA setup, you can use the *SHOW* command.) Then, follow the appropriate steps described below.

Modify Dictionary/Starting Point Values

To modify the dictionary and/or starting point values, use any of the following commands. Select a command to display an appropriate specification screen.

- **ENV[IRON] PF10** — Describe the Migration Environment
- **STA[RT] PF4** — Specify a new STARTING POINT
- **SP[OVER] PF5** — Specify STARTING POINT Overrides
- **WITHS[RC] PF19** — Include/Exclude SOURCE Types
- **WITHL[OAD] PF22** — Include/Exclude LOAD Types
- **Slist (PF21)** — List of STARTING POINTS

Modify Parameters

To modify parameters:

- Modify the dictionary and/or starting point values, as described above
- Store the parameter file. (STORE or PF17)

In this case, when you store the parameter file, DMA re-converts the values into a new set of parameter statements.

```

BROWSE - SYNTAX FOR MIGRATION                                COLUMNS 001 079
COMMAND ==>                                                SCROLL ==> PAGE
*** TOP OF DATA ***** CA-IDMS/DMA *****
PROCESS
  RUN=MIGRATE
  CVSOURCE=(015,233)
  CVOBJECT=(015,233)
  DICTIONARY=DICT1,
  OBJDICTIONARY=DICT2,
  IDSOURCE=
  IDOBJECT=
  LEVEL=DIALOG
  CXSOURCE=(COMMENT,MAP),
  DATE=(mddy),
  NEWVERSION=0005,
  VERSION=0004,
EXTRACT
  PROGRAM=DIALOG1,
  LEVEL=DIALOG,
  VERSION=0002
*** BOTTOM OF DATA ***** CA-IDMS/DMA *****

```

Modify JCL

Before submitting JCL and parameters, you should modify the CA IDMS Dictionary Migrator JCL to reflect conditions at your site. See Operations for more information.

Be sure that there is a job card and that sample parameters are deleted in USMXTRCT.

If you already have CA IDMS Dictionary Migrator JCL prepared, you can copy it to DMA at installation time.

Display JCL

To display the JCL for modification or insertion of comments:

- Type JCL in the command line and press the ENTER key, or press PF11.

The EDIT JCL screen is displayed.

Then use the Dictionary Migrator Assistant JCL Editing Commands to edit the JCL. The editing commands are described in the DMA online documentation. To access editing HELP, while viewing the Edit JCL screen enter the HELP command or press PF1.

Save Modifications

To save any modifications, you *must* use the STORE command or press PF17.

```

EDIT --- JCL FOR MIGRATION                                COLUMNS 001 079
COMMAND ==>                                             SCROLL ==> PAGE
*** TOP OF DATA ***** CA-IDMS/DMA *****
000001 //XDMEXEC JOB (ARG,4083S),CLASS=A,MSGCLASS=R,TYPRUN=SCAN
000002 /*ROUTE PRINT IRCVM.ARG
000003 /*
000004 /*
000005 /*      MODIFY THE FOLLOWING SAMPLE JCL. SEE THE OPERATIONS SECTION OF
000006 /*      THE CA IDMS/Dictionary Migrator USER GUIDE FOR DETAILS.
000007 /*      SUBSTITUTE INFORMATION SUITABLE TO YOUR SITE'S STANDARDS FOR THE
000008 /*      FOLLOWING VARIABLES:
000009 /*
000010 /*      CA LOAD LIBRARY DSN: REPLACE 'YOUR.LOADLIB' WITH THE
000011 /*      DATASET NAME OF THE LIBRARY IN WHICH CA-IDMS/DICTIONARY
000012 /*      MIGRATOR AND THE SPECIAL VERSIONS OF THE CA-IDMS UTILITIES
000013 /*      RESIDE. IF THE BLOCK SIZE OF YOUR.LOADLIB IS LESS THAN THAT
000014 /*      OF THE IDMS.LOADLIB, SPECIFY THE LARGER BLOCK SIZE IN THE
000015 /*      DCB PARAMETER OF THE FIRST LIBRARY IN THE STEPLIB STATEMENT.
000016 /*
000017 /*      CA-IDMS LOAD LIBRARY DSN: REPLACE 'IDMS.LOADLIB' WITH THE
000018 /*      DATASET NAME OF THE LIBRARY IN WHICH THE IDMS MODULES RESIDE.
000019 /*
000020 /*      SYNTAX FILES DSN: THESE FILES WERE ALLOCATED IN A SEPARATE
*** BOTTOM OF DATA ***** CA-IDMS/DMA *****
    
```

Change the Default JCL

To store the modified JCL in the DEFAULT parameter file for your user ID, follow these steps:

1. Complete the modification of JCL.
2. STORE (PF17) as parmfile DEFAULT.
3. Replace the existing parameter file by typing Y for the answer to the "Replace" question.

Copy a DMA Parameter File

Ask your DBA if you have the authority to copy another user's parameter files to your list of PARMFILES. If you have the authority:

1. Access the List of PARMFILES screen by one of the following actions:
 - a. Enter the *FILES* command or press PF6.
 - b. Select FILES on the System Commands menu.
 - c. Answer Y to the "See a list of PARMFILES?" question on the Select DMA Activity screen.
2. After For USER-ID, type the user ID of the user whose parameter file you want to copy.
3. Press the ENTER key. A list of parmfiles for the specified user ID is displayed.
4. Type S to the left of the parameter file you want to copy.
5. Press the ENTER key.
6. When the data is displayed, enter the *STO[RE]* command or press PF17.
7. When the Store Setup into PARMFILE screen is displayed, type a new name for the parameter file.
8. Press the ENTER key.

A copy of the parameter file is now stored under your user ID. That copy can be modified.

Submit Parameters and JCL Online

To submit JCL for running CA-IDMS/Dictionary Migrator with the parameter statements generated by DMA concatenated:

1. Select or specify a DMA parameter file.
2. Type *SUBMIT* in the command line and press the ENTER key, or press PF20.
3. The *SUBMIT* command may be entered from any DMA screen, except for the 'JCL' and 'SHOW' screens.

DMA takes the parameter statements and JCL contained in the parameter file and submits the job to your operating system reader.

File Allocations

Allocations for a VSAM work file and syntax files must either be included in the JCL or have been previously allocated. These allocations are described in Part 1 and Part 2 of the model JCL in Operations. The file allocations may be included in the JCL installed as the DMA default JCL.

Submit Parameters and JCL from the Batch Environment

To submit parameter statements generated by DMA to CA IDMS/Dictionary Migrator, and the job containing JCL to run CA IDMS/Dictionary Migrator, from the batch environment:

1. Create a sequential file containing the parameters generated by DMA. To do this:
 - a. Access source library member XDMBSYN (z/OS) or XDMBSYN.S (z/VSE). This member was downloaded into *your.srclib* at installation.
 - b. Replace the variables with appropriate values. To identify the appropriate values, see the key found in XDMBSYN.
2. Run the job contained in USMBSYN.
3. In the JCL you use to run CA IDMS/Dictionary Migrator in the batch environment:
 - a. Replace the statement:

```
//MIGPARM DD *
```

with:

```
//MIGPARM DD your.migrator.syntax
```

where *your.migrator.syntax* represents the name of the sequential file into which the parameters generated by DMA are to be copied.
 - b. In the JCL, after the above statement, delete all migration parameter statements.
4. Submit the JCL to run CA IDMS/Dictionary Migrator.

Copy CA IDMS/Dictionary Migrator JCL to DMA

You must copy CA IDMS/Dictionary Migrator JCL to DMA before you can use DMA.

After you have installed DMA, you may want to use JCL other than the assigned default JCL. You can do so by using the JCL batch upload utility to upload JCL from a sequential file to a DMA parameter file (PARMFILE). The JCL batch upload utility, contained in XDMBJCL, is the same utility that was used to upload the assigned default JCL when DMA was installed.

To copy JCL from a sequential file to a DMA parameter file:

1. Access source library member XDMBJCL (z/OS) or XDMBJCLS (z/VSE). This member was downloaded into *your.srclib* at installation.
2. Replace the indicated variables with appropriate values. The key to the variables is also contained in XDMBJCL.
3. Replace USER=DEFAULT with USER=*user-id*, where *user-id* is a 1- to 32-character ID. If blanks are embedded in the user ID, quotation marks must be placed around it.
4. Replace PARMFILE=DEFAULT with PARMFILE=*parmfile-id******, where *parmfile-id* represents a 1- to 8-character name for the PARMFILE.
5. Run the job contained in XDMBJCL.

Examine the output report for error messages. If the job ran successfully, the JCL from the sequential file is stored in the specified parmfile.

Note: The migrator z/OS JCL to be uploaded should contain //MIGPARAM DD*. The JCL must not contain any migrator parameter statements.

Chapter 7: Messages

This section contains the following topics:

[Introduction](#) (see page 151)

[Messages Generated by CA IDMS Dictionary Migrator](#) (see page 152)

[Severity Codes](#) (see page 152)

[Conditions](#) (see page 153)

[Using CA IDMS Dictionary Migrator Reports](#) (see page 153)

Introduction

This section lists all messages generated by CA IDMS Dictionary Migrator. Included are the codes for messages, reasons for their occurrences, and suggestions for appropriate remedial actions.

CA IDMS Dictionary Migrator produces messages that state that a migrated entity is in use by other entities that are not being migrated. If this is the case, CA IDMS Dictionary Migrator produces a message (VALD219) that lists the related entity. CA IDMS Dictionary Migrator also produces syntax that saves the current definition of the entity from the target dictionary. That definition is readded to the dictionary as part of the upload process after the migrated entity is updated. This updating is called forced regeneration.

Although this process does not allow the migrated entity to be updated, compatibility of the migrated entity and related entity is not assured. You must verify that the migrated structure is compatible with the entity in the target dictionary.

Another situation is a migrated entity that is related to another entity in the extract set, but the related entity is not being migrated because of selection criteria (DATE, DOMAIN, or CHANGEONLY). This results in a VALD219 message.

The message indicates that the relationship area exists. It does not describe exactly what the relationship is, or whether the migrated entity is compatible with the target dictionary entity. You should verify the compatibility, or change the migration strategy to where the relationship is not an issue.

After completing CA IDMS Dictionary Migrator upload, or a combined migrator/upload step, review the output. What is migrated has an effect on the target dictionary.

The following messages should be investigated:

- **VALD204**—Entity is related to something that is not migrated and it is affected by the migration.
- **VALD207**—You are moving load modules, do not intend to regenerate, and there is a conflict in the date/time stamp between the entities listed.
- **VALD219**—The entity was regenerated and changed, but was not migrated.

Messages Generated by CA IDMS Dictionary Migrator

All CA IDMS Dictionary Migrator messages are preceded by a unique eight-character code. The message code is in xxxnnns format, where xxxx is the ID of the module issuing the message, nnn is the message number for the message within the module, and s is the severity code for the message. The severity code is either **I**, **W**, **E**, **F**, or **C**. This chapter lists for each message the code, the text, and an explanation.

Severity Codes

Some messages can have their severity altered by the user. The severity shown here is for the unaltered product.

- **Informative**--The severity code **I** indicates an informative message. Informative messages are for your information only; no remedial action is required.
- **Warning**--The severity code **W** indicates a warning. Warning messages report conflicting data or various processing conditions.
- **Error**—The severity code **E** indicates an error. When CA IDMS Dictionary Migrator encounters an error condition, an error message appears on your screen. Error messages report erroneous and conflicting data that require action.
- **Fatal**—The severity code **F** indicates an error from which processing cannot recover. In most cases an OC1 (Operation Exception) abend is forced.
- **Critical**—A user invoked severity code that causes processing to stop at the end of the current "phase." In z/OS a return code of 8 is set.

Conditions

Different conditions cause varying degrees of impact. Different sites view the impact with different degrees of warning. Each site should determine what should be investigated.

When an entity that already exists in the target dictionary is migrated, the following conditions arise with regard to other entities in the target dictionary:

- **Condition 1**—The migrated entity is related to no other entities.
- **Condition 2**—The migrated entity is related to other entities.
- **Condition 3**—The migrated entity is related to other entities, one or more of which are not currently being migrated.

Conditions 1 and 2 do not cause any repercussions on the other entities because they are not related, or because they are also being migrated in the existing target entity and are being updated with the source definition.

It is condition three where one or more of the problems can arise:

- The migrated entity cannot be introduced to the target dictionary as long as the related entity is associated
- The migrated entity is structurally different from what currently exists in the target dictionary
 - The new structure is not compatible with the related entity
 - The new structure is compatible with the related entity
- The migrated entity is a documentation entity or differs only with respect to documentation from what is in the target dictionary.

Using CA IDMS Dictionary Migrator Reports

Messages are printed on four CA IDMS Dictionary Migrator reports in the sequence in which they are generated.

The Parameter Verification Report contains parameter-related messages in a two-column format. The left column contains the message code and text; the right column shows user input. Part of the information in the right column may be underscored by asterisks (*****) to show the precise location of the error. Information may also be underscored by a series of Xs (XXXXX) to indicate portions of the parameter that were not checked because of a previous error.

Processing messages are also listed on the following reports:

- Entity Discrepancy Report
- Source Dictionary Verification Report
- Syntax Production Report
- Syntax Files Display Report
- Catalog Navigation Report

These reports list messages in the right column. Entity names, types, and versions appear to the left of many of the messages.

For more information and samples of CA IDMS Dictionary Migrator reports, see the chapter "Reports".

CALL901E PARAMETER LIST ERROR - PGM = module-name

Reason:

The module could not be called because of an internal problem.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

CALL902E NOT FOUND IN LIBRARY - PGM = module-name

Reason:

The module cannot be located in the available load or core-image libraries.

Action:

Locate the load or core-image library that contains the module and make the library available to the job.

CALL903E NOT ENOUGH STORAGE - PGM = module-name

Reason:

The module cannot be loaded due to insufficient GETVIS area.

Action:

To increase the GETVIS area:

1. Allocate a smaller amount of space for the problem program area, using the SIZE= parameter; and, if necessary,
2. Use a larger partition.
3. Resubmit the job.

CALL904E ANCHOR TABLE LOAD FAILURE - PGM=GSSANKR

Reason:

An internal error has been detected.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

CALL905E UNANTICIPATED RETURN CODE = n PGM = module-name

Reason:

An internal error has been detected.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0001E COMMAND "x" IS NOT ACTIVE FOR THIS FUNCTION

Reason:

A command was entered that is inactive for the current function, or is a command that DMA does not recognize.

Action:

Enter a command that is active for the current function.

DMA0002E THE PF KEY PRESSED IS NOT ACTIVE FOR THIS FUNCTION

Reason:

A PF key was pressed. However, the PF key is associated with a command that is inactive for the current function.

Action:

Press a PF key that is associated with a command that is active for the current function.

DMA0003E SELECT OPTION "x" IS NOT VALID FOR THIS FUNCTION

Reason:

An invalid select code was entered.

Action:

Enter a correct select code, and press the ENTER key.

DMA0004E AN ERROR OCCURRED DURING RECOVERY PROCEDURES

Reason:

An internal error has been detected; DMA cannot be reaccessed through normal procedures.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

To reaccess DMA, choose one:

- First log off your terminal, log on again, and then on the CA IDMS/DC system prompt screen type **DMA**.
- Recycle the CV.

DMA0005E ENTER USERID AND PASSWORD TO CONTINUE

Reason:

A user ID and/or password were not correctly entered.

Action:

Correctly enter a user ID and password.

DMA0006E A SECURITY VIOLATION HAS OCCURRED - return code

Reason: A security violation has occurred; the associated return-code points to the exact reason for the security violation.

Action: Choose one:

- Obtain access to the function through the ACS security screen
- Select another function.

DMA0007E KEY SUBSTITUTION DISABLED UNTIL SIGNON IS COMPLETED

Reason:

The PF keys cannot be used until the sign-on is completed.

Action:

Complete sign-on.

DMA0008E UNABLE TO TRANSFER TO x

Reason:

A command was entered in an attempt to access a DMA function that is not accessible from the current function.

Action:

Enter a command to access a function that is accessible from the current function.

DMA0009E SUBROUTINE ERROR - x

Reason:

A subroutine error, indicated by message text x, has occurred.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0010E THE HIGHLIGHTED FIELDS CONTAIN AN INVALID VALUE

Reason:

One or more incorrect values were entered on a DMA screen.

Action:

Enter correct values and press ENTER.

DMA0011E PLEASE MAKE A SELECTION FROM THE INDICATED OPTIONS

Reason:

An entry required on the DMA screen has not been made.

Action:

Provide the required entry.

DMA0012E HIGHLIGHTED FIELD VALUES ARE IN CONFLICT - PLEASE RESOLVE AND RESUBMIT REQUEST

Reason:

More than one value was entered on a DMA screen when only one is permitted.

Action:

Enter only one value on the DMA screen.

DMA0013E DEFAULT PARMFILE FOR USER x FILE-ID y MISSING

Reason:

The user's default parameter file name (ID) is other than DEFAULT. DMA attempted to find, but could not locate the default parameter file.

Action:

Choose one:

- Rename the user's default parameter file to DEFAULT
- Add, for the user, another parameter file to act as the default parameter file, and name it DEFAULT.

DMA0014E THE TRANSFER FACILITY IS DISABLED FOR THIS FUNCTION

Reason:

A command to transfer to another DMA function was entered. However, the current function does not permit transferring to another function

Action:

Enter a command that is active for the current function.

DMA0015E INVALID SEQUENCE NUMBER x ENCOUNTERED IN DMA DATABASE

Reason:

The contents of a DMA parameter file (PARMFILE) structure are invalid.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0016E NO PROCESS PARAMETER FOR PARMFILE

Reason:

The contents of a DMA parameter file (PARMFILE) structure are invalid.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0017E NO EXTRACT PARAMETER FOR PARMFILE

Reason:

The contents of a DMA parameter file(PARMFILE) structure are invalid.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0018I DEFAULT PARAMETER FILE ESTABLISHED

Reason:

A default parameter file was successfully created.

Action:

None.

DMA0019E "x" ERROR - INVALID PARM LIST

Reason:

An ESAM/EDITOR interface call failed.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0020E ESAMVWV - INVALID PARAMETER LIST

Reason:

An ESAM/EDITOR interface call failed.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0021E ESAMVWV - INVALID PARAMETER LIST

Reason:

An ESAM/EDITOR interface call failed.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0022E ESAMVWV - I/O ERROR VVVV OCCURRED

Reason:

An ESAM/EDITOR interface call failed.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0023E ESAMVWV - UNEXPECTED RETURN CODE WHILE CREATING THE SOURCE TEXT AREA

Reason:

An ESAM/EDITOR interface call failed.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0024E ESAMVWV - END OF FILE REACHED (BEYOND BOTTOM)

Reason:

An ESAM/EDITOR interface call failed.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0025E INVALID ATTRIBUTE VALUE OF x ENCOUNTERED IN PARMFILE y

Reason:

The contents of a DMA parameter file (PARMFILE) structure are invalid.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0026I RECOVERY WAS SUCCESSFUL

Reason:

A previously aborted session was successfully restarted.

Action:

None.

DMA0027E BAD RETURN OF x FROM KEY HANDLER

Reason:

The PF key handler routine returned an invalid code, indicated by x.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0028E FUNCTION x HAS NO NEXT FUNCTION

Reason:

The command NEXT was entered. However, it was entered for a function that has no naturally occurring function following it, that is, no NEXT function.

Action:

Enter another command.

DMA0029I SESSION VALUES SUCCESSFULLY UPDATED

Reason:

Correct values were entered at a DMA screen, the user pressed ENTER or NEXT, and the current session acquired the values entered.

Action:

None.

DMA0030E ONLY ONE OF THE HIGHLIGHTED VALUES MAY BE (Y)ES

Reason:

Y for yes was entered for more than one of the DMA screen options.

Action:

Enter Y for yes for ONE option.

DMA0031E ONE OF THE HIGHLIGHTED VALUES MUST BE (Y)ES

Reason:

N for no and/or blanks were entered for all of the DMA screen options.

Action:

Enter Y for yes for ONE option.

DMA0032E HIGHLIGHTED DATE COMPONENTS ARE INVALID

Reason:

A date field entered does not match the standard date criteria.

Action:

Enter a day within the range for the specified month, and enter a month within the range 01 to 12.

DMA0033E USER x HAS NO PARMFILES

Reason:

A request was made to see a list of parameter files (PARMFILES) for a user who has no parameter files.

Action:

Choose one:

- To see a list of parameter files, enter a user ID that has parameter files assigned to it
- Enter another command to access another function.

DMA0034E LINE COMMAND "x" NOT VALID FOR THIS FUNCTION

Reason:

A line command was entered in a selection list. However, the line command that was entered cannot be used on this screen.

Action:

Enter a line command that can be used on this screen.

DMA0035I DELETE OF PARMFILE x CANCELED BY USER

Reason:

Upon viewing the Confirm DELETE of PARMFILE, the user terminated the deletion by using the PREVIOUS command.

Action:

None.

DMA0036E NO SESSION HAS BEEN ESTABLISHED

Reason:

A command has been entered to access a function that requires an established setup. However, a setup has not been established.

Action:

Establish a setup. You can establish a setup by providing DMA with the minimal values required for DMA to generate a parameter file. From the Select a DMA Activity screen, select or specify a parameter file, or else type Y after Define a new Migration? Press ENTER. Choose one:

- If you are at the Describe the Migration Environment screen, press ENTER
- If you specified a parameter file, press ENTER
- From the List of PARMFILES, select a parameter file; once you have accessed the parameter file, press ENTER.

DMA0037E ANOTHER USER'S PARMFILES MAY NOT BE DELETED

Reason:

A D for delete was entered to the left of a parameter file belonging to a user other than the signed-on user.

Action:

Remove the D for delete.

DMA0038I PLEASE SELECT A COMMAND

Reason:

The ENTER command was used on the System Commands screen. However, a command was not typed in the command line, nor was an S for select typed to the left of the commands in the command list.

Action:

Choose one:

- Type a command, other than ENTER, in the command line
- Type S for select to the left of one or more commands in the command list.

Then, press Enter.

DMA0039I SETUP TERMINATED

Reason:

Upon viewing the Confirm Setup RESET screen, the user pressed ENTER, erasing all non-stored values and returning to the Select DMA Activity screen.

Action:

None.

DMA0040I SETUP VALUES RETAINED

Reason:

Upon viewing the Confirm Setup RESET screen, the user used the PREVIOUS command. Consequently, DMA returned the user to the System Commands menu.

Action:

None.

DMA0041E ATTRIBUTE x MISSING FROM SYNTAX FRAGMENT TABLE

Reason:

A severe error occurred as DMA attempted to generate parameters.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0042E STORE FAILURE - CLUSTER FOR USER x MISSING

Reason:

A severe error occurred as DMA attempted to generate parameters.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0043E USER x PARMFILE y EXISTS - SPECIFY "YES" FOR REPLACE

Reason:

From the Store into a PARMFILE screen, the user attempted to store a session. And, the parameter file name typed in at the right of 'Save as PARMFILE' is the same as another parameter file name. However, N for no was typed to the right of the 'Replace Existing PARMFILE'.

Action

Choose one:

- Type in a different parameter file name
- Type in Y for yes at the right of 'Replace Existing PARMFILE'.

Note: If Y is entered for the replace option, the contents of the parameter file that is replaced are erased.

DMA0044E STORE FAILED - SYNTAX QUEUE IS MISSING**Reason:**

A severe error occurred as DMA attempted to generate parameters.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0045E STORE FAILURE - JCL QUEUE IS MISSING**Reason:**

A severe error occurred as DMA attempted to generate parameters.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0046I PARMFILE x STORED FOR USER y**Reason:**

The parameter file (PARMFILE) named in the message was successfully stored.

Action:

None.

DMA0047E NO JCL PROVIDED FOR THE UPLOAD**Reason:**

DMA attempted to upload CA IDMS Dictionary Migrator JCL to DMA. However, no JCL was available.

Action:

Provide CA IDMS Dictionary Migrator JCL to the syntax batch upload utility in file XDMPJCL.

DMA0048I JCL UPLOADED FOR USER x PARMFILE y

Reason:

DMA successfully uploaded JCL to the parameter file (PARMFILE) named in the message.

Action:

None.

DMA0049E NO DMA CLUSTER ESTABLISHED FOR USER x

Reason:

An attempt was made from the batch environment to use parameters generated by DMA. However, no parameter file (PARMFILE) was available for the user named in the message.

Action:

Sign on to DMA.

1. Use DMA to define a new migration. For more information see the section *Defining a New Migration: Sequence of Screens*.

DMA0050E PARMFILE x MISSING FOR USER y

Reason:

An attempt was made from the batch environment to use parameters generated by DMA. However, no parameter file (PARMFILE) by the name given in this message was available.

Action:

Choose one:

- Specify another parameter filename
- Create a parameter file with the name given in the message. To do this:
 1. Sign on to DMA.
 2. Use DMA to define a new migration. For more information see the section *Defining a New Migration: Sequence of Screens*.

DMA0051E DEFAULT JCL HAS NOT BEEN LOADED

Reason:

When DMA was installed, the JCL batch upload utility XDMPJCL was not executed. Therefore, CA IDMS Dictionary Migrator JCL has not been uploaded to DMA.

Action:

Execute the JCL batch upload utility XDMPJCL. Retain as values:

USER=DEFAULT

PARMFILE=DEFAULT

DMA0052I STARTING POINT ESTABLISHED

Reason:

A STARTING POINT has been established. If the user just began to define a new migration, this message indicates that DMA collected values for the first starting point defined for the migration. If DMA collected values for the first starting point, this message means that values for a new starting point, either added to the parameter file, or to replace previously-specified values, have been collected.

Action:

None.

DMA0053E PLEASE ENTER A STARTING POINT

Reason:

At the Specify a New STARTING POINT screen, ENTER was pressed. However, no starting point was specified on the screen.

Action:

Provide values on the screen for one starting point.

Press Enter.

DMA0054I EXIT FROM SYSTEM CANCELED BY USER

Reason:

Upon viewing the Confirm Session EXIT screen, the user used the PREVIOUS command. Consequently, DMA returned the user to the previous screen.

Action:

None.

DMA0055E MISSING SETUP OWNER DURING LIST PROCESSING

Reason:

A severe error occurred when DMA attempted to display starting points

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0056I ACTIVE STARTING POINT ALTERED

Reason:

From the List of STARTING POINTS screen, S for select was typed to the left of a starting point that was not the active--current--starting point, and ENTER was pressed. The newly selected starting point was made the active, or current starting point.

Action:

None.

Note: Values subsequently supplied by DMA or provided by the user on DMA screens are for the newly selected starting point.

DMA0057I DELETE OF STARTING POINT CANCELED BY USER

Reason:

Upon viewing the Confirm DELETE of Starting Point screen, the user used the PREVIOUS command. Consequently, DMA returned the user to the List of STARTING POINTS screen.

Action:

None.

DMA0058E THE ACTIVE STARTING POINT CANNOT BE DELETED - PLEASE SELECT ANOTHER

Reason:

An attempt was made to delete the active--current--starting point.

Action:

Type S for select to the left of another starting point and press ENTER. This makes the newly selected starting point the active--current--starting point.

You can now type D for delete to the left of the previously active (previously current) starting point, and press Enter.

DMA0059E SUBMIT FAILED - SYNTAX FILE IN-LINE INDICATOR MISSING

Reason:

The JCL contains changes that prevent DMA from submitting parameters and JCL to CA IDMS Dictionary Migrator.

Action:

Make the following changes in the JCL in the member USMXTRCT step USMGRTR:

For z/OS users: after

```
//MIGRPARM DD *
```

delete all parameters (syntax) for CA IDMS Dictionary Migrator.

For z/VSE users: after

```
//EXEC USMGRTR
```

delete all parameters (syntax) for CA IDMS Dictionary Migrator.

For VM/ESA users: let

```
OSRUN=USMGRTR
```

DMA0060I MIGRATION JOB SUBMITTED

Reason:

The parameters for CA IDMS Dictionary Migrator and the JCL to run CA IDMS Dictionary Migrator were successfully submitted.

Action:

None.

DMA0061E SUBMIT FAILED - x IS NOT A SUPPORTED OPERATING SYSTEM

Reason:

An attempt was made to submit parameters for CA IDMS Dictionary Migrator and JCL to run CA IDMS Dictionary Migrator. However, the attempt failed because DMA is installed in a CV that has an active operating system other than z/OS; z/VSE; or z/VM.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0062E SUBMIT FAILED - JCL QUEUE IS MISSING

Reason:

An attempt was made to submit parameters to CA IDMS Dictionary Migrator and JCL to run CA IDMS Dictionary Migrator. However, JCL to run CA IDMS Dictionary Migrator was not available.

Action:

All JCL lines may have been deleted when the JCL was edited. If so, choose one:

- Access the parameter file. Use the JCL command to go to the EDIT JCL FOR MIGRATION screen. Type in the required JCL.
- Run the JCL batch upload utility XDMPJCL to upload again the CA IDMS Dictionary Migrator JCL to DMA.

If the JCL lines were not deleted when the JCL was edited, contact Technical Support at <http://ca.com/support>.

DMA0063E SUBMIT FAILED - SYNTAX QUEUE IS MISSING

Reason:

An attempt was made to submit parameters to CA IDMS Dictionary Migrator and JCL to run CA IDMS Dictionary Migrator. However, parameters (syntax) were not available.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0064E SEVERE ERROR DURING EDITOR QUIESCE

Reason:

An attempt to use the SHOW or JCL functions failed.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

DMA0065E INVALID OR EXTRANEIOUS COMMAND PARAMETERS

Reason:

During the use of the SHOW or JCL functions, extra commands were specified in addition to the PREVIOUS or END commands.

Action:

Delete the extra commands.

DMA0066I MIGRATION JCL MODIFIED

Reason:

From the JCL FOR MIGRATION screen, JCL was modified, ENTER was pressed, and the changes to the JCL were successfully processed.

Action:

None.

DMA0067I PLEASE ENTER THE PARMFILE NAME UNDER WHICH THE SETUP WILL BE STORED

Reason:

From the Store into a PARMFILE screen, ENTER was pressed. However, no parameter file (PARMFILE) name was provided to the right of the line 'Save as PARMFILE'.

Action:

Choose one:

- Type in the parameter file (PARMFILE) name after 'Save as PARMFILE'
- Leave the Store into a PARMFILE screen, using the command PREVIOUS, or another active transfer command.

DMA0068I PARAMETERS DOWNLOADED FOR USER x PARMFILE y

Reason:

The syntax batch download utility XDMPSYN was successfully executed. This execution means that the parameter file named in the message was copied from the PARMFILE database to a sequential file.

Action:

None.

ES00500E message MESSAGE NOT IN DICTIONARY

Reason:

While extracting a map or a CA ADS dialog module, a DISPLAY MESSAGE CODE statement was encountered, but the indicated message number was not in the source dictionary.

Action:

If the message in question already resides in the object dictionary, the migration is still valid and no action is needed. If the message is not in the object dictionary, the dialog terminates when it tries to execute the message display, so the migration is invalid. You must either add the message to the source dictionary and rerun CA IDMS Dictionary Migrator or add the message directly to the object dictionary.

ES00501E NOT IN module-name

Reason:

A CA ADS INCLUDE statement referenced a module that is not in the source dictionary.

Action:

Correct the INCLUDE statement and rerun CA IDMS Dictionary Migrator

ES00502E UNKNOWN TYPE OF xx IN USMES00

Reason:

An unexpected field was encountered in a CA IDMS Dictionary Migrator work file.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

ES00503E CURRENT OF RUN-UNIT MISMATCHES ENTITY-TYPE

Reason:

An unexpected record currency was returned to CA IDMS Dictionary Migrator.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

ES00504E UNKNOWN VALUE PASSED FROM USMEXTR TO USMES00

Reason:

An unexpected program communication was passed between CA IDMS Dictionary Migrator modules.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

ES00505E INDICATED ENTITY HAD NO RELATED LOAD MODULE

Reason:

The LOADS option was specified for the referenced entity, but there is no corresponding load module.

Action:

If you are migrating load modules only, or if you are not going to regenerate load modules from the migrated source, take either of these actions: regenerate the load module into the source dictionary and rerun CA IDMS Dictionary Migrator, or regenerate the object dictionary load modules from the migrated source.

ES00506E A TABLE IN MAP map-name IS NOT IN DICTIONARY

Reason:

An embedded code or edit table associated with an extracted map is not in source code form in the indicated dictionary.

Action:

If the map includes the table in link format, then the map is still usable. Otherwise, if the source for the table has been accidentally erased, it must be reconstructed for the migration to proceed correctly.

ES00507E A MODULE NEST INCLUDES ITSELF

Reason:

Using the INCLUDE feature of CA ADS, a module attempts to include itself.

Action:

This is a clear indication that this module has not been processed by the CA ADS compiler. ADSC processing ensures that generated dialogs cannot have this sort of nested dialog. Change the INCLUDE statement.

ES00508E PROCESSING HAS EXCEEDED DBKEY SAVE TABLE

Reason:

In processing one of the nested entity types, CA IDMS Dictionary Migrator has detected more than 500 levels of nesting. This message indicates an internal processing error during CA IDMS Dictionary Migrator processing.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

ES00509W THIS SCHEMA HAS A COMPILE ERROR

Reason:

If a schema currently has a compilation error, an error flag is set in the schema record.

Action:

Correct the schema source in the syntax file before uploading the schema.

ES00510W UNABLE TO DETERMINE MESSAGE ID IN "DISPLAY"

Reason:

A CA ADS verb DISPLAY, included in the source module of an extracted dialog, may contain a syntax error or a variable name.

Action:

If the verb DISPLAY contains a syntax error, correct the error. If the verb DISPLAY contains a variable name, you must extract the possible messages from the source dictionary.

ES00511E SYNTAX ERROR "INCLUDE" IN MODULE

Reason:

A CA ADS verb INCLUDE included in the source module of an extracted dialog contains syntax errors that prevents regeneration.

Action:

Correct the source of the dialog module.

ES00512E UNDETERMINED DIALOG REFERENCE AT LINE NUMBER: line-number

Reason:

While parsing for dialog name, an invalid value was found.

Action:

Correct the source statement at the indicate line number.

ES00513E FLAG CHECKED IS NOT A SINGLE BIT

Reason:

An internal error has been detected.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

ES00514E ENTITY NEST EXPLOSION TABLE SIZE EXCEEDED

Reason:

In processing one of the nested entity types, CA IDMS Dictionary Migrator has detected more than 500 levels of nesting. In this case, one of the entity occurrences has as a component another entity occurrence of the same type, and the second entity occurrence also has as a component the first entity occurrence.

Action:

Check the relationships of the entities in the dictionary, decide which *one* of the entity occurrences should have as a component the other entity occurrence, and correct the connections.

ES00515E SOURCE CHANGE WITHOUT COMPILATION OF LOAD

Reason:

The indicated SOURCE entity has been changed without recompiling the associated LOAD version.

Action:

Recompile the indicated SOURCE entity and resubmit the migration.

ES00516E RELATED x/NAME y/VERSION v HAS A DATE DISCREPANCY WITH THE INDICATED ENTITY

Reason:

This message appears in the Source Dictionary Verification report when a subordinate entity has been changed, but one of its owners has not.

Action:

Review the relationship between the entities and determine whether to continue with the migration.

ES00517W REDUNDANCY TABLE ROUTINE - NON ZERO RC : return-code

Reason:

An error has occurred in table processing. See the return codes table for a list of possible return codes.

Action:

Processing continues, results of the migration will be the same. However, performance degradation may occur. If errors occur repeatedly, contact Technical Support at <http://ca.com/support>.

ES00518E REFERENCED PROGRAM TABLE OVERFLOW - LINE line-number

Reason:

A CA ADS process module contains references to more than 500 different dialogs. An internal table has filled up.

Action:

The EXSOURCE=SUBPROGRAM parameter may be used to avoid the extraction of all referenced dialogs.

ES00519F USER UNAUTHORIZED FOR ACCESS TO ENTITY TYPE

Reason:

The user named in the IDSOURCE parameter does not have IDD authority for the entity type named.

Action:

The user must have at least DISPLAY authority for entity types which may be migrated from the source dictionary. Either update the user to grant DISPLAY authority for the entity types needed, or change the parameters for the CA IDMS Dictionary Migrator run to avoid entity types for which the user ID has no authority.

ES11450E ADSA DIALOG IN SYSTEM NOT IN DICTIONARY

Reason:

The ADSA system related a function to a dialog, but the dialog is not in the dictionary.

Action:

If the system designer intended to migrate a partially completed system, no action is necessary. If the intent was to migrate a completed system, but no dialog was located, then locate or create the appropriate dialog and add it to the dictionary.

ES11451E CURRENT OF RUN-UNIT MISMATCHES ENTITY-TYPE

Reason:

An unexpected record currency was returned to CA IDMS Dictionary Migrator.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

ES11452E GLOBAL RECORD NOT FOUND

Reason:

A global record specified by a ADSA system was not located in the dictionary.

Action:

Add the record to the dictionary and rerun CA IDMS Dictionary Migrator.

ES11453E NO LOAD MODULE FOUND FOR APPLICATION

Reason:

The load module for the ADSA application was not located in the dictionary.

Action:

If the load module is in a load library, copy the load module to the load area and rerun CA IDMS Dictionary Migrator. If the load module is not in a load library, regenerate the ADSA application and rerun CA IDMS Dictionary Migrator.

ES11454E A SEVERE ERROR OCCURRED IN USMES12

Reason:

An internal error has occurred.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

ES11455E LOADTEXT RECORD NOT FOUND PREMATURE END OF SET

Reason:

An internal error was detected.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

EXTR200F UNKNOWN TYPE OF xx IN SELECT FILE

Reason:

An invalid value was encountered in a CA IDMS Dictionary Migrator work file.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

EXTR202F INVALID EXPLOSION OWNER OF record-name

Reason:

In processing one of the nested entity types during extraction, an invalid record currency was encountered.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

EXTR203E INDICATED ENTITY AND VERSION NOT FOUND

Reason:

An item that was specified for extraction could not be found in the source dictionary.

Action:

Review the EXTRACT statement for which this message is issued. Note the date and version used to identify items that are to be extracted. If you do not specify the desired date and version in your EXTRACT statement, these parameters default to values specified in the PROCESS statement, or CA IDMS Dictionary Migrator assigns default values in accordance with rules described in the chapter "Parameters".

EXTR205F RUN-UNIT FAILS. STATUS=nnnn

Reason:

The bind of the object database failed.

Action:

Check the meaning of the returned CA IDMS error status code in the *CA IDMS Messages and Codes Guide* manual. If the status code is 1469, check your NODE and DICTIONARY statements for accuracy. If the status code is 0966, check that your cv.sysctl file was correctly specified and that all dictionary areas are available to that CV.

EXTR207I NOTHING EXTRACTED FOR THIS REQUEST

Reason:

Nothing was extracted for the specified EXTRACT statement.

Action:

No action is required. Note that this message applies only to the specified EXTRACT statement.

EXTR208F RUN TERMINATED, NOTHING EXTRACTED**Reason:**

None of the EXTRACT statements resulted in an extraction.

Action:

Review the selection criteria you specified to determine why no extractions were performed.

EXTR209F INVALID USER FOUND IN IDSOURCE STATEMENT. USER=(user-id)**Reason:**

A user ID was specified in an IDSOURCE statement but the user is not in the source dictionary.

Action:

Correct the user ID.

EXTR210E SUBSCHEMA IN MULTIPLE SCHEMAS: (. . .schema list)**Reason:**

In an EXTRACT statement, a subschema that has duplicates of the same name was identified only by its subschema name. Because there are duplicate subschemas with that name, the schema name is also needed to conclusively identify the subschema.

Action:

Add the name of the appropriate schema to the statement in question. See the chapter "Parameters" for information on correct syntax.

EXTR211E DICTIONARY RELEASE LEVELS NOT EQUAL PROCESS STATEMENT DICTIONARY dictionary-name AT RELEASE release-level EXTRACT STATEMENT DICTIONARY dictionary-name AT RELEASE release-level**Reason:**

The release level of a source dictionary defined in an EXTRACT statement is different from the release level of a source dictionary defined in the PROCESS statement.

Action:

The EXTRACT statement is bypassed.

EXTR214F PROCESSING HAS EXCEEDED DBKEY SAVE TABLE

Reason:

In processing one of the nested entity types, CA IDMS Dictionary Migrator has detected more than 500 levels of nesting.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

EXTR215E UNSUPPORTED LEVEL OF ADSA

Reason:

An application has been encountered that was generated using an obsolete level of ADSA.

Action:

Recompile the indicated application using the current ADSA compiler, or use CA IDMS Dictionary Migrator Release 2.3 or earlier.

EXTR216F IDD SIGNON REQUIRED. NO USERID SPECIFIED

Reason:

Dictionary security requires a signon. IDSOURCE was not specified.

Action:

Provide the IDSOURCE parameter and rerun.

EXTR217F RESPONSIBILITY OVERRIDE NOT AUTHORIZED. USER=userid

Reason:

The user ID named in IDSOURCE does not have authority all and cannot use Responsibility override.

Action:

Remove the Responsibility override parameter or use a user ID that has authority all. Rerun the job.

EXTR218W REDUNDANCY TABLE ROUTINE - NON ZERO RC : return-code**Reason:**

An error has occurred in table processing. See the return codes table for a list of possible return codes.

Action:

Processing continues, results of the migration are the same. However, performance degradation may occur. If errors occur repeatedly, contact Technical Support at <http://ca.com/support>.

EXTR219W ENTITY EXCLUSION TABLE ROUTINE - RETURN CODE return-code**Reason:**

An error has occurred in table processing. A return code of 4 indicates a possible GSSLOAD failure for the exclusion table module. Ensure the exclusion table specified in the EXCLTAB= process parameter is in the load library concatenation for the Dictionary Migrator run.

Action:

Processing continues, results of the migration are the same. However, performance degradation may occur. If errors occurs repeatedly, contact Technical Support at <http://ca.com/support>.

FILE905E GSSFILE RETURNED AN ERROR DURING file-function WITH FILE file-name, RETURN CODES ARE n1, n2, n3, n4**Reason:**

The indicated file function could not be performed with the specified file.

Action:

See the following table to determine the reason the error occurred and appropriate action. If you receive a return code combination 0, 24, 0, 8, see z/VSE File Assignments for more information.

GSFL999I file-id IS NOT VSAM--WILL TRY QSAM**Reason:**

The indicated file is not a VSAM file (z/VSE only).

Action:

CA IDMS Dictionary Migrator attempted to open the indicated file for VSAM processing. No action is required.

IDMS001E PROGRAM x ABORTED WITH A STATUS OF y AT SEQUENCE z

Reason:

An unexpected CA IDMS status error occurred during database processing.

Action:

See the *CA IDMS Messages and Codes Guide* manual for the status as indicated by y. If the cause is not apparent, contact Technical Support at <http://ca.com/support>.

Two types of errors can be reported by the return codes of n1, n2, n3, and n4--non-VSAM file errors and VSAM file errors. The error is described by n2 and n4. For VSAM file errors, n4 is always equal to 28. The error is described by n1, n2, and n3. A general return code is given by n4 for both non-VSAM and VSAM errors. All return codes are decimal values.

n4	Reason	Action
4	End-of-file	Contact Technical Support.
8	Open error or file is not open	Look for JCL errors or for the use of improper files.
12	An I/O error has occurred	Find cause for I/O error.
16	Request not recognized	Contact Technical Support.
20	File was already opened	Contact Technical Support.
24	Parameter list error	Contact Technical Support.
28	VSAM error n1=R15 return code from VSAM n2=low order byte from R0 GENCB/MODCB type of error n3=VSAM feedback byte error in I/O request	Use n1, n2, and n3 to check for possible user errors. If there are no user errors, Contact Technical Support.
32	Insufficient storage	Increase storage for job step.
36	SYNAD error occurred	Contact Technical Support.
40	BPAM FIND error n1=R15 n2=R0	Use n1 and n2 (as described in Data Management Macro Instructions) to check for errors.
44	BPAM STOW error n1=R15 n2=0	Use n1 and n2 (as described in Data Management Macro Instructions) to check for errors.
n2	Reason	Action

n4	Reason	Action
0	n4=8, use of unopened file n4=24, parameter list error	Contact Technical Support. Contact Technical Support.
1	JCL/label override parm list	Remove DCB information from JCL and ensure that the correct files are referenced.
2	Parm list override JCL/label	Remove DCB information from JCL and ensure that the correct files are referenced.
3	Unrecognized request	Contact Technical Support.
4	z/OS x13 ABEND trapped at open	Fix cause for x13 ABEND.
5	Tried to update seq. file	Contact Technical Support.
6	VSAM write at other than load	Contact Technical Support.
7	SOS table could not expand	Contact Technical Support.
8	z/OS DCB open failed	Contact Technical Support.
9	SOS table buffer pointer lost	Contact Technical Support.
10	SOS table file CB not built	Contact Technical Support.
11	z/OS DD statement Missing	Supply missing DD statement.
12	VSAM ACB open failed	Contact Technical Support.
13	Record format invalid	Contact Technical Support.
14	Macro format invalid	Contact Technical Support.
15	Record length not numeric	Contact Technical Support.
16	Record length too large	Contact Technical Support.
17	Block size not numeric	Contact Technical Support.
18	Block size too large	Contact Technical Support.
19	Invalid z/VSE sysname table	Assemble a valid sysname table.
20	z/VSE sysname table entry missing	Assemble a sysname table with an entry for the missing one.
21	z/VSE LU number too large	Use an LU number within range.
22	z/VSE sysname is not numeric or is misspelled	Correct to a valid sysname.
23	z/VSE sysname blank	Do not use blank sysname.
24	z/VSE LU not assigned	Contact Technical Support.

n4	Reason	Action
25	z/VSE DTF prototype missing	Contact Technical Support.
26	z/VSE logic module missing	Generate missing logic module.
27	z/VSE CCW mismatch	Contact Technical Support.
28	File is not a PDS	Allocate file to a PDS.

LINK901E PARAMETER LIST ERROR - PGM = module-name

Reason:

The module could not be called because of an internal problem.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

LINK902E NOT FOUND IN LIBRARY - PGM = module-name

Reason:

The module cannot be located in the available load or core-image libraries.

Action:

Locate the load or core-image library that contains the module and make the library available to the job.

LINK903E NOT ENOUGH STORAGE - PGM = module-name

Reason:

The module cannot be loaded due to insufficient GETVIS area (z/VSE only).

Action:

To increase the GETVIS area:

Allocate a smaller amount of space for the problem program area, using the SIZE= parameter; and, if necessary,

1. Use a larger partition.
2. Resubmit the job.

LINK905E UNANTICIPATED RETURN CODE = n PGM = module-name**Reason:**

An internal error has been detected.

Action:

For assistance, contact Technical Support <http://ca.com/support>.

NKWP001E UNRECOGNIZABLE MAJOR KEYWORD**Reason:**

The major keyword underscored by asterisks (*****) is misspelled or is in error and cannot be processed.

Action:

Correct the spelling of the keyword and resubmit the job.

NKWP002E DUPLICATE MINOR KEYWORD ENTERED**Reason:**

The minor keyword underscored by asterisks (*****) has already been entered on the parameter statement underscored.

Action:

Delete the duplicate minor keyword phrase and resubmit the job.

NKWP003E AN EQUAL SIGN (=) AND VALUE MUST FOLLOW THE KEYWORD**Reason:**

The required equal sign and value after the keyword are missing.

Action:

At the location underscored by an asterisk (*), enter an equal sign followed by an appropriate value for the parameter. Resubmit the job.

NKWP005E MISSING RIGHT PARENTHESIS. PHRASE TERMINATED**Reason:**

A set of open and closed parentheses is required.

Action:

Enter a right parenthesis to enclose the phrase or list, and resubmit the job.

NKWP006E REPEATED LIST IMPROPERLY CONTINUED

Reason:

The last phrase in this statement is a list (enclosed in parentheses) that ends with a comma. However, the next statement entered does not continue the list.

Action:

Continue the list by adding another keyword after the comma inside the parentheses; or move the comma outside the final parenthesis. Resubmit the job.

NKWP007E MISSING CONTINUATION CARD

Reason:

The previous statement ended with a comma, but this statement begins with a major keyword.

Action:

Supply a continuation card or delete the comma, and resubmit the job.

NKWP009E VALUE MUST BE xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Reason:

The value underscored by an asterisk (*) is incorrect for the keyword.

Action:

Supply the appropriate value, which is specified in the message, and resubmit the job.

NKWP010E KEYWORD PHRASES NOT SEPARATED BY COMMA

Reason:

Two keyword phrases are not separated by a comma.

Action:

Supply a comma at the location underscored by an asterisk (*) and resubmit the job.

NKWP011E UNRECOGNIZABLE MINOR KEYWORD

Reason:

The minor keyword underscored by asterisks (*****) is misspelled or is in error and was not processed.

Action:

Correct the spelling of the minor keyword and resubmit the job.

NKWP012E MINOR KEYWORD OMITTED**Reason:**

A minor keyword was omitted at the location underscored by an asterisk (*).

Action:

Enter the missing minor keyword and resubmit the job.

NKWP015E THESE KEYWORD(S) MUST BE SPECIFIED:XXXXXXXXXXXXXXXXXXXX**Reason:**

The minor keywords listed in the message were omitted. CA IDMS Dictionary Migrator cannot continue processing unless the indicated keyword or keywords are entered.

Action:

Supply the missing keyword or keywords and resubmit the job.

NKWP018E MISSING VALUE NEEDED XXXXXXXXXXXXXXXX**Reason:**

A keyword was entered without a value. The acceptable values are listed in the variable portion of the error message.

Action:

Enter the correct value at the location underscored by an asterisk (*) and resubmit the job.

NKWP019E VALUE ENTERED PREVIOUSLY**Reason:**

The value underscored by asterisks (****) was already entered in the list. CA IDMS Dictionary Migrator cannot process this parameter unless you supply two different values.

Action:

Correct the value and resubmit the job.

NKWP020E VALUE MUST BE ASCENDING

Reason:

The value underscored by asterisks (*****) must be in ascending numerical sequence from the previous value in the list.

Action:

Rearrange the values in the list in ascending sequence and resubmit the job.

NKWP021E VALUE MUST BE DESCENDING

Reason:

The value underscored by asterisks (*****) must be in descending numerical sequence from the previous value in the list.

Action:

Rearrange the values in the list in descending sequence and resubmit the job.

NKWP050E GSSNKWP INTERNAL ERROR. NOTIFY COMPUTER ASSOCIATES PRODUCTSUPPORT.

Reason:

An irrecoverable error has been detected.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

NKWP051E ERROR OCCURRED DURING CALL OF xxxxxxxx - GSSCALL RETURN CODE IS nn

Reason:

A module call processor error occurred.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

NKWP052E ERROR OCCURRED DURING LOAD OF xxxxxxxx - GSSLOAD RETURN CODE IS nn

Reason:

A module load processor error occurred.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

NKWP053E ERROR OCCURRED DURING WRITE OF xxxxxxxx - GSSPRINT RETURN CODE IS nn**Reason:**

An error occurred during a call of the printer module.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

NKWP061E DD STATEMENT MISSING FOR DDNAME: xxxxxxxx**Reason:**

A JCL DD card was not supplied.

Action:

Correct the JCL and resubmit the job.

NKWP062E LOGICAL UNIT NOT ASSIGNED: xxxxxxxx**Reason:**

An ASSGN card was not supplied.

Action:

Correct the JCL and resubmit the job.

NKWP063E SYSIPT LRECL MUST BE 80.**Reason:**

A parameter file with an LRECL other than 80 was supplied.

Action:

Create a parameter file with an LRECL of 80 and resubmit the job.

NKWP064E ERROR OCCURRED DURING OPEN OF xxxxxxxx FILE - GSSFILE RETURN CODE IS n1, n2, n3, n4**Reason:**

The file handler is unable to open the file.

Action:

See the previous table to determine the reason the error occurred and appropriate action.

NKWP065E ERROR OCCURRED DURING READ OF xxxxxxxx FILE - GSSFILE RETURN CODE IS n1, n2, n3, n4

Reason:

The file handler is unable to read the file.

Action:

See the previous table to determine the reason the error occurred and appropriate action.

NKWP066E ERROR OCCURRED DURING CLOSE OF xxxxxxxx FILE - GSSFILE RETURN CODE IS n1, n2, n3, nnn

Reason:

The file handler is unable to close the file.

Action:

See the previous table to determine the reason the error occurred and appropriate action.

NKWP080W SKIPPING UNTIL NEXT RECOGNIZABLE STATEMENT

Reason:

Because of a previous error, the phrase or phrases underscored by the letter X in the Parameter Verification Report have not been processed.

Action:

None.

NKWP081W SKIPPING UNTIL NEXT RECOGNIZABLE MINOR KEYWORD

Reason:

Because of a previous error, the value or values underscored by the letter X in the Parameter Verification Report have not been processed.

Action:

None.

NKWP091I INPUT PARAMETER STATEMENT

Reason:

A CA IDMS Dictionary Migrator input parameter statement has been read. The entire 80-byte image is displayed. Columns 1 through 72 are processed.

Action:

None.

NKWP092I CONTINUATION STATEMENT

Reason:

The parameter statement continues a previous statement entered.

Action:

None.

NKWP093I COMMENT STATEMENT

Reason:

The parameter statement contains an asterisk (*) as the first non-blank character. This indicates that it is a comment statement, and it is not processed.

Action:

None.

NKWP094I END OF PARAMETER INPUT

Reason:

All parameters that were entered have been read and processed by CA IDMS Dictionary Migrator.

Action:

None.

NKWP095I BLANK PARAMETER STATEMENT

Reason:

The parameter statement contains only blanks in columns 1 through 72 and is not processed.

Action:

Correct and resubmit the job.

NKWP096E UNMATCHED APOSTROPHE. NEED xxxxxxxxxxxxxx

Reason:

A value was entered that contains an odd number of apostrophes. Literal values must begin and end with single apostrophes. To represent a single apostrophe within a literal, you must enter two apostrophes. The acceptable values are listed in the variable portion of the error message.

Action:

Correct the value underscored by asterisks (****) and resubmit the job.

NKWP097E BAD LITERAL FORMAT. NEED xxxxxxxxxxxxxx

Reason:

A literal value was entered with improperly placed apostrophes. Literal values must begin and end with single apostrophes. To represent a single apostrophe within a literal, you must enter two apostrophes. The acceptable values are listed in the variable portion of the error message.

Action:

Correct the value underscored by asterisks (****) and resubmit the job.

PARM100E THE PROCESS PARAMETER CARD MUST BE THE FIRST CARD ENTERED

Reason:

The first statement in the input stream was not a PROCESS statement.

Action:

Correct the input stream so that the first statement it contains is a PROCESS statement.

PARM103E (CLASS,ATTRIBUTE) MUST BOTH BE ENTERED FOR CLSATTR

Reason:

Only one parameter was specified in the CLSATTR= clause.

Action:

Enter the missing class or attribute. Both the class and the attribute must be inside parentheses, separated by a comma.

PARM105I SYSTEM, CLASS, ATTR, AND USER DEFAULT TO "LEVEL=DIALOG"

Reason:

LEVEL=ENTITY was specified for one of these extract types, but CA IDMS Dictionary Migrator has changed it to LEVEL=DIALOG because LEVEL=ENTITY is not permitted in this context.

Action:

If you specified LEVEL=ENTITY intentionally because you wanted to extract only one entity occurrence, you will need to find another way to migrate the record. One way is to use OnLine IDD to add the entity to the object dictionary. If you specified LEVEL=ENTITY unintentionally, no action is needed.

PARM106E NO MIGRATOR STATEMENTS WERE PROVIDED FOR THIS RUN

Reason:

The input stream did not contain any CA IDMS Dictionary Migrator statements.

Action:

Specify CA IDMS Dictionary Migrator statements in the job stream.

PARM107E AT LEAST ONE "EXTRACT" STATEMENT IS REQUIRED

Reason:

The input stream did not include an EXTRACT statement.

Action:

Enter at least one EXTRACT statement.

PARM108E IRRECOVERABLE KEYWORD PROCESS ERROR = gssnkwp-return

Reason:

The code returned by the keyword processor identifies an irrecoverable error.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

PARM113E NEWVERSION MUST BE 0-9999 OR "NULL"

Reason:

An invalid value has been specified for NEWVERSION.

Action:

Correct the NEWVERSION value and resubmit.

PARM114E CLISTVERSION MUST BE 0-9999 OR "NULL"

Reason:

An invalid value has been specified for CLISTVERSION.

Action:

Correct the CLISTVERSION value and resubmit.

PARM115E SEARCH AND CHANGE STRINGS MUST BE PROVIDED

Reason:

CHANGE command was entered, but a SEARCH and REPLACE were not provided.

Action:

Correct the CHANGE command and resubmit.

PARM116E SEARCH STRING CANNOT BE NULL

Reason:

CHANGE command was entered, but the SEARCH string was null (").

Action:

Correct the SEARCH string of the CHANGE command to an explicit value.

PARM117E THE NUMERIC MESSAGE MUST BE 1 TO 6 DIGITS

Reason:

If only a message number is given, it can be no longer than 6 digits.

Action:

Correct the message number and resubmit the run.

PARM118E PREFIX IS PROVIDED, NUMERIC PORTION MUST BE 6 DIGITS**Reason:**

When the message prefix is specified, the message ID must be in the format AA999999 (A = Alphanumeric, 9 = numeric)

Action:

Correct the message ID and resubmit the run.

PARM121I LEVEL=ONLY NOT VALID FOR SYSTEM ENTITY DEFAULTS TO DIALOG.**Reason:**

Entity type of system can only be used with LEVEL=DIALOG.

Action:

Processing continues, the specified level parameter is ignored and dialogs related to the system are migrated.

PARM122W CVSOURCE/CVOBJECT NOT VALID USE SYSCTL AND SYSIDMS FILES TO SPECIFY ENVIRONMENT**Reason:**

CVSOURCE and CVOBJECT are not valid parameters.

Action:

The processing environment for CA IDMS Dictionary Migrator must be defined using the SYSCTL and SYSIDMS files, For more information, see the chapter "Operations" for details.

PARM123E EXTRACT STATEMENT INVALID FOR SPECIFIED RUNTYPE**Reason:**

An EXTRACT statement has been specified when the run type is IMPORT or CREATESYNTAX. For these run types the extract processing has been done in a previous job step and cannot be respecified now.

Action:

Correct the parameters and resubmit.

PARM124E INVALID IDENTIFIED FOR AN SQL ENTITY

Reason:

The specified identifier does not conform to correct SQL naming standards. Identifiers can be at most 18 characters long. Tables and view identifiers must be qualified by a schema name.

Action:

Correct the name of the SQL identity and resubmit.

PARM125E INVALID RUNTYPE SPECIFIED WITH SQLONLY PARAMETER

Reason:

SQLONLY parameter requires runtime to be EXPORT or PARMCHECK.

Action:

Change the runtime to EXPORT or PARMCHECK and resubmit.

PRNT910E DURING OUTPUT TO THE PRINT FILE, A SEVERE ERROR OCCURRED, ERROR=return code.

Reason:

An irrecoverable error condition has been detected.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

PS10600E EXTRACT FILE WAS EMPTY IN MODULE USMPS10

Reason:

An unexpected end-of-file has been encountered in a CA IDMS Dictionary Migrator work file.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

PS10601E INVALID EXTRACT ENTITY TYPE OF xx IN EXTRACT FILE READ IN MODULE USMPS10

Reason:

An unexpected value has been encountered in a CA IDMS Dictionary Migrator work file.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

PS10602E INVALID EXTRACT ENTITY UTILITY TYPE xx IN EXTRACT FILE READ IN MODULE USMPS10

Reason:

An unexpected value has been encountered in a CA IDMS Dictionary Migrator work file.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

PS10603E RETURN CODE FROM utility-name GREATER THAN 7. CHECK ERROR LISTING

Reason:

A utility has returned an error code that implies syntax errors in PUNCH statements.

Action:

Review the utility output messages for PUNCH statements, and evaluate the severity of the error based on the context in which it occurred. If necessary, correct the source dictionary and rerun CA IDMS Dictionary Migrator.

PS10604W WARNING-THIS IS AN ANALYSIS/WORK FILE AND MUST BE REVIEWED.

Reason:

Syntax created by CA IDMS Dictionary Migrator for schemas may require review.

Action:

Review the contents of the file and change the syntax in the appropriate syntax file (SCHMUPD).

PS10605W file-name FILE CONTAINS NO TRANSACTIONS: DO NOT UPLOAD.

Reason:

The file named in this message does not contain any transactions.

Action:

Do not include this file in the upload processing.

PS10606W FILE HAS MULTIPLE TRANSACTIONS: EDIT FILE BEFORE UPLOADING.

Reason:

The utility can process only one transaction each time it accesses a file. The file, however, contains multiple transactions that must be separated.

Action:

Edit the file to separate the transactions before performing the upload steps. Delete the message.

PS10607E FILE RETURNED FROM utility-name IS EMPTY. CHECK ERROR LISTING.

Reason:

A utility returned an empty syntax file that should have contained syntax statements.

Action:

Review the utility output error messages listed on the Syntax Production Report. Correct the error and rerun CA IDMS Dictionary Migrator.

PS10608I NEW-COPY MODULE IS a VERSION is b

Reason:

This message appears after a successful non-IMPORT run. The name and version of the MODULE containing a CLIST of VARY NEW-COPY statement is provided. This module is to be executed after migration load modules have been generated, thereby enabling the new copies.

Action:

Refer to the indicated module name and version for new-copying the migration components.

**PS10609F AN ERROR HAS OCCURRED WHEN ATTEMPTING TO CHANGE PROFILE #SETPROF
RETURN CODE = setprof-rc****Reason:**

An attempt to change the dictionary failed.

Action:

Contact Technical Support and report the return code. Ensure that the dictionary and node specifications in the parameters are correct.

PS10610W REDUNDANCY TABLE ROUTINE - NON ZERO RC : return-code**Reason:**

An error has occurred in table processing. See the return codes table for a list of possible return codes.

Action:

Processing continues, results of the migration will be the same. However, performance degradation may occur. If error occurs repeatedly, contact Technical Support at <http://ca.com/support>.

- **0**--Entity added -- No error has occurred.
- **1**--No further entries of this type in table space -- This is for information only; it does not necessarily imply an error.
- **2**--Element added -- No error has occurred.
- **4**--Occurrence found -- No error has occurred.
- **8**--Unknown entity-type -- Internal error; contact Technical Support at <http://ca.com/support>.
- **16**--Table space exhausted -- Increase region size.

PS20700W file-name FILE IS EMPTY**Reason:**

While generating hard copy of all the control statements that are used during the migration, CA IDMS Dictionary Migrator has found that one of the control statement files contains no statements.

Action:

The file is empty because the user-specified extraction criteria did not yield any control statements of the kind stored in this file.

PS20701E SEVERE ERROR IN PROCESSING EMBEDDED VERSIONS

Reason:

Unexpected condition encountered in load module processing.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

PS20702E ONE OF THE CHANGE COMMANDS OVERFLOWS LINE. CHANGE BYPASSED

Reason:

The new string of a change command is longer than the old string and there is not enough space on the line to absorb the difference.

Action:

The change must be made manually.

SORT500E ERROR DURING SORT, REVIEW SORT MESSAGES

Reason:

A CA IDMS Dictionary Migrator internal sort procedure encountered a fatal sort error.

Action:

Review the sort messages in the output. If the cause of the error is apparent, correct it and rerun CA IDMS Dictionary Migrator. If you cannot find the error, contact Technical Support at <http://ca.com/support>.

SORT501E INVALID USER FOUND IN IDOBJECT STATEMENT. USER=a

Reason:

This message appears during sort of CHANGEONLY extractions. A user has been specified in the IDOBJECT statement that is not in the OBJECT dictionary.

Action:

Correct the IDOBJECT user and resubmit.

SORT502E UNABLE TO CONTINUE MIGRATION DUE TO DEFAULT VERSION IN OBJECT DICTIONARY

Reason:

This message appears when NEWVERSION=NULL has been specified, and the installed values of default versions for new and existing entities do not meet the requirements of CA IDMS Dictionary Migrator.

Action:

Correct the values of default version in the Object dictionary, or do not use the NEWVERSION=NULL feature. The default version for new and existing entities must be a numeric integer (that is, not NEXT HIGHEST or LOWEST, etc.), and new and existing default specifications must be equal.

SPRS001E A REQUEST OF var WAS PASSED TO GSISPRS**Reason:**

An invalid action was requested for the submit function.

Action:

For assistance, contact Technical Support at <http://ca.com/> (have a listing of the dump available for reference).

SPRS002E AN INVALID SUBMIT SEQUENCE WAS ENCOUNTERED IN GSISPRS**Reason:**

An internal error has occurred.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

SPRS003E THE GET OF (USER, LONG) STORAGE FAILED WHEN IN GSISPRS**Reason:**

The submit function was unable to obtain storage.

Action:

Check the meaning of the return code from the get storage request in the *CA IDMS Messages and Codes Guide* manual and take the action suggested there

SPRS004E NOTHING WAS SUBMITTED**Reason:**

The file passed to the submit function was empty.

Action:

Ensure that the correct file is being used for the submit and that the file has the proper content.

SPRS005E AN INVALID OPERATING SYSTEM OF var WAS PASSED TO GISISPRS

Reason:

An invalid operating system code was provided to the submit function.

Action:

The value of 'var' was determined by the program calling the submit function. Contact Technical Support (have the listing of the dump available for reference).

SPRS006E SUBMIT RESOURCES TEMPORARILY UNAVAILABLE, RETRY

Reason:

The submit function is being used by another user.

Action:

Try to submit again. The submit function must have exclusive use of certain resources to avoid conflicts, but the resources are held for a short period of time.

SPRS007E var ALLOCATED INTRDR FAILED RC= var ERR CDE= var INFO CDE= var RETRY

Reason:

An attempt to dynamically allocate the internal reader failed with the codes indicated.

Action:

Refer to the *IBM System Macro and Facilities* manual SVC 99 for explanations of code and take the action indicated.

SPRS008E XPCC IDENT/CONNECT FAILED. ID CDE=identification-code ERR CDE=error-reason-code INFO CDE=information-reason-code

Reason:

The online job submission facility received an unsuccessful return code from z/VSE macro XPCC invoking the IDENTIFY or CONNECT function. The ID CDE indicates which macro call and situation occurred. Further information can be determined from the ERR CDE and INFO CDE values. See the *IBM z/VSE Power Application Programming* manual for further details.

Action:

This situation may be temporary. Re-issue the submit request. If the problem persists, make note of all information in the message and notify your DBA of the situation.

SPRS009E XPCC DISCONN/TERMIN FAILED. ID CDE=identification-code

Reason:

The online job submission facility received an unsuccessful return code from z/VSE macro XPCC invoking the DISCONNECT or TERMINATE function. The ID CDE indicates which macro call and situation occurred.

Action:

This situation can only occur after the JCL has successfully been submitted. If the problem persists, make note of all information in the message and notify your DBA of the situation.

SPRS010E XPCC PUT FAILED. ERR CDE=error-code INFO CDE=information-reason-code

Reason:

The online job submission facility received an unsuccessful return code from z/VSE macro XPCC invoking the SENDR function. Further information can be determined from the ERR CDE and the INFO CDE values. See the *IBM z/VSE Power Application Programming* manual for further details.

Action:

This situation may be temporary. Re-issue the submit request. If the problem persists, make note of all information in the message and notify your DBA of the situation.

ULOD800E INVALID NODE OR DB SPECIFIED DBNAME=a NODE=b

Reason:

The OBJECTDICTIONARY OBJECTNODE as specified for the migration run was not able to be bound.

Action:

Correct the invalid parameter and resubmit.

ULOD801E USERUPD FILE IS EMPTY

Reason:

There are no users in the USERUPD file. This error can be caused by specifying the DISCONNECT=(USER) parameter, or by the lack of USER associations in the source dictionary.

Action:

Either remove the DISCONNECT=USER parameter, or continue the migration without change

ULOD802W USER a MISSING FROM OBJECT DICTIONARY

Reason:

The indicated USER cannot be located in the Object dictionary for password restoration.

Action:

Review the migration to determine whether the USER was migrated. If the USER was deleted prior to this step, no further action is required. Otherwise, the USER should be in the Object dictionary and eligible for modification.

ULOD803I ALL USERS AND PASSWORDS PROCESSED

Reason:

Each migrated USER now has their password converted to the value in the SOURCE dictionary.

Action:

None.

USMS001I CATALOG NAVIGATION COMPLETED

Reason:

Navigation of the SQL catalog has been successfully completed.

Action:

None.

USMS002I EXTRACT TABLE-SCHEMA: schema-name , TABLE: table-name

Reason:

An extract record has been written for the named SQL schema and table.

Action:

None.

USMS003I EXTRACT VIEW-SCHEMA: schema-name , VIEW: view-name

Reason:

An extract record has been written for the named SQL schema and view.

Action:

None.

USMS004I EXTRACT SCHEMA-SCHEMA: schema-name

Reason:

An extract record has been written for the named SQL schema.

Action:

None.

USMS005E INVALID SQL ENTITY TYPE CODE: CODE

Reason:

An invalid type code for an SQL entity was passed to the catalog navigation program.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

USMS006E REDUNDANCY TABLE ROUTINE - NOW ZERO RC: return-code

Reason:

An error has occurred in table processing during catalog navigation. See Figure 7.2 for a list of possible return codes.

Action:

Processing continues but catalog navigation terminates for the SQL entity being extracted.

USMS007E SQL TABLE SPACE EXHAUSTED

Reason:

An out-of-storage condition has occurred which will limit the number of SQL entities to be extracted.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

USMS008E SQL ENTITY - Schema: schema-name , Table: table-name - is not a view

Reason:

An extract statement for an SQLVIEW was submitted but the named SQL entity is a table.

Action:

None. Processing continues and an extract is written for the named SQL schema and table.

USMS009E NETWORK REFERENCE NOT EXTRACTED - Schema: schema-name , Table: table-name

Reason:

Network references are not extracted from the SQL catalog.

Action:

None. Processing continues.

USMS010I EXTRACT CONSTRAINT - SCHEMA: schema-name CONSTRAINT: constraint-name

Reason:

An extract record has been written for the named SQL constraint.

Action:

None.

USMS011I EXTRACT TABLE PROCEDURE - SCHEMA: schema-name , PROCEDURE: table procedure name

Reason:

An extract record has been written for the named table procedure.

Action:

None.

USMS012I EXTRACT KEY - KEYNAME: keyname , ON SCHEMA: schema-name , PROCEDURE: table procedure name**Reason:**

An extract record has been written for the name key.

Action:

None.

USMS013E SQL ENTITY NOT FOUND - SCHEMA: schema-name , TABLE/VIEW: table-or-view**Reason:**

The requested SQL entity was not found in the catalog.

Action:

Correct the extract statement and resubmit

USXD002E UNRECOGNIZED COMMAND**Reason:**

An invalid primary command was encountered. The cursor is on the command.

Action:

Check the syntax of the command. Correct and reenter the command.

USXD202E INVALID PARAMETER ON COMMAND**Reason:**

A valid command has been entered with an unrecognizable parameter.

Action:

Check the syntax of command parameter. Correct and reenter the parameter.

USXD204E ENTER LOCATE PARAMETER**Reason:**

A line number was not included as a parameter in the LOCATE command.

Action:

Check the syntax of the command. Correct the command (include line number parameter) and reexecute.

USXD206E ONLY A NUMBER, MAX, HALF, CSR, OR PAGE IS ALLOWED

Reason:

An illegal option has been used with the LEFT, RIGHT, UP, or DOWN command.

Action:

Check the syntax of the parameter. Correct the parameter (you may have included some erroneous characters in the syntax line) and reexecute

USXD208E ONLY "ON" OR "OFF" IS ALLOWED. "ON" IS THE DEFAULT

Reason:

An illegal option has been used with the NULLS command.

Action:

Check the syntax of the parameter. Correct the parameter (you may have included some erroneous characters in the syntax line) and reexecute.

USXD210E NO PARAMETER IS ALLOWED ON COMMAND

Reason:

A parameter has been included in the syntax of the CANCEL command which excludes parameters.

Action:

Check the syntax of the command. Correct the command (delete the parameter) and reexecute.

USXF001I FOUND CHARS: string

Reason:

The string indicated has been found in the text. The cursor will be positioned at the beginning of the found string.

Action:

None.

USXF003I REP CHAR: "string" WITH "replacement-string"

Reason:

The string indicated has been replaced with the replacement string. The cursor is positioned after the replacement string.

Action:

None.

USXF004E MUST ENTER FIND/CHANGE FIRST

Reason:

The RFIND (repeat find) or RCHANGE (repeat change) command was entered without a FIND or a CHANGE command being entered first.

Action:

Be sure to enter the FIND command or the CHANGE command before the RFIND command or the RCHANGE command.

USXF005I BOTTOM OF DATA REACHED

Reason:

The bottom of the text was reached before the string was found. If you enter an RFIND (repeat find) or an RCHANGE (repeat change) at this time, the editor begins searching for the string at the top of the text.

Action:

None.

USXF006E ENCLOSE STRING IN QUOTES

Reason:

A single quotation mark was found where a pair is required or embedded blanks were found in the search string.

Action:

Check quotation marks, correct, and reenter the command. If a string has embedded blanks, enclose it in quotation marks.

USXF007E INVALID FIND SYNTAX

Reason:

The string specified in the FIND command is invalid. If the string has embedded blanks or quotation marks, it may appear to the editor to be more than one string: you can only specify one string in the FIND command.

Action:

Check the syntax of the command. If the string has embedded blanks, enclose it in quotation marks. If the string has embedded single quotation marks, enclose it in double quotation marks. If the string has embedded double quotation marks, enclose it in single quotation marks. Then reenter the command.

USXF008E INVALID CHANGE SYNTAX

Reason:

A string specified in the CHANGE command is invalid. If one of the strings has embedded blanks or quotation marks, it may appear to the editor that you are entering more than two strings: you can only specify two strings in the CHANGE command.

Action:

Check the syntax of the command. If a string has embedded blanks, enclose it in quotation marks. If a string has embedded single quotation marks, enclose it in double quotation marks. If a string has embedded double quotation marks, enclose it in single quotation marks. Then reenter the command.

USXF009I NO CHAR: string FOUND

Reason:

The editor has searched the entire text without finding the indicated string.

Action:

None.

USXF010W ERROR CHARACTERS "string"

Reason:

The string used to replace the one found is too long.

Action:

Adjust the length of the found string or reduce the length of the replacement string to match the length of the one found.

USXF012I FOUND nnn OCCURRENCES OF "string"**Reason:**

A string specified in the FIND ALL command has been found the specified number of times.

Action:

None.

USXF014I CHANGED nnn OCCURRENCES OF "string"**Reason:**

A string specified in the CHANGE ALL command has been changed in the text the specified number of times.

Action:

None.

USXF016E CHANGE COMMAND NOT ENABLED IN BROWSE MODE**Reason:**

The CHANGE command was used in the Browse Option to search for and change the next occurrence of a string.

Action:

None. The CHANGE command can only be used when editing text (Edit Option) and not while in Browse mode.

USXF018W CHANGED nnn OCCURRENCES WITH nnn ERRORS OF "string"**Reason:**

The string identified in the CHANGE ALL command has been changed successfully (nnn OCCURRENCES) with (nnn ERRORS) unsuccessful changes.

Action:

Look for the error message on the line number field(s) and make the necessary correction(s).

USXI002E INVALID COMMAND

Reason:

An invalid line command has been entered.

Action:

Check the syntax of the line command on the line number field. Correct and reenter the command.

USXI006E INVALID ON THIS LINE

Reason:

A line command has been entered on a line for which the command is invalid.

Action:

Reenter the line command on an appropriate line.

USXI008E INVALID SCROLL AMOUNT

Reason:

An invalid Scroll Option was entered in the SCROLL field.

Action:

Check the syntax of the Scroll Option next to the word SCROLL. Be sure the option starts in the left-most position of the field. Correct syntax or position.

USXL002W BLOCK COMMAND INCOMPLETE

Reason:

A partial sequence of Line Commands, in block form (CC, DD, MM, RR), is incomplete and cannot be executed.

Action:

If you are just scrolling or using the FIND command to locate the line to complete the block command, this message serves as a warning. Complete the command and execute. Otherwise, check the syntax of the commands to be sure all commands in block form are in pairs. Correct the commands and reexecute.

USXL004E COMMAND CONFLICT

Reason:

A sequence of Line Commands cannot be executed.

Action:

Check the order of the commands. Correct the commands and reexecute.

USXL006W MOVE/COPY PENDING

Reason:

A move or a copy cannot be executed because of a missing A (after) or B (before) line command.

Action:

Enter an A or a B command to indicate the location to which lines are to be moved or copied. This message serves as a warning if you are just scrolling or using the FIND command to locate the line for the A or B command.

USXL008E ITERATION COUNT MISMATCH

Reason:

A set of repeat (RR) line command iterations have conflicting values.

Action:

Check the syntax of the command. Make sure that the values of the iterations match (one or both may be left blank).

USXL009E LINE COMMAND RANGE CONFLICT

Reason:

A sequence of Line Commands cannot be executed because the range of one line command overlaps the range of another.

Action:

Check the order of the commands. Correct the commands and reexecute.

VALD201F AN INVALID TYPE OF xx FOUND IN VSAMEXT

Reason:

An unexpected value has been encountered in a CA IDMS Dictionary Migrator work file.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

VALD203F REPOSITION ON VSAMEXT FILE FAILED USING SAVED KEY

Reason:

An internal file error occurred during CA IDMS Dictionary Migrator processing.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

VALD204W NO EXTRACT FOR entity-type entity-name VERSION nn

Reason

A relationship encountered in the object dictionary is not among the entities extracted from the source dictionary.

Action:

Review the missing relationship to determine the impact of extracting or not extracting the entity.

VALD206W ELEMENT OVERRIDE IN RECORD record-name VERSION nn

Reason:

An element override has been specified for a record element.

Action:

Review the element in the source dictionary to see whether the override should be migrated. Otherwise, correct the element on the source dictionary and rerun CA IDMS Dictionary Migrator or edit the syntax in the syntax subfiles.

VALD208F AN INVALID UTILITY TYPE xx FOUND IN VSAMEXT

Reason:

An unexpected value has been encountered in a CA IDMS Dictionary Migrator work file.

Action:

For assistance, contact Technical Support at <http://ca.com/support>.

VALD210F INVALID NODE OR DB SPECIFIED**Reason:**

The bind of the object database failed.

Action:

Review the OBJDICTIONARY and parameters of the PROCESS statement and SYSCTL and SYSIDMS file allocations to determine the cause of the invalid run-unit.

VALD211F INVALID USER FOUND IN IDOBJECT STATEMENT. USER=user-id**Reason:**

A user ID was specified for the IDOBJECT parameter, but the user is not authorized to access the object dictionary.

Action:

Correct the user ID in the IDOBJECT parameter to allow the upload to process correctly.

VALD214W file-name FILE CONTAINS NO TRANSACTIONS: DO NOT UPLOAD**Reason:**

The file named in the message contains no transactions.

Action:

step for the file named in the above message. Remove this set of JCL statements from the upload job.

VALD215W FILE file-name CONTAINS MULTIPLE TRANSACTIONS: FILE MUST BE EDITED BEFORE UPLOADING.**Reason:**

The utility can process only one transaction each time it accesses a file. The file, however, contains multiple transactions that must be separated.

Action:

Edit the file to separate the transactions before performing the upload steps.

VALD216W EXISTING PROG-051 RECORD NOT AN APPLICATION**Reason:**

While checking to see whether an application can be loaded into the object dictionary, CA IDMS Dictionary Migrator encountered in the object dictionary a PROG-051 record for an entity other than an application.

Action:

To load the application:

- Delete or rename the PROG-051 record in the object dictionary or
- Rename the application you are intending to migrate, and resubmit the job.

VALD217F VSAMEXT FILE NOT DEFINED AS A VSAM DATASET

Reason:

Because the VSAMEXT file is in some format other than VSAM, processing has terminated.

Action:

Check whether an allocation was made for a VSAM dataset. If not, allocate a VSAM data set. If an allocation was made for a VSAM data set, delete any data set with the same name as the VSAMEXT VSAM data set. Resubmit the job.

VALD218E PICTURE CONFIGURATION MISMATCH SOURCE=a OBJECT=b

Reason:

This message appears when the primary picture clause in the source and object dictionaries do not match. "a" and "b" represent the COBOL picture clause associated with the element.

Action:

Review the information and make a decision based upon the context of the picture mismatch.

VALD219I EXTRACT FORCED FOR entity-type entity-name VERSION version-number

Reason:

An entity, which is to be migrated, cannot be changed unless the entity named in this message is regenerated. The entity named was found in the source dictionary while traversing the extraction path, but was not extracted due to selection criteria.

Action:

Review the named entity and the changes to the entity that caused the forced regeneration. If there is no effect, upload the files as created. If the effect is detrimental, either the entity causing the forced regeneration must be removed from the migration, or the entity requiring regeneration added to the migration. Because the entity was part of the structure being migrated from the source dictionary, the regeneration often has no adverse effects.

Note: If the entity named in the message is related to several migrated entities, this message appears repeatedly, listing each relationship. The entity is regenerated only once.

VALD220W NEWVERSION PROCESSING

Reason:

When using NEWVERSION option, two entities with the same type and name, but with different versions were extracted. Because only one version can exist after migration, the conflicting entity occurrences are eliminated.

Action:

Review all versions of the entity occurrence named to determine if the single version (shown in the entity detail report) can be used in all instances. If it can, proceed with the migration; otherwise, resolve the conflict in entity names.

VALD221W MULTIPLE VERSIONS IN SOURCE DICTIONARY entity-type entity-name entity-version - ENTITY AND VERSION SHOWN WILL NOT BE MIGRATED'

Reason:

When using NEWVERSION option, the two entities with the same type and name, but with different versions, were extracted.

Action:

Review all versions of the entity occurrence named to determine if the single version (shown in the entity detail report) can be used in all instances. If it can, proceed with the migration, otherwise, resolve the conflict in entity names.

VALD222W entity name VERSION version number NO LOAD MODULE TEXT FOUND

Reason:

A loadhdr-156 record was found in the source DDLDCLOD area for the entity named, but no loadtext-157 records are connected to it.

Action:

This is an error in the object dictionary load area. Run CA IDMS/DB Audit to determine if there are other errors.

VALD223W PROGRAM program-name VERSION version-number REQUIRES RECOMPILATION DUE TO MAP map-name

Reason:

A map has been migrated or forced to regenerate, which in turn requires the programs using it to be recompiled to update the map date stamp. Dialogs are automatically forced to regenerate.

Action:

Recompile the named program.

VALD224F CHANGEONLY PARAMETER VALUES ARE DIFFERENT IN EXPORT AND IMPORT RUNS. PROCESSING TERMINATED.

Reason:

The CHANGEONLY or ADDONLY parameters were not consistently specified in the EXPORT and IMPORT run.

Action:

Determine which parameter should be used and rerun the appropriate steps.

VALD225W REDUNDANCY TABLE ROUTINE ERROR CODE number - ENTITY TYPE/NAME: entity type-code - entity-name

Reason:

An internal error has occurred (possibly insufficient region size for a get storage request). See the return codes table for a list of return codes.

Action:

Processing continues, results of migration are the same. Performance degradation may occur, however. If error occurs repeatedly, contact Technical Support at <http://ca.com/support>.

VALD226F TARGET DICTIONARY SIGNON REQUIRED

Reason:

Target dictionary security requires a IDOBJECT user ID be provided.

Action:

Supply a valid IDOBJECT

VALD227F RESPONSIBILITY OVERRIDE NOT AUTHORIZED. USER=user-name**Reason:**

The IDOBJECT user ID provided does not have authority to use Registration-Override.

Action:

Either omit the Registration-Override option from IDOBJECT, or provide an IDOBJECT user ID with authority to use Registration-Override.

VALD228F USER UNAUTHORIZED FOR ACCESS TO ENTITY TYPE: entity-type**Reason:**

The IDOBJECT user ID does not have at least DISPLAY authority for the named entity type and validation processing (for this run) requires access to that entity type. The authorization at an entity type level is only done when the entity type is actually encountered.

Action:

Use an IDOBJECT user ID with at least display authority for all entity types, OR insure that entity types for which the IDOBJECT user ID does not have authority are not selected for migration.

VALD229W REDUNDANCY TABLE ROUTINE - NON ZERO RC number**Reason:**

The free storage request for an internal table failed.

Action:

In the short run, this error should have no effect on a migration run, or its results unless the region/partition is not large enough to allow that storage not to be reused.

VALD230W REGENERATION FORCED FOR EXTRANEIOUS entity-type entity-name VERS version-number**Reason:**

An entity, which is to be migrated, cannot be changed unless the entity named in this message is regenerated. The entity named is not among the entities extracted from the source dictionary.

Action:

Review the named entity and the effect changes to the entity causing the forced regeneration has upon the named entity. If there is no effect, upload the files as created. If the effect is detrimental, either the entity causing the forced regeneration must be removed from the migration, or the entity requiring regeneration is added to the migration (after correcting whatever ill effects that might occur).

Note: If the entity named in the message is related to several migrated entities, this message appears repeatedly, listing each relationship. The entity is regenerated only once.

VALD231W – OPTIONS DIFFERENT options list

Reason:

The table named has different characteristics in the source and object dictionaries. The characteristics cannot be changed in a MODIFY command, therefore the update of the table is complete unless remedial action is taken first.

Action:

Determine which characteristics should be used and correct the appropriate entity.

Appendix A: Sample JCL

This appendix contains the following topics:

This section contains the following topics:

[Introduction](#) (see page 221)

[Allocate the Syntax Files—z/OS](#) (see page 223)

[Allocate the VSAM Work File—z/OS](#) (see page 225)

[Extract Information from the Source Dictionary—z/OS](#) (see page 226)

[Export Syntax Files—z/OS](#) (see page 229)

[Import the Extract Tape--z/OS](#) (see page 232)

[Upload Syntax Files--z/OS](#) (see page 235)

[Delete Migrated Source from Source Dictionary—z/OS](#) (see page 241)

[Allocate the VSAM Work File--z/VSE](#) (see page 243)

[Extract Information from the Source Dictionary - z/VSE](#) (see page 244)

[Export Syntax Files--z/VSE](#) (see page 248)

[Import the Extract Tape—z/VSE](#) (see page 252)

[Upload Syntax Files—z/VSE](#) (see page 256)

[Delete Migrated Source from Source Dictionary—z/VSE](#) (see page 267)

[Allocate the VSAM Work File--z/VM](#) (see page 271)

[Extract Information from the Source Dictionary—z/VM](#) (see page 272)

[Export Syntax Files--z/VM](#) (see page 275)

[Import the Extract Tape--z/VM](#) (see page 278)

[Upload Syntax Files—z/VM](#) (see page 281)

[Delete Migrated Source from Source Dictionary - z/VM](#) (see page 292)

Introduction

This appendix contains the z/OS and z/VSE JCL for the CA IDMS Dictionary Migrator.

File Assignments

Even if you use a storage management tool such as CA DYNAM, CA IDMS Dictionary Migrator requires an ASSGN statement for every file except SORTWKnn. This ASSGN is required because CA IDMS Dictionary Migrator has its own device-independent support which dynamically builds a DTF based on the device type indicated by the ASSGN. Unless the ASSIGN specifies VSAM or BDAM, the file may be defined with either DLBL or TLBL.

z/OS

- USMSYNTAX
- USMVSAM
- USMXTRCT
- USMEXPRT
- USMIMPRT
- USMLOAD1
- USMLOAD3

z/VSE

- USMVSAM.S
- USMXTRCT.S
- USMEXPRT.S
- USMIMPRT.S
- USMLOAD1.S
- USMLOAD3.S

z/VM

- USMVSAM
- USMXTRCT
- USMEXPRT
- USMIMPRT
- USMLOAD1
- USMLOAD3

Allocate the Syntax Files—z/OS

The JCL model contained in the distribution SAMPJCL library member USMSYNTAX allocates the syntax files. You must supply a DD statement for each of the syntax files listed in the Syntax Files Table in the chapter "Operations."

```

/*
//DELETE EXEC PGM=IEFBR14
//RHDCDEL DD DSN=syntax.file.RHDCDEL,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//SCHMDEL DD DSN=syntax.file.SCHMDEL,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//SUBSDEL DD DSN=syntax.file.SUBSDEL,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//DDDLDEL DD DSN=syntax.file.DDDLDEL,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//DDLUPD DD DSN=syntax.file.DDLUPD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//BCFUPD DD DSN=syntax.file.BCFUPD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//TABLLOD DD DSN=syntax.file.TABLLOD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//DDDLLOD DD DSN=syntax.file.DDDLLOD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//SCHMUPD DD DSN=syntax.file.SCHMUPD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//SUBSUPD DD DSN=syntax.file.SUBSUPD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//SUBSLOD DD DSN=syntax.file.SUBSLOD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//RHDCUPD DD DSN=syntax.file.RHDCUPD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//RHDCLOD DD DSN=syntax.file.RHDCLOD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//DDLPGM DD DSN=syntax.file.DDLPGM,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//ADSOBGN DD DSN=syntax.file.ADSOBGN,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//ADSBTAT DD DSN=syntax.file.ADSBTAT,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
//USERUPD DD DSN=syntax.file.USERUPD,DISP=(MOD,DELETE),
//          UNIT=disk,SPACE=(TRK,0)
/*
/* ----- ALLOCATE SYNTAX FILES ----- 'USMSYNTAX' SAMPLE JCL -----
/*
//ALLOCATE EXEC PGM=IEFBR14
//RHDCDEL DD DSN=syntax.file.RHDCDEL,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk

```

```
//SCHMDEL DD DSN=syntax.file.SCHMDEL,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//SUBSDEL DD DSN=syntax.file.SUBSDEL,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//DDDDEL  DD DSN=syntax.file.DDDDEL,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//DDLUPD  DD DSN=syntax.file.DDDLUPD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(060,30)),
//          VOL=SER=volser,UNIT=disk
//BCFUPD  DD DSN=syntax.file.BCFUPD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//TABLLOD DD DSN=syntax.file.TABLLOD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//DDLLOD  DD DSN=syntax.file.DDDLLOD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//SCHMUPD DD DSN=syntax.file.SCHMUPD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//SUBSUPD DD DSN=syntax.file.SUBSUPD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//SUBSLOD DD DSN=syntax.file.SUBSLOD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//RHDCUPD DD DSN=syntax.file.RHDCUPD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(060,30)),
//          VOL=SER=volser,UNIT=disk
//RHDCLOD DD DSN=syntax.file.RHDCLOD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//DDLPGM  DD DSN=syntax.file.DDDLPGM,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//ADSOBGN DD DSN=syntax.file.ADSOBGN,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//ADSBTAT DD DSN=syntax.file.ADSBTAT,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(001,02)),
//          VOL=SER=volser,UNIT=disk
//USERUPD DD DSN=syntax.file.USERUPD,DISP=(,CATLG,DELETE),
//          SPACE=(TRK,(001,02)),
//          VOL=SER=volser,UNIT=disk
```


- **syntax.file**—We recommend that you retain the DDNAME corresponding to each syntax file within each data set name. Replace *syntax.file* with a name suitable to your site.
- **disk**—An appropriate unit designation for each syntax file.
- **volser**—The volume serial number of the disk that contains a particular syntax file.

Allocate the VSAM Work File—z/OS

The JCL model contained in the distribution SAMPJCL library member USMVSAM allocates the VSAM work file. See the chapter "Operations" for details.

```

//*
//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=a
//SYSIN DD *
DELETE -
          (work.file.vsamext) -
          CLUSTER
DEFINE CLUSTER ( -
               NAME(work.file.vsamext) -
               TRACKS(prialloc secalloc) -
               INDEXED -
               VOLUMES(volser) -
               KEYS(84 2) -
               RECORDSIZE(228 228) -
               usetype -
               )

```

- **a**—An appropriate SYSOUT class for your site.
- **work.file.vsamext** — The data set name of the VSAM KSDS work file. We suggest that the data set name include the work file name VSAMEXT.
- **prialloc secalloc**—The primary and secondary space allocations for the VSAM KSDS work file. A track allocation of 60 primary tracks and 30 secondary tracks (60 30) on a 3350 disk device should be adequate for this file.
- **volser**—The volume serial number of the disk that contains the VSAM KSDS work file.
- **usetype**—The appropriate usage type: REUSE or UNIQUE. If you specify UNIQUE, you must delete and define this cluster prior to each execution of CA IDMS Dictionary Migrator.

Extract Information from the Source Dictionary—z/OS

The JCL model contained in the distribution SAMPJCL library member USMXTRCT extracts information from the source dictionary, compares the extract to the object dictionary, reports on the comparison, and produces the syntax files.

```

/**
/**  ---CA IDMS/Dictionary Migrator Step --- 'USMEXTRCT Sample JCL ---
/**
/**MIGRATOR EXEC PGM=USMGRTR,REGION=####K
/**STEPLIB DD DSN=your.loadlib,DISP=SHR,DCB=BLKSIZE=blksize
/**      DD DSN=idms.loadlib,DISP=SHR
/**
/**  SYSIDMS PARAMETER FILE
/**
/**SYSIDMS DD DSN=your.sysidms,DISP=SHR
/**
/**  A SYSIDMS Parameter File must be provided.
/**  If you wish to run in Local Mode specify the following values :
/**  LOCAL=dmcl-name,LOCAL=ON,JOURNAL=OFF
/**  If you wish to run CV a 80 byte blank record will suffice.
/**
/**  MIGRATOR WORK FILES
/**SYSCTL DD DSN=your.cv.sysctl,DISP=SHR
/**
/**SYSIPT DD DSN=&&SYSIPT,DISP=(,DELETE,DELETE),
/**      SPACE=(TRK,(060,30)),
/**      VOL=SER=volser,UNIT=disk
/**SYSPCH DD DSN=&&SYSPCH,DISP=(,DELETE,DELETE),
/**      SPACE=(TRK,(150,60)),
/**      VOL=SER=volser,UNIT=disk
/**SELECT DD DSN=&&SELECT,DISP=(,DELETE,DELETE),
/**      SPACE=(TRK,(002,01)),
/**      VOL=SER=volser,UNIT=disk
/**EXTRACT DD DSN=&&EXTRACT,DISP=(,DELETE,DELETE),
/**      SPACE=(TRK,(150,60)),
/**      VOL=SER=volser,UNIT=disk
/**VALDRPT DD DSN=&&VALDRPT,DISP=(,DELETE,DELETE),
/**      SPACE=(TRK,(005,02)),
/**      VOL=SER=volser,UNIT=disk
/**WORKFIL DD DSN=&&WORKFIL,DISP=(,DELETE,DELETE),
/**      SPACE=(TRK,(060,30)),
/**      VOL=SER=volser,UNIT=disk
/**WORKFL2 DD DSN=&&WORKFL2,DISP=(,DELETE,DELETE),
/**      SPACE=(TRK,(060,30)),
/**      VOL=SER=volser,UNIT=disk
/**VSAMEXT DD DSN=work.file.vsamext,DISP=OLD
/**SORTWK01 DD UNIT=disk,SPACE=(CYL,(5,2))
/**SORTWK02 DD UNIT=disk,SPACE=(CYL,(5,2))

```

```

//SORTWK03 DD UNIT=disk,SPACE=(CYL,(5,2))
//*
//* SYNTAX SUBFILES
//*
//RHDCDEL DD DSN=syntax.file.RHDCDEL,DISP=OLD
//SCHMDEL DD DSN=syntax.file.SCHMDEL,DISP=OLD
//SUBSDEL DD DSN=syntax.file.SUBSDEL,DISP=OLD
//DDDDEL DD DSN=syntax.file.DDDDEL,DISP=OLD
//DDLUPD DD DSN=syntax.file.DDLUPD,DISP=OLD
//BCFUPD DD DSN=syntax.file.BCFUPD,DISP=OLD
//TABLLOD DD DSN=syntax.file.TABLLOD,DISP=OLD
//DDLLOD DD DSN=syntax.file.DDLLOD,DISP=OLD
//SCHMUPD DD DSN=syntax.file.SCHMUPD,DISP=OLD
//SUBSUPD DD DSN=syntax.file.SUBSUPD,DISP=OLD
//SUBSLOD DD DSN=syntax.file.SUBSLOD,DISP=OLD
//RHDCUPD DD DSN=syntax.file.RHDCUPD,DISP=OLD
//RHDCLOD DD DSN=syntax.file.RHDCLOD,DISP=OLD
//DDLPGM DD DSN=syntax.file.DDLPGM,DISP=OLD
//ADSOBGN DD DSN=syntax.file.ADSOBGN,DISP=OLD
//ADSBTAT DD DSN=syntax.file.ADSBTAT,DISP=OLD
//USERUPD DD DSN=syntax.file.USERUPD,DISP=OLD
//*
//* REPORT FILES
//*
//AUDIT DD SYSOUT=a
//SYSOUT DD SYSOUT=a
//*
//* Output from dynamically involved CA IDMS utilities and
//* decompilers is written to SYSLIST. This will be a large file.
//*
//SYSLST DD SYSOUT=a
//*
//* CA IDMS Dictionary Migrator USER PARAMETERS. See the Parameters
//* Chapter of the CA IDMS Dictionary Migrator User Guide for
//* keyword parameter information.
//*
//MIGPARM DD *
PROCESS,
    RUN= ????????,
*
*** IDENTIFY SOURCE DICTIONARY
    IDSOURCE=(userid,password),
    DICT=?????????,
*
*** IDENTIFY OBJECT DICTIONARY
    IDOBJECT=(userid,password),
    OBJDICT=?????????
.
.

```

```
EXTRACT,.....  
.  
.  
.  
EXTRACT,.....
```

- **####**—The number of kilobytes of storage required to run CA IDMS Dictionary Migrator. The recommended region size is 2000K.
- **your.loadlib**—The data set name of the OS/PDS library into which CA IDMS Dictionary Migrator was installed.
- **idms.loadlib**—The data set name of the OS/PDS library that contains the CA/IDMS utility load modules.
- **your.sysidms**—The data set name of the SYSIDMS parameter file that is used to customize your execution environment.
- **your.cv.sysctl**—The sysctl file for the CV by which the dictionary is accessed.
- **blksize**—If the block size of *your.loadlib* is smaller than that of *idms.loadlib*, specify the size of *idms.loadlib*.
- **volser**—The volume serial number of the disk to be used for a particular work file. Temporary space may not require a VOL parameter at your site.
- **disk**—An appropriate unit designation for each work or sort-work file.
- **work.file.vsamext**—The data set name of the VSAM KSDS work file work or sort-work file.
- **syntax.file**—We recommend that you retain the DDNAME corresponding to each syntax file within each data set name. Replace *syntax.file* with a name suitable to your site.
- **a**—The appropriate SYSOUT class for your site.
- **userid**—The standard identification of the user of the specified dictionary.
- **password**—A valid password that identifies the user of the specified dictionary.

Export Syntax Files—z/OS

The JCL model contained in the distribution SAMPJCL library member USMEXPRT allocates the work files, extracts information from the source dictionary and puts it onto disk, and produces syntax files on disk.

```

//*      ---- DICT MIGRATOR STAGE 1 EXPORT --- 'USMEXPRT' SAMPLE JCL ----
//*
//MIGRATOR EXEC PGM=USMGRTR,REGION=2000K
//STEPLIB DD DSN=your.loadlib,DISP=SHR,DCB=BLKSIZE=blksize
//          DD DSN=idms.loadlib,DISP=SHR
//*
//*      SYSIDMS PARAMETER FILE
//*
//SYSIDMS DD DSN=your.sysidms,DISP=SHR
//SYSCTL DD DSN=your.cv.sysctl,DISP=SHR
//*
//*      A SYSIDMS Parameter File must be provided.
//*      If you wish to run in Local Mode specify the following values :
//*      LOCAL=dmcl-name,LOCAL=ON,JOURNAL=OFF
//*      If you wish to run CV an 80 byte blank record will suffice
//*      and use the SYSCTL for that CV.
//*
//*      EXPORT 'EXTRACT' FILE ON TAPE FOR INPUT TO IMPORT STAGE
//*
//EXTRACT DD DSN=SYNTAX.FILE.EXTRACT,DISP=(NEW,CATLG,DELETE)
//          SPACE=(TRK,(150,060)),
//          VOL=SER=volser,UNIT=disk
//*
//*      MIGRATOR DISK WORK FILES
//*
//SYSIPT DD DSN=&&SYSIPT,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(060,30)),
//          VOL=SER=volser,UNIT=disk
//SYSPCH DD DSN=&&SYSPCH,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(150,60)),
//          VOL=SER=volser,UNIT=disk
//SELECT DD DSN=&&SELECT,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(002,01)),
//          VOL=SER=volser,UNIT=disk
//WORKFIL DD DSN=&&WORKFIL,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(060,30)),
//          VOL=SER=volser,UNIT=disk
//WORKFL2 DD DSN=&&WORKFL2,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(060,30)),
//          VOL=SER=volser,UNIT=disk
//SORTWK01 DD UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK02 DD UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK03 DD UNIT=disk,SPACE=(CYL,(5,2))

```

```
//*
//*  SYNTAX FILES
//*
//RHDCDEL DD DSN=syntax.file.RHDCDEL,DISP=OLD
//SCHMDEL DD DSN=syntax.file.SCHMDEL,DISP=OLD
//SUBSDEL DD DSN=syntax.file.SUBSDEL,DISP=OLD
//DDDLEL DD DSN=syntax.file.DDDLEL,DISP=OLD
//DDLUPD DD DSN=syntax.file.DDLUPD,DISP=OLD
//BCFUPD DD DSN=syntax.file.BCFUPD,DISP=OLD
//TABLLOD DD DSN=syntax.file.TABLLOD,DISP=OLD
//DDLLOD DD DSN=syntax.file.DDLLOD,DISP=OLD
//SCHMUPD DD DSN=syntax.file.SCHMUPD,DISP=OLD
//SUBSUPD DD DSN=syntax.file.SUBSUPD,DISP=OLD
//SUBSLOD DD DSN=syntax.file.SUBSLOD,DISP=OLD
//RHDCUPD DD DSN=syntax.file.RHDCUPD,DISP=OLD
//RHDCLOD DD DSN=syntax.file.RHDCLOD,DISP=OLD
//DDLPGM DD DSN=syntax.file.DDLPGM,DISP=OLD
//ADSOBGN DD DSN=syntax.file.ADSOBGN,DISP=OLD
//ADSBTAT DD DSN=syntax.file.ADSBTAT,DISP=OLD
//USERUPD DD DSN=syntax.file.USERUPD,DISP=OLD
//*
//*  REPORT FILES
//*
//AUDIT DD SYSOUT=a
//SYSOUT DD SYSOUT=a
//*
//*  OUTPUT FROM DYNAMICALLY INVOKED IDMS UTILITIES AND DECOMPILERS
//*  IS WRITTEN TO SYSLST. THIS WILL BE A LARGE FILE.
//*
//SYSLST DD SYSOUT=a
//*
//*  USER PARAMETERS
//*
//MIGPARM DD *
PROCESS,
  RUN=EXPORT,
*
*** IDENTIFY SOURCE DICTIONARY
*
  IDSOURCE=(user-id,password),
  DICT=????????,
*
  THE OBJECT DICTIONARY WILL NOT BE ACCESSED IN THIS STEP.
*
  VALUES USED FOR IDOBJECT AND OBJDICT WILL BE PASSED TO THE
  UPLOAD STEP VIA THE SYNTAX FILES CREATED IN THIS STEP.
*
  IDOBJECT=(user-id,password),
  OBJDICT=????????
```

```

      .
      .
EXTRACT,.....
      .
      .
EXTRACT,.....

```

- **####**—The number of kilobytes of storage required to run the EXPORT stage of CA IDMS Dictionary Migrator. The recommended region size is 2000K.
- **your.loadlib**—The data set name of the OS/PDS library into which CA IDMS Dictionary Migrator was installed.
- **blksize**—If the block size of *your.loadlib* is smaller than that of *idms.loadlib*, specify the size of *idms.loadlib*.
- **idms.loadlib**—The data set name of the OS/PDS library that contains CAIDMS utility modules.
- **your.sysidms**—The data set name of the SYSIDMS parameter file that is used to customize your execution environment.
- **your.cv.sysctl**—The sysctl file for the CV by which the dictionary is accessed.
- **volser**—The volume serial number of the disk that contains a particular work file. Temporary space may not require a VOL parameter in your site.
- **disk**—An appropriate unit designation for each work or sort-work file.
- **syntax.file**—We recommend that you retain the DDNAME corresponding to each syntax file within each data set name. Replace *syntax.file* with a name suitable to your site.
- **a** — An appropriate SYSOUT class for your site.
- **userid**—The standard identification of the user of the specified dictionary.
- **password**—A valid password that identifies the user of the specified dictionary.

Import the Extract Tape--z/OS

The JCL model contained in the distribution SAMPJCL library member USMIMPRT imports the extract tape, compares extract to object dictionary, and reports on the comparison.

Note: Suggested space allocations for all workfiles are contained in the Work Files Table of in the chapter "Operations."

```

/*      ----- Dictionary Migrator STEP ----- 'USMIMPRT' SAMPLE JCL -----
/*
//MIGRATOR EXEC PGM=USMGRTR,REGION=####K
//STEPLIB DD DSN=your.loadLib,DISP=SHR,DCB=BLKSIZE=blksize
//          DD DSN=idms.loadLib,DISP=SHR
/*
/*      SYSIDMS PARAMETER FILE
/*
//SYSIDMS DD DSN=your.sysidms,DISP=SHR
//SYSCTL  DD DSN=your.cv.sysctl,DISP=SHR
/*
/*      A SYSIDMS Parameter File must be provided.
/*      If you wish to run in Local Mode specify the following values :
/*      LOCAL=dmcl-name,LOCAL=ON,JOURNAL=OFF
/*      If you wish to run CV an 80 byte blank record will suffice.
/*
/*
/*      MIGRATOR WORK FILES
/*
//SYSIPT  DD DSN=&&SYSIPT,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(060,30)),
//          VOL=SER=volser,UNIT=disk
//SYSPCH DD DSN=&&SYSPCH,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(150,60)),
//          VOL=SER=volser,UNIT=disk
//SELECT  DD DSN=&&SELECT,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(002,01)),
//          VOL=SER=volser,UNIT=disk
//VALDRPT DD DSN=&&VALDRPT,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(005,02)),
//          VOL=SER=volser,UNIT=disk
//WORKFIL DD DSN=&&WORKFIL,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(060,30)),
//          VOL=SER=volser,UNIT=disk
//WORKFL2 DD DSN=&&WORKFL2,DISP=(,DELETE,DELETE),
//          SPACE=(TRK,(060,30)),
//          VOL=SER=volser,UNIT=disk
//VSAMEXT DD DSN=WORK.FILE.VSAMEXT,DISP=OLD
//SORTWK01 DD UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK02 DD UNIT=disk,SPACE=(CYL,(5,2))

```



```

//SORTWK03 DD UNIT=disk,SPACE=(CYL,(5,2))
//*
//* SYNTAX SUBFILES
//*
//RHDCDEL DD DSN=syntax.file.RHDCDEL,DISP=OLD
//SCHMDEL DD DSN=syntax.file.SCHMDEL,DISP=OLD
//SUBSDEL DD DSN=syntax.file.SUBSDEL,DISP=OLD
//DDDDEL DD DSN=syntax.file.DDDDEL,DISP=OLD
//DDLUPD DD DSN=syntax.file.DDLUPD,DISP=OLD
//BCFUPD DD DSN=syntax.file.BCFUPD,DISP=OLD
//TABLLOD DD DSN=syntax.file.TABLLOD,DISP=OLD
//DDLLOD DD DSN=syntax.file.DDLLOD,DISP=OLD
//SCHMUPD DD DSN=syntax.file.SCHMUPD,DISP=OLD
//SUBSUPD DD DSN=syntax.file.SUBSUPD,DISP=OLD
//SUBSLOD DD DSN=syntax.file.SUBSLOD,DISP=OLD
//RHDCUPD DD DSN=syntax.file.RHDCUPD,DISP=OLD
//RHDCLOD DD DSN=syntax.file.RHDCLOD,DISP=OLD
//DDLPGM DD DSN=syntax.file.DDLPGM,DISP=OLD
//ADSOBGN DD DSN=syntax.file.ADSOBGN,DISP=OLD
//ADSBTAT DD DSN=syntax.file.ADSBTAT,DISP=OLD
//USERUPD DD DSN=syntax.file.USERUPD,DISP=OLD
//*
//* MIGRATED EXTRACTION FILE FROM USMEXPRT
//*
//EXTRACT DD DSN=syntax.file.EXTRACT,DISP=OLD
//*
//* REPORT FILES
//*
//AUDIT DD SYSOUT=a
//SYSOUT DD SYSOUT=a
//*
//* OUTPUT FROM DYNAMICALLY INVOKED IDMS UTILITIES AND DECOMPILERS
//* IS WRITTEN TO SYSLST. THIS WILL BE A LARGE FILE.
//*
//SYSLST DD SYSOUT=a
//*
//* USER PARAMETERS
//*
//MIGPARM DD *
PROCESS,
    RUN=IMPORT,
*
*** IDENTIFY SOURCE DICTIONARY
    IDSOURCE=(user-id,password),
    DICT=????????,
*
*** IDENTIFY OBJECT DICTIONARY
    IDOBJECT=(user-id,password),
    OBJDICT=????????

```

·
·
·

- **####**—The number of kilobytes of storage required to run the IMPORT stage of CA IDMS Dictionary Migrator. The recommended region size is 2000K.
- **blksize**—If the block size of *your.loadlib* is smaller than that of *idms.loadlib*, specify the size of *idms.loadlib*.
- **your.loadlib**—The data set name of the OS/PDS library into which CA IDMS Dictionary Migrator was installed.
- **idms.loadlib**—The data set name of the OS/PDS library that contains CAIDMS utility load modules
- **your.sysidms**—The data set name of the SYSIDMS parameter file that is used to customize your execution environment.
- **your.cv.sysctl**—The sysctl file for the CV by which the dictionary is accessed.
- **volser**—The volume serial number of the disk that contains a particular work file. Temporary space may not require a VOL parameter in your site.
- **disk** — An appropriate unit designation for each work or sort-work file.
- **syntax.file**—We recommend that you retain the DDNAME corresponding to each syntax file within each data set name. Replace *syntax.file* with a name suitable to your site.
- **a**—An appropriate SYSOUT class for your site.
- **userid**—The standard identification of the user of the specified dictionary.
- **password**—A valid password that identifies the user of the specified dictionary.

Upload Syntax Files--z/OS

The JCL model contained in the distribution SAMPJCL library member USMLOAD1 is used to upload syntax files to the object dictionary using CA IDMS utilities, or using the CA user password upload utility supplied by CA. There can be from 1 to 16 steps involved in the upload, depending on your site and type of migration.

For more information about uploading syntax files, see the Syntax Files Table in the chapter "Operations."

If you want to delete entities from the source dictionary after migration, make a second non-CHANGEONLY run using the same parameters as the real migration, specify the same dictionary for both source and object.

```

--- USE IDMS TO LOAD OBJECT DICT ----- 'USMLOAD1' SAMPLE JCL ---

/* RHDCDEL - DELETE PANELS AND MAPS *
/*-----*
//RHDCDEL EXEC PGM=RHDCMAP1,REGION=4000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadLib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.RHDCDEL,DISP=OLD
/*
/*-----*
/* SCHMDEL - DELETE SCHEMA AND SUBSCHEMA SOURCE *
/*-----*
//SCHMDEL EXEC PGM=IDMSCHEM,REGION=3000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadLib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.SCHMDEL,DISP=OLD
/*
/*-----*
/* SUBSDEL - DELETE SUBSCHEMA *
/* N O T E: THIS IS REQUIRED ONLY IF A SUBSCHEMA WAS MIGRATED *
/* WITHOUT ITS SCHEMA *
/*-----*
//SUBSDEL EXEC PGM=IDMSUBSC,REGION=2500K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadLib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here

```

```
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.SUBSDEL,DISP=OLD
//*
/*-----*
/* DDDLDEL - DELETE OTHER ENTITIES *
/*-----*
/*NOTE: DO NOT RUN THIS STEP FOR CHANGEONLY PROCESSING. *
/*-----*
/*DDDLDEL EXEC PGM=IDMSDDL,REGION=3000K,COND=((16,LT),EVEN)
/*STEPLIB DD DSN=idms.loadlib,DISP=SHR
/*SYSCTL DD DSN=idms.sysctl,DISP=SHR
/*SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
/*SYSLST DD SYSOUT=a
/*SYSPCH DD DUMMY
/*SYSIPT DD DSN=syntax.file.DDDLDEL,DISP=OLD
/*-----*
/* DDDLUPD - ADD/MODIFY CLASS, ATTRIBUTE, SYSTEM, RECORD, ELEMENT, *
/* MESSAGE, MODULE, AND TABLE ENTITIES *
/*-----*
/*DDLUPD EXEC PGM=IDMSDDL,REGION=3000K,COND=((16,LT),EVEN)
/*STEPLIB DD DSN=idms.loadlib,DISP=SHR
/*SYSCTL DD DSN=idms.sysctl,DISP=SHR
/*SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
/*SYSLST DD SYSOUT=a
/*SYSPCH DD DUMMY
/*SYSIPT DD DSN=syntax.file.DDLUPD,DISP=OLD
/*
/*-----*
/* TABLLOD - GENERATE TABLE LOAD MODULES *
/*-----*
/*TABLLOD EXEC PGM=IDMSDDL,REGION=3000K,COND=((16,LT),EVEN)
/*STEPLIB DD DSN=idms.loadlib,DISP=SHR
/*SYSCTL DD DSN=idms.sysctl,DISP=SHR
/*SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
/*SYSLST DD SYSOUT=a
/*SYSPCH DD DUMMY
/*SYSIPT DD DSN=syntax.file.TABLLOD,DISP=OLD
/*
/*-----*
/* DDDLLOD - ADD/MODIFY LOAD MODULES FOR SUBSCHEMA, MAP, MAP EDIT *
/* TABLE, AND DIALOG *
/*-----*
/*DDLLOD EXEC PGM=IDMSDDL,REGION=3000K,COND=((16,LT),EVEN)
/*STEPLIB DD DSN=idms.loadlib,DISP=SHR
/*SYSCTL DD DSN=idms.sysctl,DISP=SHR
```

```

//SYSIDMS DD *
//* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.DDDLLOD,DISP=OLD
//*
/*-----*
/* SCHMUPD - ADD SCHEMA SOURCE *
/*-----*
//SCHMUPD EXEC PGM=IDMSCHEM,REGION=2500K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
//* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.SCHMUPD,DISP=OLD
//*
/*-----*
/* SUBSUPD - ADD SUBSCHEMA SOURCE *
/*-----*
//SUBSUPD EXEC PGM=IDMSUBSC,REGION=2500K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
//* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.SUBSUPD,DISP=OLD
//*
/*-----*
/* SUBSLOD - GENERATE SUBSCHEMA LOAD MODULE *
/*-----*
//SUBSLOD EXEC PGM=IDMSUBSC,REGION=2500K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
//* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.SUBSLOD,DISP=OLD
//*
/*-----*
/* RHDCUPD - ADD/MODIFY PANEL AND MAP SYNTAX *
/*-----*
//RHDCUPD EXEC PGM=RHDCMAP1,REGION=4000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *

```

```

/** Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.RHDCUPD,DISP=OLD
/**
/**-----*
/** RHDCLOD - COMPILE MAP *
/**-----*
//RHDCLOD EXEC PGM=RHDCMPUT,REGION=4000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadLib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/** Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.RHDCLOD,DISP=OLD
/**
/**-----*
/** ADSOBGN - GENERATE ADS DIALOG *
/** N O T E: THIS STEP CAN ONLY BE EXECUTED IF DIALOG LOAD MODULES *
/** WERE MIGRATED. *
/**-----*
//ADSOBGN EXEC PGM=ADSOBCOM,REGION=5000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadLib,DISP=SHR
//CDMSLIB DD DSN=idms.loadLib,DISP=SHR
// DD DSN=idms.c.loadLib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/** Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.ADSOBGN,DISP=OLD
/**
/**-----*
/** DDDLPGM - ADD/MODIFY PROGRAM SYNTAX *
/**-----*
//DDLPGM EXEC PGM=IDMSDDL,REGION=3000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadLib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/** Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.DDDLPGM,DISP=OLD
/**
/**-----*
/** ADSBTAT - UPDATE TASK APPLICATION TABLE *
/** THIS STEP IS ONLY FOR APPLICATIONS WHICH USE THE *
/** ADSA. *

```

```

/*
/*-----*
//ADSBTAT EXEC PGM=ADSOBTAT,REGION=3000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
// DD DSN=your.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.ADSBTAT,DISP=OLD
/*
/*-----*
/* BCFUPD - ADD SQL ENTITY DEFINITIONS
/*
/*-----*
//DDDUPD EXEC PGM=IDMSBCF,REGION=3000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.BCFUPD,DISP=OLD
/*
/*-----*
/* USERUPD - UPDATE USER PASSWORDS
/* THIS STEP IS ONLY FOR MIGRATIONS WHICH DO NOT USE
/* DISCONNECT=(USER). THE PASSWORD FROM THE SOURCE
/* DICTIONARY IS PLACED INTO THE OBJECT DICTIONARY.
/*
/* N O T E: IF THIS STEP IS NOT RUN, THE PASSWORD ASSIGNED TO ALL
/* MIGRATED USERS WILL BE "DEMSINT".
/*
/*-----*
//USERUPD EXEC PGM=USMULOD,REGION=3000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
// DD DSN=your.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSOUT DD SYSOUT=a
//AUDIT DD SYSOUT=a
//SYSPCH DD DUMMY
//USERUPD DD DSN=syntax.file.USERUPD,DISP=OLD
/*

```

- **REGION=####K**—The number of kilobytes of storage needed to run each utility is already specified. See the Syntax Files Table in the chapter "Operations."
- **your.loadlib**—The data set name of the OS/PDS library into which CA IDMS Dictionary Migrator was installed.
- **idms.loadlib**—The data set name of the OS/PDS library in which each CA IDMS utility resides.
- **idms.c.loadlib**—The data set name of the OS/PDS library in which all CA IDMS C runtime load modules reside.
- **idms.sysctl**—The data set name of the CA IDMS SYSCTL file used to direct the upload process to the selected object CV.
- **a**—The appropriate SYSOUT class for your site.
- **syntax.file**—We recommend that you retain the DDNAME corresponding to each syntax file within each data set name. Replace *syntax.file* with a name suitable to your site.

Delete Migrated Source from Source Dictionary—z/OS

The JCL model contained in the distribution SAMPJCL library member USMLOAD3 is an optional job that, when executed, deletes the migrated source from the source dictionary after migration.

Note: If you want to delete entities from the source dictionary after migration, make a second non-CHANGEONLY run using the same parameters as the real migration, specify the same dictionary for both source and object.

```

/* ----- REMOVE MIGRATED ENTITIES FROM SOURCE DICTIONARY -----
/*
/*-----*
/* RHDCDEL - DELETE PANELS AND MAPS *
/*-----*
//RHDCDEL EXEC PGM=RHDCMAP1,REGION=4000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.RHDCDEL,DISP=OLD
/*
/*-----*
/* SCHMDEL - DELETE SCHEMA AND SUBSCHEMA SOURCE *
/*-----*
/*SCHMDEL EXEC PGM=IDMSCHEM,REGION=3000K,COND=((16,LT),EVEN)
//SCHMDEL EXEC PGM=IDMSDDL,REGION=3000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.SCHMDEL,DISP=OLD
/*
/*-----*
/* SUBSDEL - DELETE SUBSCHEMA *
/* N O T E: THIS IS REQUIRED ONLY IF A SUBSCHEMA WAS MIGRATED *
/* WITHOUT ITS SCHEMA *
/*-----*
//SUBSDEL EXEC PGM=IDMSUBSC,REGION=2500K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY

```

```
//SYSIPT DD DSN=syntax.file.SUBSDEL,DISP=OLD
//*
/*-----*
/* DDDLDEL - DELETE OTHER ENTITIES *
/*-----*
//DDLDEL EXEC PGM=IDMSDDL,REGION=3000K,COND=((16,LT),EVEN)
//STEPLIB DD DSN=idms.loadlib,DISP=SHR
//SYSCTL DD DSN=idms.sysctl,DISP=SHR
//SYSIDMS DD *
/* Insert the relevant SYSIDMS parameters here
//SYSLST DD SYSOUT=a
//SYSPCH DD DUMMY
//SYSIPT DD DSN=syntax.file.DDDLDEL,DISP=OLD
```

- **idms.loadlib**—The data set name of the OS/PDS library in which each CA IDMS utility resides.
- **idms.sysctl**—The data set name of the CA IDMS/DB SYSCTL file used to direct the upload process to the selected object CV
- **a** — An appropriate SYSOUT class for your site.
- **syntax.file**—We recommend that you retain the DDNAME corresponding to each syntax file within each data set name. Replace *syntax.file* with a name suitable to your site.

Allocate the VSAM Work File--z/VSE

The JCL in TOOLJCL library member USMVSAM.S (z/VSE) allocates the VSAM work file. See in the chapter "Operations" for details.

```
* ----- ALLOCATE VSAM WORK FILE ----- 'USMVSAM' SAMPLE JCL -----
*
// EXEC IDCAMS,SIZE=AUTO
  DELETE          -
                 work.file.vsamext -
                 CLUSTER
  DEFINE CLUSTER ( -
                 NAME(work.file.vsamext) -
                 TRACKS(prialloc secalloc) -
                 INDEXED -
                 VOLUMES(volser) -
                 KEYS(84 2) -
                 RECORDSIZE(228 228) -
                 usetype -
                 )
/*
/ &
```

- **work.file.vsamext**—The data set name of the VSAM KSDS work file. It is recommended that you retain the DLBL of VSAMEXT. Replace *work.file* with a name suitable to your site.
- **prialloc secalloc**—The primary and secondary space allocations for the VSAM KSDS work file. A track allocation of 60 primary tracks and 30 secondary tracks (60 30) on a 3350 disk device should be adequate for this file. You may need to change these allocations for your DASD type and migration needs.
- **volser**—The volume serial of the disk pack where the VSAM file is defined.
- **usetype**—The appropriate usage type. Specify REUSE.

Extract Information from the Source Dictionary - z/VSE

The JCL in TOOLJCL library member USMXTRCT.S (z/VSE) allocates the work files, allocates the syntax files, extracts information from the source dictionary, compares the extract to the object dictionary, reports on the comparison, and produces the syntax files.

```
* ----- Dictionary Migrator STEP ----- 'USMXTRCT' SAMPLE JCL -----
*      **** CREATE A SYSIDMS PARAMETER FILE ****
// UPSI  1
// OPTION LOG,PARTDUMP
// DLBL   anyname,'work.file.SYSIDMS',0,SD
// EXTENT SYS041,volser,,,948,1
// ASSGN  SYS041,DISK,VOL=volser,SHR
// EXEC   DITTO
$$DITTO CSQ FILEOUT=anyname
* SYSIDMS parameters.
* For Local Mode specify :
*
DMCL=dmcl-name,LOCAL=ON,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
*
* For CV runs specify :
*
DMCL=dmcl-name,LOCAL=OFF,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
/*
*
// OPTION PARTDUMP
*      **** z/VSE library where CA IDMS Dictionary Migrator resides
// DLBL   DBMS,'your.corelib'
// EXTENT ,volser
*
*      **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL   DBMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
*      ***** PARAMETER INPUT FOR DICT. MIGRATOR AND IDMS UTILITIES *****
*
// ASSGN  SYS011,YSRDR                      SYNTAX FILE
// ASSGN  SYS010,SYSPCH                      HOLD SYSPCH ASSGN
// DLBL   IJSYSPH,'work.file.SYSIPT',0,SD    IJSYSPH IS USED TO
// EXTENT          SYSPCH,volser,,,00100,050      CREATE VTOC ENTRY
//          ASSGN          SYSPCH,DISK,VOL=volser,SHR      FOR SYSIPT ASSGN
//          CLOSE         SYSPCH,SYS010
// DLBL   IJSYSIN,'work.file.SYSIPT',0,SD    SAME FILE-ID AND
// EXTENT          SYSIPT,volser,,,00100,050      TRACKS AS IJSYSPH
```

```

        ASSGN          SYSIPT,DISK,VOL=volser,SHR      PERMANENT ASSIGNMENT
// DLBL    SYSIPT,'work.file.SYSIPT',0,SD           SAME FILE-ID AND
// EXTENT          SYS012,volser,,00100,050         TRACKS AS IJSYSPH
// ASSGN          SYS012,DISK,VOL=volser,SHR
*
* **** OUTPUT FROM IDMS UTILITIES AND INPUT TO Dictionary Migrator ****
*
// ASSGN    SYSPCH,IGN
// DLBL    SYSPCH,'work.file.SYSPCH',0,SD
// EXTENT          SYS018,volser,,00150,200
// DLBL    IDMSPCH,'work.file.SYSPCH',0,SD          SAME FILE-ID AND
// EXTENT          SYS018,volser,,00150,200         TRACKS AS SYSPCH
// ASSGN          SYS018,DISK,VOL=volser,SHR
*
* ***** REPORT FILE *****
*
// ASSGN    SYS013,SYSLST
*
* ***** Dictionary Migrator WORK FILES *****
*
// DLBL    EXTRACT,'work.file.EXTRACT',0,SD
// EXTENT          SYS015,volser,,00350,200
// ASSGN          SYS015,DISK,VOL=volser,SHR
// DLBL    VSAMEXT,'vsam.file.VSAMEXT',0,VSAM
// EXTENT          SYS016,volser
// ASSGN          SYS016,DISK,VOL=volser,SHR
// DLBL    SELECT,'work.file.SELECT',0,SD
// EXTENT          SYS014,volser,,00550,003
// ASSGN          SYS014,DISK,VOL=volser,SHR
// DLBL    VALDRPT,'work.file.VALDRPT',0,SD
// EXTENT          SYS017,volser,,00553,007
// ASSGN          SYS017,DISK,VOL=volser,SHR
// DLBL    WORKFIL,'work.file.WORKFIL',0,SD
// EXTENT          SYS034,volser,,00560,100
// ASSGN          SYS034,DISK,VOL=volser,SHR
// DLBL    WORKFL2,'work.file.WORKFL2',0,SD
// EXTENT          SYS039,volser,,00660,100
// ASSGN          SYS039,DISK,VOL=volser,SHR
*
* ***** SYNTAX FILES FROM DICT MIGRATOR FOR IDMS UTILITIES *****
*
// DLBL    RHDCDEL,'SYNTAX.FILE.RHDCDEL',0,SD      MAP & PANEL DELETES
// EXTENT          SYS019,volser,,00760,005
// ASSGN          SYS019,DISK,VOL=volser,SHR
// DLBL    SCHMDEL,'SYNTAX.FILE.SCHMDEL',0,SD     SCHEMA DELETES
// EXTENT          SYS020,volser,,00765,005
// ASSGN          SYS020,DISK,VOL=volser,SHR
// DLBL    SUBSDEL,'SYNTAX.FILE.SUBSDEL',0,SD     SUBSCHEMA DELETES

```

```

// EXTENT      SYS021,volser,,00770,005
// ASSGN      SYS021,DISK,VOL=volser,SHR
// DLBL      DDDLDEL,'SYNTAX.FILE.DDDLDEL',0,SD      ENTITY DELETES
// EXTENT      SYS022,volser,,00775,010
// ASSGN      SYS022,DISK,VOL=volser,SHR
// DLBL      DDDLUPD,'SYNTAX.FILE.DDDLUPD',0,SD      ENTITY ADD/MODIFIES
// EXTENT      SYS023,volser,,00785,100
// ASSGN      SYS023,DISK,VOL=volser,SHR
// DLBL      BCFUPD,'SYNTAX.FILE.BCFUPD',0,SD      SQL UPDATES
// EXTENT      SYS040,volser,,00950,003
// ASSGN      SYS040,DISK,VOL=volser,SHR
// DLBL      TABLLOD,'SYNTAX.FILE.TABLLOD',0,SD      TABLE LOAD MODULES
// EXTENT      SYS024,volser,,00885,005
// ASSGN      SYS024,DISK,VOL=volser,SHR
// DLBL      DDDLLOD,'SYNTAX.FILE.DDDLLOD',0,SD      ENTITY LOAD MODULES
// EXTENT      SYS025,volser,,00890,005
// ASSGN      SYS025,DISK,VOL=volser,SHR
// DLBL      SCHMUPD,'SYNTAX.FILE.SCHMUPD',0,SD      ADD SCHEMAS
// EXTENT      SYS026,volser,,00895,005
// ASSGN      SYS026,DISK,VOL=volser,SHR
// DLBL      SUBSUPD,'SYNTAX.FILE.SUBSUPD',0,SD      ADD SUBSCHEMA
// EXTENT      SYS028,volser,,00905,005
// ASSGN      SYS028,DISK,VOL=volser,SHR
// DLBL      SUBSLOD,'SYNTAX.FILE.SUBSLOD',0,SD      REGENERATE SUBSCHEMAS
// EXTENT      SYS029,volser,,00910,005
// ASSGN      SYS029,DISK,VOL=volser,SHR
// DLBL      RHDCUPD,'SYNTAX.FILE.RHDCUPD',0,SD      ADD PANELS & MAPS
// EXTENT      SYS030,volser,,00915,005
// ASSGN      SYS030,DISK,VOL=volser,SHR
// DLBL      RHDCLOD,'SYNTAX.FILE.RHDCLOD',0,SD      RECOMPILE MAPS
// EXTENT      SYS031,volser,,00920,005
// ASSGN      SYS031,DISK,VOL=volser,SHR
// DLBL      ADSOBN,'SYNTAX.FILE.ADSOBN',0,SD      REGENERATE DIALOGS
// EXTENT      SYS032,volser,,00930,005
// ASSGN      SYS032,DISK,VOL=volser,SHR
// DLBL      DDDLPGM,'SYNTAX.FILE.DDDLPGM',0,SD      ADD PROGRAM
// EXTENT      SYS033,volser,,00925,005
// ASSGN      SYS033,DISK,VOL=volser,SHR
// DLBL      ADSBTAT,'SYNTAX.FILE.ADSBTAT',0,SD      REGENERATE ADSA
// EXTENT      SYS037,volser,,00942,003
// ASSGN      SYS037,DISK,VOL=volser,SHR
// DLBL      USERUPD,'SYNTAX.FILE.USERUPD',0,SD      PASSWORD MIGRATION
// EXTENT      SYS038,volser,,00945,003
// ASSGN      SYS038,DISK,VOL=volser,SHR
*
// DLBL      SYSIDMS,'work.file.SYSIDMS'
// EXTENT      SYS041,volser
// ASSGN      SYS041,DISK,VOL=volser,SHR
// EXEC      USMGRTR,SIZE=(USMGRTR,400K)

```

```

*
* ***** Dictionary Migrator SYNTAX *****
PROCESS,
  RUN= ????????,
*
*** IDENTIFY SOURCE DICTIONARY
  IDSOURCE=(USER-ID,PASSWORD),
  DICT=?????????,
*
*** IDENTIFY OBJECT DICTIONARY
  IDOBJECT=(USER-ID,PASSWORD),
  OBJDICT=?????????
.
.
.
EXTRACT,.....
.
.
.
/*
/&
// JOB RESET SYSIPT
  CLOSE SYSIPT,SYSDR
/&

```

- **your.corelib**—The file id of the z/VSE library into which CA IDMS Dictionary Migrator was installed.
- **idms.corelib**—The file id of the z/VSE library which contains the CA IDMS utility executable phases.
- **volser**—The volume serial number or generic assignment of the disk volume on which the library, as specified in the previous statement, resides.
- **.sublib**—The name of the sublibrary of the z/VSE library specified by the previous filename.
- **vsam.file**—It is recommended that you retain the DLBL name VSAMEXT within the vsam file name. Replace *vsam.file* with a name suitable to your site.
- **work.file**—It is recommended that you retain the corresponding DLBL name within each file-id. Replace *work.file* with names suitable to your site.
- **syntax.file**—It is recommended that you retain the corresponding DLBL name within each file-id. Replace *syntax.file* with names suitable to your site.
- **anyname**—Any suitable name for your SYSIDMS file. Note that the name chosen must be identical to the FILEOUT value in the DITTO copy step.

Export Syntax Files--z/VSE

The JCL in TOOLJCL library member USMEXPR.T (z/VSE) allocates the work files, allocates the syntax files, extracts information from the source dictionary and puts it onto disk, and produces syntax files on disk.

```

* ----- Dictionary Migrator STEP ----- 'USMEXPR.T' SAMPLE JCL -----
*
*      **** CREATE A SYSIDMS PARAMETER FILE ****
// UPSI  1
// OPTION LOG,PARTDUMP
// DLBL   anyname,'work.file.SYSIDMS',0,SD
// EXTENT SYS041,volser,,,948,1
// ASSGN  SYS041,DISK,VOL=volser,SHR
// EXEC   DITTO
$$$$DITTO CSQ FILEOUT=anyname
* SYSIDMS parameters.
* For Local Mode specify :
*
DMCL=dmcl-name,LOCAL=ON,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
*
* For CV runs specify :
*
DMCL=dmcl-name,LOCAL=OFF,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
/*
// OPTION PARTDUMP
*      **** z/VSE library where CA IDMS Dictionary Migrator resides
// DLBL   DBMS,'your.corelib'
// EXTENT ,volser
*      **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL   IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
*      ***** PARAMETER INPUT FOR DICT. MIGRATOR AND IDMS UTILITIES *****
*
// ASSGN  SYS011,YSRDR                                SYNTAX FILE
// ASSGN  SYS010,SYSPCH                                HOLD SYSPCH ASSGN
// DLBL   IJSYSPH,'work.file.sysipt',0,SD              IJSYSPH IS USED TO
// EXTENT          SYSPCH,volser,,,00100,050          CREATE VTOC ENTRY
//          ASSGN          SYSPCH,DISK,VOL=volser,SHR          FOR SYSIPT ASSGN
//          CLOSE         SYSPCH,SYS010
// DLBL   IJSYSIN,'work.file.sysipt',0,SD              SAME FILE-ID AND
// EXTENT          SYSIPT,volser,,,00100,050          TRACKS AS IJSYSPH
//          ASSGN          SYSIPT,DISK,VOL=volser,SHR          PERMANENT ASSIGNMENT
// DLBL   SYSIPT,'work.file.sysipt',0,SD              SAME FILE-ID AND

```



```

// EXTENT      SYS012,volser,,00100,050      TRACKS AS IJSYSPH
// ASSGN       SYS012,DISK,VOL=volser,SHR
*
* **** OUTPUT FROM IDMS UTILITIES AND INPUT TO Dictionary Migrator ***
*
// ASSGN  SYSPCH,IGN
// DLBL   SYSPCH,'work.file.SYSPCH',0,SD
// EXTENT      SYS018,volser,,00150,200
// DLBL   IDMPCH,'work.file.SYSPCH',0,SD      SAME FILE-ID AND
// EXTENT      SYS018,volser,,00150,200      TRACKS AS SYSPCH
// ASSGN       SYS018,DISK,VOL=volser,SHR
*
* ***** REPORT FILE *****
*
// ASSGN  SYS013,SYSLST
*
* ***** Dictionary Migrator WORK FILES *****
*
* ***** EXPORT EXTRACT FILE ON TAPE FOR INPUT TO IMPORT STAGE *****
*
// DLBL   EXTRACT,'WORK.FILE.EXTRACT'
// EXTENT      SYS015,volser,,00350,200
// ASSGN       SYS015,DISK,VOL=volser,SHR
// DLBL   SELECT,'WORK.FILE.SELECT',0,SD
// EXTENT      SYS014,volser,,00550,003
// ASSGN       SYS014,DISK,VOL=volser,SHR
// DLBL   WORKFIL,'WORK.FILE.WORKFIL',0,SD
// EXTENT      SYS034,volser,,00560,100
// ASSGN       SYS034,DISK,VOL=volser,SHR
// DLBL   WORKFL2,'WORK.FILE.WORKFL2',0,SD
// EXTENT      SYS039,volser,,00660,100
// ASSGN       SYS039,DISK,VOL=volser,SHR
*
* ***** SYNTAX FILES FROM DICT MIGRATOR FOR IDMS UTILITIES *****
*
// DLBL   RHDCDEL,'syntax.file.RHDCDEL',0,SD      MAP & PANEL DELETES
// EXTENT      SYS019,volser,,00760,005
// ASSGN       SYS019,DISK,VOL=volser,SHR
// DLBL   SCHMDEL,'syntax.file.SCHMDEL',0,SD      SCHEMA DELETES
// EXTENT      SYS020,volser,,00765,005
// ASSGN       SYS020,DISK,VOL=volser,SHR
// DLBL   SUBSDEL,'syntax.file.SUBSDEL',0,SD      SUBSCHEMA DELETES
// EXTENT      SYS021,volser,,00770,005
// ASSGN       SYS021,DISK,VOL=volser,SHR
// DLBL   DDDLDEL,'syntax.file.DDDLDEL',0,SD      ENTITY DELETES
// EXTENT      SYS022,volser,,00775,010
// ASSGN       SYS022,DISK,VOL=volser,SHR
// DLBL   DDDLUPD,'syntax.file.DDDLUPD',0,SD      ENTITY ADD/MODIFIES
// EXTENT      SYS023,volser,,00785,100

```

```

// ASSGN      SYS023,DISK,VOL=volser,SHR
// DLBL      BCFUPD,'syntax.file.BCFUPD',0,SD      SQL UPDATES
// EXTENT    SYS040,volser,,00950,003
// ASSGN      SYS040,DISK,VOL=volser,SHR
// DLBL      TABLLOD,'syntax.file.TABLLOD',0,SD      TABLE LOAD MODULES
// EXTENT    SYS024,volser,,00885,005
// ASSGN      SYS024,DISK,VOL=volser,SHR
// DLBL      DDDLLOD,'syntax.file.DDDLLOD',0,SD      ENTITY LOAD MODULES
// EXTENT    SYS025,volser,,00890,005
// ASSGN      SYS025,DISK,VOL=volser,SHR
// DLBL      SCHMUPD,'syntax.file.SCHMUPD',0,SD      ADD SCHEMAS
// EXTENT    SYS026,volser,,00895,005
// ASSGN      SYS026,DISK,VOL=volser,SHR
// DLBL      SUBSUPD,'syntax.file.SUBSUPD',0,SD      ADD SUBSCHEMA
// EXTENT    SYS028,volser,,00905,005
// ASSGN      SYS028,DISK,VOL=volser,SHR
// DLBL      SUBSLOD,'syntax.file.SUBSLOD',0,SD      REGENERATE SUBSCHEMAS
// EXTENT    SYS029,volser,,00910,005
// ASSGN      SYS029,DISK,VOL=volser,SHR
// DLBL      RHDCUPD,'syntax.file.RHDCUPD',0,SD      ADD PANELS & MAPS
// EXTENT    SYS030,volser,,00915,005
// ASSGN      SYS030,DISK,VOL=volser,SHR
// DLBL      RHDCLOD,'syntax.file.RHDCLOD',0,SD      RECOMPILE MAPS
// EXTENT    SYS031,volser,,00920,005
// ASSGN      SYS031,DISK,VOL=volser,SHR
// DLBL      ADSOBN,'syntax.file.ADSOBN',0,SD      REGENERATE DIALOGS
// EXTENT    SYS032,volser,,00930,005
// ASSGN      SYS032,DISK,VOL=volser,SHR
// DLBL      DDDLPGM,'syntax.file.DDDLPGM',0,SD      ADD PROGRAM
// EXTENT    SYS033,volser,,00925,005
// ASSGN      SYS033,DISK,VOL=volser,SHR
// DLBL      ADSBTAT,'syntax.file.ADSBTAT',0,SD      REGENERATE ADSA
// EXTENT    SYS037,volser,,00942,003
// ASSGN      SYS037,DISK,VOL=volser,SHR
// DLBL      USERUPD,'syntax.file.USERUPD',0,SD      PASSWORD UPDATE
// EXTENT    SYS038,volser,,00945,003
// ASSGN      SYS038,DISK,VOL=volser,SHR
*
// DLBL      SYSIDMS,'work.file.SYSIDMS'
// EXTENT    SYS041,volser
// ASSGN      SYS041,DISK,VOL=volser,SHR
// EXEC      USMGRTR,SIZE=(USMGRTR,400K)
*
* ***** Dictionary Migrator SYNTAX *****
PROCESS,
  RUN=EXPORT,
*
*** IDENTIFY SOURCE DICTIONARY
*
```

```

        IDSOURCE=(USER-ID,PASSWORD),
        DICT=?????????,
*
*   THE OBJECT DICTIONARY WILL NOT BE ACCESSED IN THIS STEP.
*
*   VALUES USED FOR IDOBJECT AND OBJDICT WILL BE PASSED TO THE
*   UPLOAD STEP VIA THE SYNTAX FILES CREATED IN THIS STEP.
*
        IDOBJECT=(USER-ID,PASSWORD),
        OBJDICT=?????????
        .
        .
EXTRACT,.....
        .
        .
EXTRACT,.....
/*
/&
// JOB RESET SYSIPT
    CLOSE SYSIPT,SYSRDR
/&

```

- **your.corelib**—The file id of the z/VSE library into which CA IDMS Dictionary Migrator was installed.
- **idms.corelib**—The file id of the z/VSE library which contains the CA IDMS utility executable phases.
- **volser**—The volume serial number or generic assignment of the disk volume on which the library, as specified in the previous statement, resides.
- **.sublib**—The name of the sublibrary of the z/VSE library specified by the previous filename.
- **work.file**—It is recommended that you retain the corresponding DLBL name within each file-id. Replace *work.file* with names suitable to your site.
- **syntax.file**—It is recommended that you retain the corresponding DLBL name within each file-id. Replace *syntax.file* with names suitable to your site.
- **anyname**—Any suitable name for your SYSIDMS file. Note that the name chosen must be identical to the FILEOUT value in the DITTO copy step.

Import the Extract Tape—z/VSE

The JCL in TOOLJCL library member USMIMPRT.S (z/VSE) imports the extract tape, compares extract to object dictionary, and reports on the comparison.

Note: Suggested space allocations for all work files are contained in the Work Files Table in the chapter "Operations."

```
* ----- Dictionary Migrator STEP ----- 'USMIMPRT' SAMPLE JCL -----
*
*      **** CREATE A SYSIDMS PARAMETER FILE ****
// UPSI  1
// OPTION LOG,PARTDUMP
// DLBL   anyname,'work.file.SYSIDMS',0,SD
// EXTENT SYS041,volser,,,948,1
// ASSGN  SYS041,DISK,VOL=volser,SHR
// EXEC   DITTO
$$DITTO CSQ FILEOUT=anyname
* SYSIDMS parameters.
* For Local Mode specify :
*
DMCL=dmcl-name,LOCAL=ON,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
*
* For CV runs specify :
*
DMCL=dmcl-name,LOCAL=OFF,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
/*
// OPTION PARTDUMP
*      **** z/VSE library where CA IDMS Dictionary Migrator resides
// DLBL   DBMS,'your.corelib'
// EXTENT ,volser
*      **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL   IDMS,'your.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
*      ***** PARAMETER INPUT FOR DICT. MIGRATOR AND IDMS UTILITIES *****
*
// ASSGN  SYS011,YSRDR                               SYNTAX FILE
// ASSGN  SYS010,SYSPCH                               HOLD SYSPCH ASSGN
// DLBL   IJSYSPH,'work.file.sysipt',0,SD             IJSYSPH IS USED TO
// EXTENT          SYSPCH,volser,,,00100,050         CREATE VTOC ENTRY
// ASSGN          SYSPCH,DISK,VOL=volser,SHR         FOR SYSIPT ASSGN
// CLOSE          SYSPCH,SYS010
// DLBL   IJSYSIN,'work.file.sysipt',0,SD             SAME FILE-ID AND
// EXTENT          SYSIPT,volser,,,00100,050         TRACKS AS IJSYSPH
```

```

        ASSGN          SYSIPT,DISK,VOL=volser,SHR      PERMANENT ASSIGNMENT
// DLBL    SYSIPT,'work.file.sysipt',0,SD            SAME FILE-ID AND
// EXTENT          SYS012,volser,,00100,050          TRACKS AS IJSYSPH
// ASSGN          SYS012,DISK,VOL=volser,SHR
*
* **** OUTPUT FROM IDMS UTILITIES AND INPUT TO Dictionary Migrator ***
*
// ASSGN    SYSPCH,IGN
// DLBL    SYSPCH,'work.file.SYSPCH',0,SD
// EXTENT          SYS018,volser,,00150,200
// DLBL    IDMPCH,'work.file.SYSPCH',0,SD            SAME FILE-ID AND
// EXTENT          SYS018,volser,,00150,200          TRACKS AS SYSPCH
// ASSGN          SYS018,DISK,VOL=volser,SHR
*
* ***** REPORT FILE *****
*
// ASSGN    SYS013,SYSLST
*
* ***** Dictionary Migrator WORK FILES *****
*
// DLBL    EXTRACT,'WORK.FILE.EXTRACT'
// EXTENT          SYS015,volser,,00350,200
// ASSGN          SYS015,DISK,VOL=volser,SHR
// DLBL    VSAMEXT,'WORK.FILE.VSAMEXT',0,VSAM
// EXTENT          SYS016,volser
// ASSGN          SYS016,DISK,VOL=volser,SHR
// DLBL    SELECT,'WORK.FILE.SELECT',0,SD
// EXTENT          SYS014,volser,,00550,003
// ASSGN          SYS014,DISK,VOL=volser,SHR
// DLBL    VALDRPT,'WORK.FILE.VALDRPT',0,SD
// EXTENT          SYS017,volser,,00553,007
// ASSGN          SYS017,DISK,VOL=volser,SHR
// DLBL    WORKFIL,'WORK.FILE.WORKFIL',0,SD
// EXTENT          SYS034,volser,,00560,100
// ASSGN          SYS034,DISK,VOL=volser,SHR
// DLBL    WORKFL2,'WORK.FILE.WORKFL2',0,SD
// EXTENT          SYS039,volser,,00660,100
// ASSGN          SYS039,DISK,VOL=volser,SHR
*
* ***** SYNTAX FILES FROM DICT MIGRATOR FOR IDMS UTILITIES *****
*
// DLBL    RHDCDEL,'syntax.file.RHDCDEL',0,SD        MAP & PANEL DELETES
// EXTENT          SYS019,volser,,00760,005
// ASSGN          SYS019,DISK,VOL=volser,SHR
// DLBL    SCHMDEL,'syntax.file.SCHMDEL',0,SD        SCHEMA DELETES
// EXTENT          SYS020,volser,,00765,005
// ASSGN          SYS020,DISK,VOL=volser,SHR
// DLBL    SUBSDEL,'syntax.file.SUBSDEL',0,SD        SUBSCHEMA DELETES
// EXTENT          SYS021,volser,,00770,005

```

```

// ASSGN          SYS021,DISK,VOL=volser,SHR
// DLBL          DDDLDEL,'syntax.file.DDDLDEL',0,SD      ENTITY DELETES
// EXTENT        SYS022,volser,,00775,010
// ASSGN          SYS022,DISK,VOL=volser,SHR
// DLBL          DDDLUPD,'syntax.file.DDDLUPD',0,SD      ENTITY ADD/MODIFIES
// EXTENT        SYS023,volser,,00785,100
// ASSGN          SYS023,DISK,VOL=volser,SHR
// DLBL          BCFUPD,'syntax.file.BCFUPD',0,SD       SQL UPDATES
// EXTENT        SYS040,volser,,00950,003
// ASSGN          SYS040,DISK,VOL=volser,SHR
// DLBL          TABLLOD,'syntax.file.TABLLOD',0,SD     TABLE LOAD MODULES
// EXTENT        SYS024,volser,,00885,005
// ASSGN          SYS024,DISK,VOL=volser,SHR
// DLBL          DDDLLOD,'syntax.file.DDDLLOD',0,SD     ENTITY LOAD MODULES
// EXTENT        SYS025,volser,,00890,005
// ASSGN          SYS025,DISK,VOL=volser,SHR
// DLBL          SCHMUPD,'syntax.file.SCHMUPD',0,SD     ADD SCHEMAS
// EXTENT        SYS026,volser,,00895,005
// ASSGN          SYS026,DISK,VOL=volser,SHR
// DLBL          SUBSUPD,'syntax.file.SUBSUPD',0,SD     ADD SUBSCHEMA
// EXTENT        SYS028,volser,,00905,005
// ASSGN          SYS028,DISK,VOL=volser,SHR
// DLBL          SUBSLOD,'syntax.file.SUBSLOD',0,SD     REGENERATE SUBSCHEMAS
// EXTENT        SYS029,volser,,00910,005
// ASSGN          SYS029,DISK,VOL=volser,SHR
// DLBL          RHDCUPD,'syntax.file.RHDCUPD',0,SD     ADD PANELS & MAPS
// EXTENT        SYS030,volser,,00915,005
// ASSGN          SYS030,DISK,VOL=volser,SHR
// DLBL          RHDCLOD,'syntax.file.RHDCLOD',0,SD    RECOMPILE MAPS
// EXTENT        SYS031,volser,,00920,005
// ASSGN          SYS031,DISK,VOL=volser,SHR
// DLBL          ADSOBN,'syntax.file.ADSOBN',0,SD      REGENERATE DIALOGS
// EXTENT        SYS032,volser,,00930,005
// ASSGN          SYS032,DISK,VOL=volser,SHR
// DLBL          DDDLPGM,'syntax.file.DDDLPGM',0,SD    ADD PROGRAM
// EXTENT        SYS033,volser,,00925,005
// ASSGN          SYS033,DISK,VOL=volser,SHR
// DLBL          ADSBTAT,'syntax.file.ADSBTAT',0,SD    REGENERATE ADSA
// EXTENT        SYS037,volser,,00942,003
// ASSGN          SYS037,DISK,VOL=volser,SHR
// DLBL          USERUPD,'syntax.file.USERUPD',0,SD   PASSWORD UPDATE
// EXTENT        SYS038,volser,,00945,003
// ASSGN          SYS038,DISK,VOL=volser,SHR
*
// DLBL          SYSIDMS,'work.file.SYSIDMS'
// EXTENT        SYS041,volser
// ASSGN          SYS041,DISK,VOL=volser,SHR
// EXEC          USMGRTR,SIZE=(USMGRTR,400K)
* ***** Dictionary Migrator SYNTAX *****

```

```

PROCESS,
  RUN=IMPORT,
*
*
*   IDENTIFY OBJECT DICTIONARY
*
  IDOBJECT=(USER-ID,PASSWORD),
  OBJDICT=????????
      .
      .
/*
/&
// JOB RESET SYSIPT
  CLOSE SYSIPT,SYSRDR
/&

```

- **your.corelib**—The file id of the z/VSE library into which CA IDMS Dictionary Migrator was installed.
- **idms.corelib**—The file id of the z/VSE library which contains the CA IDMS utility executable phases.
- **volser**—The volume serial number or generic assignment of the disk volume on which the library, as specified in the previous statement, resides.
- **.sublib**—The name of the sublibrary of the z/VSE library specified by the previous filename.
- **work.file**—It is recommended that you retain the corresponding DLBL name within each file-id. Replace *work.file* with names suitable to your site.
- **syntax.file**—It is recommended that you retain the corresponding DLBL name within each file-id. Replace *syntax.file* with names suitable to your site.
- **anyname**—Any suitable name for your SYSIDMS file. Note that the name chosen must be identical to the FILEOUT value in the DITTO copy step.

Upload Syntax Files—z/VSE

The JCL in TOOLJCL library member USMLOAD1.S (z/VSE) is used to upload syntax files to the object dictionary using the CA IDMS utilities, or using the CA user password upload utility supplied by CA. There can be from 1 to 16 steps involved in the upload, depending on your site and type of migration.

For more information about uploading syntax files, see the chapter "Operations."

Note that if you want to delete entities from the source dictionary after migration, make a second non-CHANGEONLY run using the same parameters as the real migration, specify the same dictionary for both source and object.

```
* --- USE IDMS TO LOAD OBJECT DICT ----- 'USMLOAD1' SAMPLE JCL ---
*
// JOB RHDCDEL
* *-----*
* *RHDCDEL - DELETE PANELS AND MAPS *
* *-----*
* **** CREATE A SYSIDMS PARAMETER FILE ****
// UPSI 1
// OPTION LOG,PARTDUMP
// DLBL anyname,'work.file.SYSIDMS',0,SD
// EXTENT SYS041,volser,,nnnn,1
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC DITTO
$$DITTO CSQ FILEOUT=anyname
* SYSIDMS parameters.
* For Local Mode specify :
*
DMCL=dmcl-name,LOCAL=ON,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
*
* For CV runs specify :
*
DMCL=dmcl-name,LOCAL=OFF,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
/*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
```



```

// DLBL IJSYSIN,'syntax.file.RHDCDEL',0,SD
// EXTENT SYSIPT,volser
   ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC RHDCMAP1
*
/&
// JOB RESET SYSIPT
   CLOSE SYSIPT,SYSRDR
/&
// JOB SCHMDEL
*-----*
*SCHMDEL - DELETE SCHEMA, AND SUBSCHEMA SOURCE *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL IJSYSIN,'syntax.file.SCHMDEL',0,SD
// EXTENT SYSIPT,volser
   ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC IDMSCHEM
*
/&
// JOB RESET SYSIPT
   CLOSE SYSIPT,SYSRDR
/&
// JOB SUBSDEL
*-----*
*SUBSDEL - DELETE SUBSCHEMA *
*N O T E: THIS IS REQUIRED ONLY IF A SUBSCHEMA WAS MIGRATED *
*          WITHOUT ITS SCHEMA *
* *-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL DBMS,'your.corelib'
// EXTENT ,volser

```

```

*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL    IDMS,'idms.corelib'
// EXTENT  ,volser
*
// LIBDEF  PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL    IJSYSIN,'syntax.file.SUBSDEL',0,SD
// EXTENT  SYSIPT,volser
// ASSGN   SYSIPT,DISK,VOL=volser,SHR
// DLBL    SYSIDMS,'work.file.SYSIDMS'
// EXTENT  SYS041,volser
// ASSGN   SYS041,DISK,VOL=volser,SHR
// EXEC    IDMSUBSC
*
/&
// JOB RESET SYSIPT
// CLOSE  SYSIPT,SYSRDR
/&
// JOB DDDLDEL
*-----*
*DDDLDEL - DELETE OTHER ENTITIES
*-----*
// upsi 00000000
// **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL    DBMS,'your.corelib'
// EXTENT  ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL    IDMS,'idms.corelib'
// EXTENT  ,volser
*
// LIBDEF  PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
* // DLBL    IJSYSIN,'syntax.file.DDDLDEL',0,SD
* // EXTENT  SYSIPT,volser
* // ASSGN   SYSIPT,DISK,VOL=volser,SHR
// DLBL    SYSIDMS,'work.file.SYSIDMS'
// EXTENT  SYS041,volser
// ASSGN   SYS041,DISK,VOL=volser,SHR
// EXEC    IDMSDDL
*
/&
// JOB RESET SYSIPT
// CLOSE  SYSIPT,SYSRDR
/&
// JOB DDDLUPD
*-----*
*DDDLUPD - ADD/MODIFY CLASS, ATTRIBUTE, SYSTEM, RECORD, ELEMENT,
*

```

```

*          MESSAGE, MODULE, AND TABLE ENTITIES          *
*-----*
// upsi 00000000
  **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL   DBMS,'your.corelib'
// EXTENT ,volser
*
  **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL   IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL   IJSYSIN,'syntax.file.DDDLUPD',0,SD
// EXTENT SYSIPT,volser
  ASSGN  SYSIPT,DISK,VOL=volser,SHR
// DLBL   SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN  SYS041,DISK,VOL=volser,SHR
// EXEC   IDMSDDL
*
/&
// JOB RESET SYSIPT
  CLOSE  SYSIPT,SYSRDR
/&
/&
// JOB TABLLOD
*-----*
*TABLLOD - GENERATE TABLE LOAD MODULES          *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL   DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL   IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL   IJSYSIN,'syntax.file.TABLLOD',0,SD
// EXTENT SYSIPT,volser
  ASSGN  SYSIPT,DISK,VOL=volser,SHR
// DLBL   SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN  SYS041,DISK,VOL=volser,SHR
// EXEC   IDMSDDL
*

```

```

/&
// JOB RESET SYSIPT
  CLOSE SYSIPT,SYSRDR
/&
// JOB DDDLLOD
*-----*
*DDLLOD - ADD/MODIFY LOAD MODULES FOR SUBSCHEMA, MAP, MAP EDIT *
*          TABLE, AND DIALOG *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL  DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL  IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL  IJSYSIN,'syntax.file.DDDLLOD',0,SD
// EXTENT SYSIPT,volser
  ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL  SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC  IDMSDDL

/&
// JOB RESET SYSIPT
  CLOSE SYSIPT,SYSRDR
/&
// JOB SCHMUPD
*-----*
*SCHMUPD - ADD SCHEMA SOURCE *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL  DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL  IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL  IJSYSIN,'syntax.file.SCHMUPD',0,SD
// EXTENT SYSIPT,volser

```

```

        ASSGN  SYSIPT,DISK,VOL=volser,SHR
// DLBL    SYSIDMS,'work.file.SYSIDMS'
// EXTENT  SYS041,volser
// ASSGN   SYS041,DISK,VOL=volser,SHR
// EXEC    IDMSCHEM
*
// JOB RESET SYSIPT
        CLOSE  SYSIPT,SYSRDR
/&
// JOB SUBSUPD
*-----*
*SUBSUPD - ADD SUBSCHEMA SOURCE
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL    DBMS,'your.corelib'
// EXTENT  ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL    IDMS,'idms.corelib'
// EXTENT  ,volser
*
// LIBDEF  PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL    IJSYSIN,'syntax.file.SUBSUPD',0,SD
// EXTENT  SYSIPT,volser
        ASSGN  SYSIPT,DISK,VOL=volser,SHR
// DLBL    SYSIDMS,'work.file.SYSIDMS'
// EXTENT  SYS041,volser
// ASSGN   SYS041,DISK,VOL=volser,SHR
// EXEC    IDMSUBSC
*
/&
// JOB RESET SYSIPT
        CLOSE  SYSIPT,SYSRDR
/&
// JOB SUBSLOD
*-----*
*SUBSLOD - GENERATE SUBSCHEMA LOAD MODULE
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL    DBMS,'your.corelib'
// EXTENT  ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL    IDMS,'idms.corelib'
// EXTENT  ,volser
*

```

```
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL IJSYSIN,'syntax.file.SUBSLOD',0,SD
// EXTENT SYSIPT,volser
// ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC IDMSUBSC
*
/&
// JOB RESET SYSIPT
// CLOSE SYSIPT,SYSRDR
/&
// JOB RHDCUPD
*-----*
*RHDCUPD - ADD/MODIFY PANEL AND MAP SYNTAX *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL IJSYSIN,'syntax.file.RHDCUPD',0,SD
// EXTENT SYSIPT,volser
// ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC RHDCMAP1
*
/&
// JOB RESET SYSIPT
// CLOSE SYSIPT,SYSRDR
/&
// JOB RHDCLD
*-----*
*RHDCLD - COMPILE MAP *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL DBMS,'your.corelib'
// EXTENT ,volser
```

```

*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL    IDMS,'idms.corelib'
// EXTENT  ,volser
*
// LIBDEF  PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL    IJSYSIN,'syntax.file.RHDCLOD',0,SD
// EXTENT  SYSIPT,volser
// ASSGN   SYSIPT,DISK,VOL=volser,SHR
// DLBL    SYSIDMS,'work.file.SYSIDMS'
// EXTENT  SYS041,volser
// ASSGN   SYS041,DISK,VOL=volser,SHR
// EXEC    RHDCMPUT
*
/&
// JOB RESET SYSIPT
// CLOSE  SYSIPT,SYSRDR
/&
// JOB ADSOBGN
*-----*
*ADSOBGN - GENERATE ADS DIALOG                                     *
*N O T E: THIS STEP CAN ONLY BE EXECUTED IF DIALOG LOAD MODULES *
*          WERE MIGRATED.                                         *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL    DBMS,'your.corelib'
// EXTENT  ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL    IDMS,'idms.corelib'
// EXTENT  ,volser
*
// LIBDEF  PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL    IJSYSIN,'syntax.file.ADSOBGN',0,SD
// EXTENT  SYSIPT,volser
// ASSGN   SYSIPT,DISK,VOL=volser,SHR
// DLBL    SYSIDMS,'work.file.SYSIDMS'
// EXTENT  SYS041,volser
// ASSGN   SYS041,DISK,VOL=volser,SHR
// EXEC    ADSOBCOM
*
/&
// JOB RESET SYSIPT
// CLOSE  SYSIPT,SYSRDR
/&
// JOB DDDLPGM

```

```

*-----*
*DDDLPGM - ADD/MODIFY PROGRAM SYNTAX                                     *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL   DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL   IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL   IJSYSIN,'syntax.file.DDDLPGM',0,SD
// EXTENT SYSIPT,volser
// ASSGN  SYSIPT,DISK,VOL=volser,SHR
// DLBL   SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN  SYS041,DISK,VOL=volser,SHR
// EXEC   IDMSDDL
*
/&
// JOB RESET SYSIPT
// CLOSE SYSIPT,SYSRDR
/&
*-----*
*ADSBTAT - UPDATE TASK APPLICATION TABLE.                               *
*          IF THERE IS APPLICATION WHICH USES THE REVISED ADSA,        *
*          EXECUTE THIS STEP.                                           *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL   DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL   IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL   IJSYSIN,'syntax.file.ADSBTAT',0,SD
// EXTENT SYSIPT,volser
// ASSGN  SYSIPT,DISK,VOL=volser,SHR
// DLBL   SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN  SYS041,DISK,VOL=volser,SHR
// EXEC   ADSOBTAT

```



```

*
//*
//*-----*
//* BCFUPD - ADD SQL ENTITY DEFINITIONS *
//* *
//*-----*
// upsi 00000000
// **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL IDMS 'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(idms.sublib)
*
// DLBL IJSYSIN,'syntax.file.BCFUPD',0,SD
// EXTENT SYSIPT,volser
// ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC IDMSBCF
/&
// JOB RESET SYSIPT
CLOSE SYSIPT,SYSRDR
/&
// JOB USMULOD
*-----*
*USMULOD - MODIFY MIGRATED USER-PASSWORDS. DO NOT RUN THIS STEP *
* IS DISCONNECT=USER WAS SPECIFIED. *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// ASSGN SYS013,SYSLST
*
// DLBL USERUPD,'syntax.file.USERUPD',0,SD
// EXTENT SYS038,volser
ASSGN SYS038,DISK,VOL=volser,SHR
// DLBL SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC USMULOD,SIZE=USMULOD

```

*
/&

- **your.corelib**—The file id of the z/VSE library into which CA IDMS Dictionary Migrator was installed.
- **idms.corelib**—The file id of the z/VSE library which contains the CA IDMS utility executable phases.
- **volser**—The volume serial number or generic assignment of the disk volume on which the library, as specified in the previous statement, resides.
- **upsi**—Replace 'UPSI 00000000' with the central version UPSI setting appropriate to your site. If you wish to run in local mode, add the JCL appropriate to local mode processing.
- **.sublib**—The name of the sublibrary of the z/VSE library specified by the previous filename.
- **vsam.file**—It is recommended that you retain the DLBL name VSAMEXT within the VSAM file name. Replace *vsam.file* with a name suitable to your site.
- **syntax.file**—The syntax files were created in a previous step or job. Use the file-id's for these files that were used when the files were created.
- **anyname**—Any suitable name for your SYSIDMS file. Note that the name chosen must be identical to the FILEOUT value in the DITTO copy step.
- **nnnn**—The beginning relative track number for SYSIDMS file (1 track required).

Delete Migrated Source from Source Dictionary—z/VSE

The JCL in TOOLJCL library member USMLOAD3.S (z/VSE) is an optional job that, when executed, deletes the migrated source from the source dictionary after migration.

Note: If you want to delete entities from the source dictionary after migration, make a second non-CHANGEONLY run using the same parameters as the real migration, specify the same dictionary for both source and object.

```
*      --- DELETE ENTITIES FROM SOURCE DICT - 'USMLOAD3' SAMPLE JCL ---
*
// JOB RHDCDEL
*-----*
*RHDCDEL - DELETE PANELS AND MAPS
*-----*
*      **** CREATE A SYSIDMS PARAMETER FILE ****
// UPSI 1
// OPTION LOG,PARTDUMP
// DLBL   anyname,'work.file.SYSIDMS',0,SD
// EXTENT SYS041,volser,,nnnn,1
// ASSGN  SYS041,DISK,VOL=volser,SHR
// EXEC   DITTO
$$DITTO CSQ FILEOUT=anyname
* SYSIDMS parameters.
* For Local Mode specify :
*
DMCL=dmcl-name,LOCAL=ON,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
*
* For CV runs specify :
*
DMCL=dmcl-name,LOCAL=OFF,JOURNAL=OFF
FILENAME=SYSIPT,FILETYPE=D,BLKSIZE=80,DEVADDR=SYSIPT
/*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL   DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL   IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL   IJSYSIN,'syntax.file.RHDCDEL',0,SD
// EXTENT SYSIPT,volser
// ASSGN  SYSIPT,DISK,VOL=volser,SHR
// DLBL   SYSIDMS,'work.file.SYSIDMS'
```

```
// EXTENT  SYS041,volser
// ASSGN   SYS041,DISK,VOL=volser,SHR
// EXEC    RHDCMAP1
*
/&
// JOB RESET SYSIPT
      CLOSE  SYSIPT,SYSRDR
/&
// JOB SCHMDEL
*-----*
*SCHMDEL - DELETE SCHEMA, DMCL, AND SUBSCHEMA SOURCE      *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL    DBMS,'your.corelib'
// EXTENT  ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL    IDMS,'idms.corelib'
// EXTENT  ,volser
*
// LIBDEF  PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL    IJSYSIN,'syntax.file.SCHMDEL',0,SD
// EXTENT  SYSIPT,volser
      ASSGN  SYSIPT,DISK,VOL=volser,SHR
// DLBL    SYSIDMS,'work.file.SYSIDMS'
// EXTENT  SYS041,volser
// ASSGN   SYS041,DISK,VOL=volser,SHR
// EXEC    IDMSCHEM
*
/&
// JOB RESET SYSIPT
      CLOSE  SYSIPT,SYSRDR
/&
// JOB SUBSDEL
*-----*
*SUBSDEL - DELETE SUBSCHEMA      *
*N O T E: THIS IS REQUIRED ONLY IF A SUBSCHEMA WAS MIGRATED      *
*          WITHOUT ITS SCHEMA      *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL    DBMS,'your.corelib'
// EXTENT  ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL    IDMS,'idms.corelib'
// EXTENT  ,volser
```

```

*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL IJSYSIN,'syntax.file.SUBSDEL',0,SD
// EXTENT SYSIPT,volser
// ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC IDMSUBSC
*
/&
// JOB RESET SYSIPT
// CLOSE SYSIPT,SYSRDR
/&
// JOB DDDLDEL
*-----*
*DDDLDEL - DELETE OTHER ENTITIES *
*-----*
// upsi 00000000
* **** z/VSE library where CA IDMS Dictionary Migrator resided
// DLBL DBMS,'your.corelib'
// EXTENT ,volser
*
* **** z/VSE library where CA IDMS/DB executable phases reside
// DLBL IDMS,'idms.corelib'
// EXTENT ,volser
*
// LIBDEF PHASE,SEARCH=(dbms.sublib,idms.sublib)
*
// DLBL IJSYSIN,'syntax.file.DDDLDEL',0,SD
// EXTENT SYSIPT,volser
// ASSGN SYSIPT,DISK,VOL=volser,SHR
// DLBL SYSIDMS,'work.file.SYSIDMS'
// EXTENT SYS041,volser
// ASSGN SYS041,DISK,VOL=volser,SHR
// EXEC IDMSDDL
*
/&
// JOB RESET SYSIPT
// CLOSE SYSIPT,SYSRDR
/&

```

- **your.corelib**—The file id of the z/VSE library into which CA IDMS Dictionary Migrator was installed.
- **idms.corelib**—The file id of the z/VSE library which contains the CA IDMS utility executable phases.
- **volser**—The volume serial number or generic assignment of the disk volume on which the library, as specified in the previous statement, resides.
- **upsi**—Replace 'UPSI 00000000' with the central version UPSI setting appropriate to your site. If you wish to run in local mode, add the JCL appropriate to local mode processing.
- **.sublib**—The name of the sublibrary of the z/VSE library specified by the previous filename.
- **syntax.file**—The syntax files were created in a previous step or job. Use the file-id's for these files that were used when the files were created.
- **anyname**—Any suitable name for your SYSIDMS file.
Note: The name chosen must be identical to the FILEOUT value in the DITTO copy step.
- **nnnn**—The beginning relative track number for SYSIDMS file (1 track required).

Allocate the VSAM Work File--z/VM

The USMV SAM EXEC allocates the VSAM work file.

Note: For more information, see the chapter "Operations."

```

/* */
TRACE OFF; SIGNAL ON ERROR
/* */
DELETE          -
                (work.file.vsamext) -
                CLUSTER
DEFINE CLUSTER ( -
                NAME(work.file.vsamext) -
                TRACKS(prialloc secalloc) -
                INDEXED -
                VOLUMES(volser) -
                RECSZ(228 228) -
                KEYS(84 2) -
                usetype -
                )

```

- **work.file.vsamext**—The data set name of the VSAM KSDS work file. It is recommended that the data set name include the work file name VSAMEXT.
- **volser**—The volume serial of the disk that will contain the VSAM KSDS work file.
- **usetype**—The appropriate usage type: REUSE or UNIQUE. If you specify UNIQUE, you must delete and define this cluster prior to each execution of CA IDMS Dictionary Migrator>
- **prialloc secalloc**—The primary and secondary space allocations for the VSAM KSDS work file. A track allocation of 60 primary tracks and 30 secondary tracks (60 30) on a 3350 disk device should be adequate for this file. You may need to change these allocations for your DASD type and migration needs.

Extract Information from the Source Dictionary—z/VM

The USMXTRCT EXEC allocates the work files, allocates the syntax files, extracts information from the source dictionary, compares the extract to the object dictionary, reports on the comparison, and produces the syntax files.

```

/* */
TRACE OFF; SIGNAL ON ERROR
/* */
/* USMXTRCT */
/* */
CA_LOADLIB_FN      = 'yourlib'
IDMS_LOADLIB_FN    = 'idmslib'
SORT_TXTLIB_FN     = 'sortlib'
INPUT_PARM_FN      = 'USMXTRCT'
INPUT_PARM_FT      = 'MIGPARM'
INPUT_PARM_FM      = 'A'
/* */
/* Link and access the Minidisks containing the required librarie(s) */
/* */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'GLOBAL LOADLIB ' CA_LOADLIB_FN IDMS_LOADLIB_FN
'GLOBAL TXTLIB ' SORT_TXTLIB_FN
/* */
/* Create the input parameter file. */
/* */
CALL CREATE_INPUT_PARM_FILE
/* */
/* Product specific files. */
/* */
'FILEDEF SYSVRT PRINTER'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSOUT PRINTER'
'FILEDEF AUDIT PRINTER'
'FILEDEF SELECT DISK USMXTRCT SELECT a'
'FILEDEF SYSIPT DISK USMXTRCT MIGIN a'
'FILEDEF VALDRPT DISK USMXTRCT VALDRPT a4'
'FILEDEF WORKFIL DISK USMXTRCT WORKFIL a4'
'FILEDEF WORKFL2 DISK USMXTRCT WORKFL2 a4'
'FILE EXTRACT DISK USMFILE EXTRACT a4 (RECFM FB LRECL 228 BLOCK 228)'
'FILEDEF RHDCDEL DISK USMFILE RHDCDEL a'
'FILEDEF SCHMDEL DISK USMFILE SCHMDEL a'
'FILEDEF SUBSDEL DISK USMFILE SUBSDEL a'
'FILEDEF DDDLDEL DISK USMFILE DDDLDEL a'
'FILEDEF DDDLUPD DISK USMFILE DDDLUPD a'
'FILEDEF BCFUPD DISK USMFILE BCFUPD a'
'FILEDEF TABLLOD DISK USMFILE TABLLOD a'
'FILEDEF DDDLLOD DISK USMFILE DDDLLOD a'

```



```

'FILEDEF SCHMUPD DISK USMFILE SCHMUPD a'
'FILEDEF SUBSUPD DISK USMFILE SUBSUPD a'
'FILEDEF SUBSLOD DISK USMFILE SUBSLOD a'
'FILEDEF RHDCUPD DISK USMFILE RHDCUPD a'
'FILEDEF RHDCLOD DISK USMFILE RHDCLOD a'
'FILEDEF ADSOBN DISK USMFILE ADSOBN a'
'FILEDEF DDDLPGM DISK USMFILE DDDLPGM a'
'FILEDEF ADSBTAT DISK USMFILE ADSBTAT a'
'FILEDEF USERUPD DISK USMFILE USERUPD a'
'FILEDEF MIGPARM DISK' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM
/* Product specific VSAM files. */
'DLBL IJSYSCT vsam_fm DSN vsamcat'
'DLBL IJSYSUC vsam_fm DSN vsamcat'
'DLBL VSAMEXT vsam_fm DSN vsamext (VSAM'
/* */
/* You must create a file 'MIGR SYSIDMS a' containing the SYSIDMS */
/* parameters you use to specify your runtime environment. */
/* If you wish to run in Local Mode specify the following values : */
/* LOCAL=ON,JOURNAL=OFF,DMCL=dmc1-name */
/* If you wish to run CV an 80 byte blank record will suffice. */
/* */
'FILEDEF SYSIDMS DISK MIGR SYSIDMS a'
/* Insert FILEDEF statements for SORT work space as required by */
/* your SORT product. */
SIGNAL OFF ERROR
SAY 'STARTING CA IDMS/Dictionary Migrator RUN'
'EXECOS OSRJN USMGRTR'
USMXTRCT_RC = RC
IF USMXTRCT_RC > 4
THEN DO
CALL ERROR
END
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMXTRCT LISTING'
'CP SPOOL PRINTER OFF'
SAY 'USMXTRCT FINISHED WITH A RETURN CODE OF' USMXTRCT_RC
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT USMXTRCT_RC
/*+++++++ */
CREATE_INPUT_PARM_FILE:
/*+++++++ */
SIGNAL OFF ERROR
'ERASE' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM
SIGNAL ON ERROR
/* */
PUSH 'FFILE'
PUSH

```

```

PUSH 'EXTRACT,user extract statements'
PUSH 'DICT=srcdict,IDOBJECT=(userid,passwd),OBJDICT=objdict'
PUSH 'PROCESS,RUN=runtype,'
PUSH 'INPUT'
PUSH 'SET LRECL 80'
PUSH 'SET RECFM F'
'XEDIT' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM ,
' (NOPROFILE NOSCREEN NOMSG'
RETURN
/*                                                                    */
/*+++++                                                                    */
ERROR:
/*+++++                                                                    */
ERROR_RC = RC
TRACE OFF; SIGNAL OFF ERROR
SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMXTRCT LISTING'
'CP SPOOL PRINTER OFF'
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT ERROR_RC
/*                                                                    */
                                                                    */

```

- **yourlib**—The file name of the load library into which you downloaded CA IDMS Dictionary Migrator.
- **idmslib**—The file name of the load library containing your CA IDMS SUBSCHEMA and DMCL modules.
- **sortlib**—The file name of the text library containing your sort modules.
- **a**—The file mode for the relevant file.
- **vsam_fm**—The file mode of the accessed minidisk which contains your VSAM catalog(s) and file(s).
- **vsamcat**—The data set name of your VSAM catalog.
- **vsamext**—The data set name of the PAGUTIL VSAM file.

Export Syntax Files--z/VM

The USMEXPRT EXEC allocates the work files, allocates the syntax files, extracts information from the source dictionary and puts it onto disk, and produces syntax files on disk.

```

/* */
TRACE OFF; SIGNAL ON ERROR
/* */
/* USMEXPRT */
CA_LOADLIB_FN      = 'yourlib'
IDMS_LOADLIB_FN    = 'idmslib'
SORT_TXTLIB_FN     = 'sortlib'
INPUT_PARM_FN      = 'USMEXPRT'
INPUT_PARM_FT      = 'MIGPARM'
INPUT_PARM_FM      = 'A'
/* */
/* Link and access the Minidisks containing the required librariе(s) */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'GLOBAL LOADLIB ' CA_LOADLIB_FN IDMS_LOADLIB_FN
'GLOBAL TXTLIB ' SORT_TXTLIB_FN
/* */
/* Create the input parameter file. */
/* */
CALL CREATE_INPUT_PARM_FILE
/* */
/* Product specific files. */
'FILEDEF SYSVRT PRINTER'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSOUT PRINTER'
'FILEDEF AUDIT PRINTER'
'FILEDEF SYSIPT DISK USMEXPRT MIGIN a'
'FILEDEF SYSPCH DISK USMEXPRT MIGPCH a'
'FILEDEF SELECT DISK USMEXPRT SELECT a'
'FILEDEF VALDRPT DISK USMEXPRT VALDRPT a'
'FILEDEF WORKFIL DISK USMEXPRT WORKFIL a'
'FILEDEF WORKFL2 DISK USMEXPRT WORKFL2 a'
'FILE EXTRACT DISK USMFILE EXTRACT a4 (RECFM FB LRECL 228 BLOCK 228)'
'FILEDEF RHDCDEL DISK USMFILE RHDCDEL a'
'FILEDEF SCHMDEL DISK USMFILE SCHMDEL a'
'FILEDEF SUBSDEL DISK USMFILE SUBSDEL a'
'FILEDEF DDDLDEL DISK USMFILE DDDLDEL a'
'FILEDEF DDDLUPD DISK USMFILE DDDLUPD a'
'FILEDEF BCFUPD DISK USMFILE BCFUPD a'
'FILEDEF TABLLOD DISK USMFILE TABLLOD a'
'FILEDEF DDDLLOD DISK USMFILE DDDLLOD a'
'FILEDEF SCHMUPD DISK USMFILE SCHMUPD a'
'FILEDEF SUBSUPD DISK USMFILE SUBSUPD a'

```

```

'FILEDEF SUBSLOD DISK USMFILE SUBSLOD a'
'FILEDEF RHDCUPD DISK USMFILE RHDCUPD a'
'FILEDEF RHDCLD DISK USMFILE RHDCLD a'
'FILEDEF ADSOBN DISK USMFILE ADSOBN a'
'FILEDEF DDDLPGM DISK USMFILE DDDLPGM a'
'FILEDEF ADSBTAT DISK USMFILE ADSBTAT a'
'FILEDEF USERUPD DISK USMFILE USERUPD a'
'FILEDEF MIGPARM DISK' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM
/*
/* You must create a file 'SYSIDMS INPUT a' containing the SYSIDMS */
/* parameters you use to specify your runtime environment. */
/* If you wish to run in Local Mode specify the following values : */
/* LOCAL=ON,JOURNAL=OFF,DMCL=dmc1-name */
/* If you wish to run CV an 80 byte blank record will suffice. */
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
/* Insert FILEDEF statements for SORT work space as required by */
/* your SORT product. */
/*
SAY 'STARTING CA IDMS/Dictionary Migrator EXPORT RUN'
SIGNAL OFF ERROR
'EXECOS OSRJN USMGRTR'
USMEXPRT_RC = RC
IF USMEXPRT_RC > 4
THEN DO
CALL ERROR
END
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMEXPRT LISTING'
'CP SPOOL PRINTER OFF'
SAY 'USMEXPRT FINISHED WITH A RETURN CODE OF' USMEXPRT_RC
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT USMEXPRT_RC
/*+++++ */
CREATE_INPUT_PARM_FILE:
/*+++++ */
SIGNAL OFF ERROR
'ERASE' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM
SIGNAL ON ERROR
/*
PUSH 'FFILE'
PUSH
PUSH 'EXTRACT,user extract statements'
PUSH 'DICT=srcdict,IDOBJECT=(userid,passwd),OBJDICT=objdict'
PUSH 'PROCESS,RUN=EXPORT,'
PUSH 'INPUT'
PUSH 'SET LRECL 80'
PUSH 'SET RECFM F'

```

```
'XEDIT' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM ,
' (NOPROFILE NOSCREEN NOMSG'
RETURN
/*+++++ */
ERROR:
/*+++++ */
ERROR_RC = RC
TRACE OFF; SIGNAL OFF ERROR
SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMEXPRT LISTING'
'CP SPOOL PRINTER OFF'
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT ERROR_RC
/*                                     */
```

- **yourlib**—The file name of the load library into which you downloaded CA IDMS Dictionary Migrator.
- **idmslib**—The file name of the load library containing your CA IDMS SUBSCHEMA and DMCL modules.
- **sortlib**—The file name of the text library containing your sort modules.
- **a**—The file mode for the relevant file.

Import the Extract Tape--z/VM

The USMIMPRT EXEC imports the extract tape, compares extract to object dictionary, and reports on the comparison.

Note: Suggested space allocations for all work files are contained in the Work Files Table in the chapter "Operations."

```

/* */
TRACE OFF; SIGNAL ON ERROR
/* */
/* USMIMPRT */
/* */
CA_LOADLIB_FN      = 'yourlib'
IDMS_LOADLIB_FN   = 'idmslib'
SORT_TXTLIB_FN    = 'sortlib'
INPUT_PARM_FN     = 'USMIMPRT'
INPUT_PARM_FT     = 'MIGPARM'
INPUT_PARM_FM     = 'A'
/* */
/* Link and access the Minidisks containing the required librarie(s) */
/* */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'GLOBAL LOADLIB ' CA_LOADLIB_FN IDMS_LOADLIB_FN
'GLOBAL TXTLIB ' SORT_TXTLIB_FN
/* */
/* Create the input parameter file. */
/* */
CALL CREATE_INPUT_PARM_FILE
/* */
/* Product specific files. */
/* */
'FILEDEF SYSVRT PRINTER'
'FILEDEF SYSYST PRINTER'
'FILEDEF SYSOUT PRINTER'
'FILEDEF AUDIT PRINTER'
'FILEDEF SELECT DUMMY'
'FILEDEF SYSIPT DISK USMIMPRT MIGIN a'
'FILEDEF VALDRPT DISK USMIMPRT VALDRPT a'
'FILEDEF WORKFIL DISK USMIMPRT WORKFIL a'
'FILEDEF WORKFL2 DISK USMIMPRT WORKFL2 a'
'FILEDEF EXTRACT DISK USMFILE EXTRACT a'
'FILEDEF RHDCDEL DISK USMFILE RHDCDEL a'
'FILEDEF SCHMDEL DISK USMFILE SCHMDEL a'
'FILEDEF SUBSDEL DISK USMFILE SUBSDEL a'
'FILEDEF DDDLDEL DISK USMFILE DDDLDEL a'
'FILEDEF DDDLUPD DISK USMFILE DDDLUPD a'
'FILEDEF BCFUPD DISK USMFILE BCFUPD a'

```

```

'FILEDEF TABLLOD DISK USMFILE TABLLOD a'
'FILEDEF DDDLLOD DISK USMFILE DDDLLOD a'
'FILEDEF SCHMUPD DISK USMFILE SCHMUPD a'
'FILEDEF SUBSUPD DISK USMFILE SUBSUPD a'
'FILEDEF SUBSLOD DISK USMFILE SUBSLOD a'
'FILEDEF RHDCUPD DISK USMFILE RHDCUPD a'
'FILEDEF RHDCLOD DISK USMFILE RHDCLOD a'
'FILEDEF ADSOBN DISK USMFILE ADSOBN a'
'FILEDEF DDDLPGM DISK USMFILE DDDLPGM a'
'FILEDEF ADSBTAT DISK USMFILE ADSBTAT a'
'FILEDEF USERUPD DISK USMFILE USERUPD a'
'FILEDEF MIGPARM DISK' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM
/*
/* Product specific VSAM files.
/*
/*
'DLBL IJSYSCT vsam_fm DSN vsamcat'
'DLBL IJSYSUC vsam_fm DSN vsamcat'
'DLBL VSAMEXT vsam_fm DSN vsamext (VSAM'
/*
/* You must create a file 'SYSIDMS INPUT a' containing the SYSIDMS
/* parameters you use to specify your runtime environment.
/*
/* If you wish to run in Local Mode specify the following values :
/* LOCAL=ON,JOURNAL=OFF,DMCL=dmc1-name
/*
/* If you wish to run CV an 80 byte blank record will suffice.
/*
/*
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
/*
/* Insert FILEDEF statements for SORT work space as required by
/* your SORT product.
/*
/*
SAY 'STARTING CA IDMS/Dictionary Migrator IMPORT'
SIGNAL OFF ERROR
'EXECOS OSRJN USMGRTR'
USMIMPRT_RC = RC
IF USMIMPRT_RC > 4
THEN DO
CALL ERROR
END
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMIMPRT LISTING'
'CP SPOOL PRINTER OFF'
SAY 'USMIMPRT FINISHED WITH A RETURN CODE OF' USMIMPRT_RC
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT USMIMPRT_RC
/*+++++
CREATE_INPUT_PARM_FILE:
/*+++++

```

```

SIGNAL OFF ERROR
'ERASE' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM
SIGNAL ON ERROR
/*
PUSH 'FFILE'
PUSH
PUSH 'OBJDICT=objdict'
PUSH 'PROCESS,RUN=IMPORT,'
PUSH 'INPUT'
PUSH 'SET LRECL 80'
PUSH 'SET RECFM F'
'XEDIT' INPUT_PARM_FN INPUT_PARM_FT INPUT_PARM_FM ,
' (NOPROFILE NOSCREEN NOMSG'
RETURN
/*
/*+++++*/
ERROR:
/*+++++*/
ERROR_RC = RC
TRACE OFF; SIGNAL OFF ERROR
SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMIMPRT LISTING'
'CP SPOOL PRINTER OFF'
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT ERROR_RC
/*

```

- **yourlib**—The file name of the load library into which you downloaded CA IDMS Dictionary Migrator.
- **idmslib**—The file name of the load library containing your CA IDMS SUBSCHEMA and DMCL modules.
- **sortlib**—The file name of the text library containing your sort modules.
- **a**—The file mode for the relevant file.
- **vsam_fm**—The file mode of the accessed minidisk which contains your VSAM catalog(s) and file(s).
- **vsamcat**—The data set name of your VSAM catalog.
- **vsamext**—The data set name of the PAGUTIL VSAM file.

Upload Syntax Files—z/VM

The USMLOAD1 EXEC is used to upload syntax files to the object dictionary using CA IDMS utilities, or using the CA user password upload utility supplied by CA. There can be from 1 to 16 steps involved in the upload, depending on your site and type of migration.

More information:

For more information about uploading syntax files, see Syntax Files Table in the chapter "Operations."

If you want to delete entities from the source dictionary after migration, make a second non-CHANGEONLY run using the same parameters as the real migration, specify the same dictionary for both source and object.

```

/* */
TRACE OFF; SIGNAL ON ERROR
/* */
/* USMLOAD1 */
/* */
CA_LOADLIB_FN      = 'yourlib'
IDMS_LOADLIB_FN    = 'idmslib'
/* */
/* Link and access the Minidisks containing the required librarie(s) */
/* */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'GLOBAL LOADLIB ' CA_LOADLIB_FN IDMS_LOADLIB_FN
/* */
MAX_RC = 0
/* */
CALL RHDCDEL
IF RHDCDEL_RC > 0
  THEN SAY 'RHDCDEL STEP ENDED WITH RETURN CODE' RHDCDEL_RC
IF RHDCDEL_RC > MAX_RC
  THEN MAX_RC = RHDCDEL_RC
/* */
CALL SCHMDEL
IF RHDCDEL_RC > 0
  THEN SAY 'SCHMDEL STEP ENDED WITH RETURN CODE' SCHMDEL_RC
IF SCHMDEL_RC > MAX_RC
  THEN MAX_RC = SCHMDEL_RC
/* */
CALL SUBSDEL
IF SUBSDEL_RC > 0
  THEN SAY 'SUBSDEL STEP ENDED WITH RETURN CODE' SUBSDEL_RC
IF SUBSDEL_RC > MAX_RC
  THEN MAX_RC = SUBSDEL_RC
/* */

```

```
CALL DDDLDEL
IF DDDLDEL_RC > 0
    THEN SAY 'DDLDEL STEP ENDED WITH RETURN CODE' DDDLDEL_RC
IF DDDLDEL_RC > MAX_RC
    THEN MAX_RC = DDDLDEL_RC
/*
CALL DDDLUPD
IF DDDLUPD_RC > 0
    THEN SAY 'DDLUPD STEP ENDED WITH RETURN CODE' DDDLUPD_RC
IF DDDLUPD_RC > MAX_RC
    THEN MAX_RC = DDDLUPD_RC
/*
CALL TABLLOD
IF TABLLOD_RC > 0
    THEN SAY 'TABLLOD STEP ENDED WITH RETURN CODE' TABLLOD_RC
IF TABLLOD_RC > MAX_RC
    THEN MAX_RC = TABLLOD_RC
/*
CALL DDDLLOD
IF DDDLLOD_RC > 0
    THEN SAY 'DDLLOD STEP ENDED WITH RETURN CODE' DDDLLOD_RC
IF DDDLLOD_RC > MAX_RC
    THEN MAX_RC = DDDLLOD_RC
/*
CALL SCHMUPD
IF SCHMUPD_RC > 0
    THEN SAY 'SCHMUPD STEP ENDED WITH RETURN CODE' SCHMUPD_RC
IF SCHMUPD_RC > MAX_RC
    THEN MAX_RC = SCHMUPD_RC
/*
CALL SUBSUPD
IF SUBSUPD_RC > 0
    THEN SAY 'SUBSUPD STEP ENDED WITH RETURN CODE' SUBSUPD_RC
IF SUBSUPD_RC > MAX_RC
    THEN MAX_RC = SUBSUPD_RC
/*
CALL SUBSLOD
IF SUBSLOD_RC > 0
    THEN SAY 'SUBSLOD STEP ENDED WITH RETURN CODE' SUBSLOD_RC
IF SUBSLOD_RC > MAX_RC
    THEN MAX_RC = SUBSLOD_RC
/*
CALL RHDCUPD
IF RHDCUPD_RC > 0
    THEN SAY 'RHDCUPD STEP ENDED WITH RETURN CODE' RHDCUPD_RC
IF RHDCUPD_RC > MAX_RC
    THEN MAX_RC = RHDCUPD_RC
/*
CALL RHDCLOD
```

```

IF RHDCLD_RC > 0
  THEN SAY 'RHDCLD STEP ENDED WITH RETURN CODE' RHDCLD_RC
IF RHDCLD_RC > MAX_RC
  THEN MAX_RC = RHDCLD_RC
/*
CALL ADSOBN
IF ADSOBN_RC > 0
  THEN SAY 'ADSOBN STEP ENDED WITH RETURN CODE' ADSOBN_RC
IF ADSOBN_RC > MAX_RC
  THEN MAX_RC = ADSOBN_RC
/*
CALL DDDLPGM
IF DDDLPGM_RC > 0
  THEN SAY 'DDLPGM STEP ENDED WITH RETURN CODE' DDDLPGM_RC
IF DDDLPGM_RC > MAX_RC
  THEN MAX_RC = DDDLPGM_RC
/*
CALL ADSBTAT
IF ADSBTAT_RC > 0
  THEN SAY 'ADSBTAT STEP ENDED WITH RETURN CODE' ADSBTAT_RC
IF ADSBTAT_RC > MAX_RC
  THEN MAX_RC = ADSBTAT_RC
/*
CALL BCFUPD
IF BCFUPD_RC > 0
  THEN SAY 'BCFUPD STEP ENDED WITH RETURN CODE' BCFUPD_RC
IF BCFUPD_RC > MAX_RC
  THEN MAX_RC = BCFUPD_RC
/*
CALL USERUPD
IF USERUPD_RC > 0
  THEN SAY 'USERUPD STEP ENDED WITH RETURN CODE' USERUPD_RC
IF USERUPD_RC > MAX_RC
  THEN MAX_RC = USERUPD_RC
/*
'GLOBAL LOADLIB'
'FILEDEF * CLEAR'
EXIT MAX_RC
/* ----- */
/* RHDCDEL - DELETE PANELS AND MAPS
RHDCDEL:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE RHDCDEL a'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR

```

```

SAY 'STARTING DELETION OF MAPS AND PANELS'
'EXECOS OSRUN RHDCMAP1'
RHDCDEL_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME RHDCDEL LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* SCHMDEL - DELETE SCHEMA AND SUBSCHEMA SOURCE */
SCHMDEL:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE SCHMDEL a'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING DELETION OF SCHEMA AND SUBSCHEMA SOURCE'
'EXECOS OSRUN IDMSCHEM'
IDMSCHEM_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME IDMSCHEM LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* SUBSDEL - DELETE SUBSCHEMA */
/* NOTE: THIS IS ONLY REQUIRED IF A SUBSCHEMA WAS MIGRATED */
/* WITHOUT ITS SCHEMA */
SUBSDEL:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE SUBSDEL a'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING DELETION OF SUBSCHEMA SOURCE'
'EXECOS OSRUN IDMSUBSC'
IDMSUBSC_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME IDMSUBSC LISTING'
'CP SPOOL PRINTER OFF'

```

```

' FILEDEF * CLEAR'
RETURN
/* ----- */
/* DDDLDEL - DELETE OTHER ENTITIES */
/* NOTE: DO NOT RUN THIS STEP FOR CHANGEONLY PROCESSING */
DDLDEL:
/* ----- */
' CP SPOOL PRINTER NOCONT CLOSE'
' CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
' FILEDEF SYSLST PRINTER'
' FILEDEF SYSPCH DUMMY'
' FILEDEF SYSIPT DISK FILE DDDLDEL a'
' FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING DELETION OF DDDL ENTITIES'
' EXECOS OSRUN IDMSDDL'
IDMSDDL_RC = RC
SIGNAL ON ERROR
' CP SPOOL PRINTER NOCONT'
' CP CLOSE PRINTER NAME IDMSDDL LISTING'
' CP SPOOL PRINTER OFF'
' FILEDEF * CLEAR'
RETURN
/* ----- */
/* DDDLUPD - ADD/MODIFY CLASS, ATTRIBUTE, SYSTEM, RECORD, ELEMENT, */
/* MESSAGE, MODULE, AND TABLE ENTITIES */
DDLUPD:
/* ----- */
' CP SPOOL PRINTER NOCONT CLOSE'
' CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
' FILEDEF SYSLST PRINTER'
' FILEDEF SYSPCH DUMMY'
' FILEDEF SYSIPT DISK FILE DDDLUPD A'
' FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING ADD/MODIFY OF DDDL ENTITIES'
' EXECOS OSRUN IDMSDDL'
DDLUPD_RC = RC
SIGNAL ON ERROR
' CP SPOOL PRINTER NOCONT'
' CP CLOSE PRINTER NAME DDDLUPD LISTING'
' CP SPOOL PRINTER OFF'
' FILEDEF * CLEAR'
RETURN
/* ----- */
/* TABLLOD - GENERATE TABLE LOAD MODULES */
TABLLOD:
/* ----- */

```

```

'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY' 'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE TABLLOD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING GENERATION OF TABLE LOAD MODULES'
'EXECOS OSRJN IDMSDDL'
TABLLOD_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME TABLLOD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/*                                                                 */
/* -----                                                        */
/* DDDLLOD - ADD/MODIFY LOAD MODULES FOR SUBSCHEMA, MAP, MAP EDIT */
/*           TABLE AND DIALOG                                     */
/*                                                                 */
DDLLOD:
/* -----                                                        */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE DDDLLOD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING ADD/MODIFY OF DDDL LOAD MODULES'
'EXECOS OSRJN IDMSDDL'
DDLLOD_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME DDDLLOD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/*                                                                 */
/* -----                                                        */
/* SCHMUPD - ADD SCHEMA SOURCE                                     */
/*                                                                 */
SCHMUPD:
/* -----                                                        */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE SCHMUPD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'

```

```

SIGNAL OFF ERROR
SAY 'STARTING ADD OF SCHEMA SOURCE'
'EXECOS OSRUN IDMSCHEM'
SCHMUPD_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME SCHMUPD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* RHDCLD - COMPILE MAP ----- */
RHDCLD:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE RHDCLD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING MAP COMPILE'
'EXECOS OSRUN RHDCLD'
RHDCLD_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME RHDCLD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* ADSOBN - GENERATE ADS DIALOG ----- */
/* NOTE: THIS STEP CAN ONLY BE EXECUTED IF DIALOG MODULES ----- */
/* WERE MIGRATED ----- */
ADSOBN:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE ADSOBN A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING GENERATION OF ADS DIALOGS'
'EXECOS OSRUN ADSOBN'
ADSOBN_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'

```

```

'CP CLOSE PRINTER NAME ADSOBN LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* DDDLPGM - ADD/MODIFY PROGRAM SYNTAX */
DDLPGM:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE DDDLPGM A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING ADD/MODIFY OF PROGRAM SYNTAX'
'EXECOS OSRJN IDMSDDL'
DDLPGM_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME DDDLPGM LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* ----- */
/* SUBSUPD - ADD SUBSCHEMA SOURCE */
SUBSUPD:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE SUBSUPD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING ADD OF SUBSCHEMA SOURCE'
'EXECOS OSRJN IDMSUBSC'
SUBSUPD_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME SUBSUPD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* ----- */
/* SUBSLOD - GENERATE SUBSCHEMA LOAD MODULE */
SUBSLOD:
/* ----- */

```



```

'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE SUBSLOD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING GENERATION OF SUBSCHEMA LOAD MODULE'
'EXECOS OSRJN IDMSUBSC'
SUBSLOD_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME SUBSLOD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* RHDCUPD - ADD/MODIFY PANEL AND MAP SYNTAX */
RHDCUPD:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE RHDCUPD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING ADD/MODIFY OF PANEL AND MAP SYNTAX'
'EXECOS OSRJN RHDCMAP1'
RHDCUPD_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME RHDCUPD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* ADSBTAT - UPDATE TASK APPLICATION TABLE */
/* NOTE: THIS STEP IS ONLY FOR APPLICATIONS WHICH USE THE */
/* REVISD ADSA. */
ADSBTAT:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE ADSBTAT A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'

```

```

SIGNAL OFF ERROR
SAY 'STARTING UPDATE OF TASK APPLICATION TABLE'
'EXECOS OSRUN ADSBTAT'
ADSBTAT_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME ADSBTAT LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* BCFUPD - ADD SQL ENTITY DEFINITIONS */
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE BCFUPD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING UPLOAD OF SQL ENTITY DEFINITIONS'
'EXECOS OSRUN IDMSBCF'
BCFUPD_RC = RC
SIGNAL ON ERROR
SAY 'BCFUPD STEP ENDED WITH RETURN CODE' BCFUPD_RC
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMSQLOD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* USERUPD - UPDATE USER PASSWORDS */
/* NOTE: THIS STEP IS ONLY FOR MIGRATIONS WHICH DO NOT USE */
/* DISCONNECT=(USER). THE PASSWORD FROM THE SOURCE */
/* DICTIONARY IS PLACED INTO THE OBJECT DICTIONARY */
/* IF THIS STEP IS NOT RUN, THE PASSWORD ASSIGNED TO ALL */
/* MIGRATED USERS WILL BE "DBMSINT" */
USERUPD:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE USERUPD A'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING UPDATE OF USER PASSWORDS'
'EXECOS OSRUN USMULOD'
USERUPD_RC = RC

```

```

SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USERUPD LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/*+++++ */
ERROR:
/*+++++ */
  ERROR_RC = RC
  TRACE OFF; SIGNAL OFF ERROR
/*
  SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
/*
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMLOAD LISTING'
'CP SPOOL PRINTER OFF'
'GLOBAL LOADLIB'
'FILEDEF * CLEAR'
/*
EXIT ERROR_RC
/*

```

- **yourlib**—The file name of the load library into which you downloaded CA IDMS Dictionary Migrator.
- **idmslib**—The file name of the load library containing your CA IDMS SUBSCHEMA and DMCL modules.
- **a**—The file mode for the relevant file.

Delete Migrated Source from Source Dictionary - z/VM

The USMLOAD3 EXEC is an optional job that, when executed, deletes the migrated source from the source dictionary after migration.

Note: If you want to delete entities from the source dictionary after migration, make a second non-CHANGEONLY run using the same parameters as the real migration, specify the same dictionary for both source and object.

```
/* */
TRACE OFF; SIGNAL ON ERROR
/* */
/* USMLOAD3 */
/* */
/* */
CA_LOADLIB_FN      = 'yourlib'
IDMS_LOADLIB_FN    = 'idmslib'
/* */
/* Link and access the Minidisks containing the required librarie(s) */
/* */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'GLOBAL LOADLIB ' CA_LOADLIB_FN IDMS_LOADLIB_FN
/* */
MAX_RC = 0
/* */
CALL RHDCDEL
IF RHDCDEL_RC > 0
  THEN SAY 'RHDCDEL STEP ENDED WITH RETURN CODE' RHDCDEL_RC
IF RHDCDEL_RC > MAX_RC
  THEN MAX_RC = RHDCDEL_RC
/* */
CALL SCHMDEL
IF SCHMDEL_RC > 0
  THEN SAY 'SCHMDEL STEP ENDED WITH RETURN CODE' SCHMDEL_RC
IF SCHMDEL_RC > MAX_RC
  THEN MAX_RC = SCHMDEL_RC
/* */
CALL SUBSDEL
IF SUBSDEL_RC > 0
  THEN SAY 'SUBSDEL STEP ENDED WITH RETURN CODE' SUBSDEL_RC
IF SUBSDEL_RC > MAX_RC
  THEN MAX_RC = SUBSDEL_RC
/* */
CALL DDDLDEL
IF DDDLDEL_RC > 0
  THEN SAY 'DDLDEL STEP ENDED WITH RETURN CODE' DDDLDEL_RC
IF DDDLDEL_RC > MAX_RC
  THEN MAX_RC = DDDLDEL_RC
```

```

/*                                                                 */
'GLOBAL LOADLIB'
'FILEDEF * CLEAR'
EXIT MAX_RC
/*                                                                 */
/* ----- */
/* DELETE PANELS AND MAPS */
RHCDEL:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE RHCDEL a'
'FILEDEF SYSIDMS DISK SYSIDMS INPUT a'
SIGNAL OFF ERROR
SAY 'STARTING DELETE OF PANELS AND MAPS'
'EXECOS OSRJN RHCDEL'
RHCDEL_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME RHCDEL LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* DELETE SCHEMA AND SUBSCHEMA SOURCE */
/* N O T E   IN RELEASE 10, USE IDMSCHEM TO DELETE SCHEMAS */
SCHMDEL:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE SCHMDEL a'
'FILEDEF SYSIDMS DISK SYSIDMS PARMS a'
SIGNAL OFF ERROR
SAY 'STARTING DELETE OF SCHEMA AND SUBSCHEMA SOURCE'
'EXECOS OSRJN IDMSDDL'
SCHMDEL_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME SCHMDEL LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/* ----- */
/* DELETE SCHEMA AND SUBSCHEMA SOURCE */
/* N O T E:  THIS IS REQUIRED ONLY IF A SUBSCHEMA WAS MIGRATED */

```

```

/*          WITHOUT ITS SCHEMA          */
SUBSDEL:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE SUBSDEL a'
'FILEDEF SYSIDMS DISK SYSIDMS PARS a'
SIGNAL OFF ERROR
SAY 'STARTING DELETE OF SUBSCHEMA SOURCE'
'EXECOS OSRJN IDMSUBSC'
SUBSDEL_RC = RC
SIGNAL OFF ERROR
SAY 'STARTING DELETE OF SUBSCHEMA SOURCE'
'EXECOS OSRJN IDMSUBSC'
SUBSDEL_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME SUBSDEL LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/*          */
/* ----- */
/* DELETE RECORD, ELEMENT AND TABLE ENTITIES */
DDDLDEL:
/* ----- */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'FILEDEF SYSLST PRINTER'
'FILEDEF SYSPCH DUMMY'
'FILEDEF SYSIPT DISK FILE DDDLDEL a'
'FILEDEF SYSIDMS DISK SYSIDMS PARS a'
SIGNAL OFF ERROR
SAY 'STARTING DELETE OF RECORD, ELEMENT AND TABLE ENTITIES'
'EXECOS OSRJN IDMSDDL'
DDDLDEL_RC = RC
SIGNAL ON ERROR
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME DDDLDEL LISTING'
'CP SPOOL PRINTER OFF'
'FILEDEF * CLEAR'
RETURN
/*          */
/*+++++*/
ERROR:
/*+++++*/
ERROR_RC = RC

```

```
TRACE OFF; SIGNAL OFF ERROR
/*                                                                    */
SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
/*                                                                    */
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USMLOAD LISTING'
'CP SPOOL PRINTER OFF'
'GLOBAL LOADLIB'
'FILEDEF * CLEAR'
/*                                                                    */
EXIT ERROR_RC
/*                                                                    */
                                                                    */
```

- **yourlib**—The file name of the load library into which you downloaded CA IDMS Dictionary Migrator.
- **idmslib**—The file name of the load library containing your CA IDMS SUBSCHEMA and DMCL modules.
- **a**—The file mode for the relevant file.

Appendix B: Extract Paths

This appendix contains the following topics:

This section contains the following topics:

[Introduction](#) (see page 297)

[Extract Path for DIALOG](#) (see page 298)

[Extract Path for RECORD](#) (see page 301)

[Extract Path for RECORD when LEVEL = ENTITY](#) (see page 305)

[Extract Path for ATTRIBUTE when LEVEL = DIALOG](#) (see page 308)

[EXTRACT Path for ATTRIBUTE when LEVEL = ENTITY](#) (see page 311)

[Extract Path for ATTRIBUTE when LEVEL = ONLY](#) (see page 315)

[Extract Path for SUBSCHEMA when LEVEL = DIALOG](#) (see page 317)

[Extract Path for SUBSCHEMA when LEVEL = ENTITY](#) (see page 321)

[Extract Path for APPLICATION when LEVEL = DIALOG](#) (see page 324)

Introduction

This appendix describes and illustrates the paths CA IDMS Dictionary Migrator takes when extracting entities from the source dictionary.

CA IDMS Dictionary Migrator extracts entities from the source dictionary based on the entity type and the name specified in an EXTRACT statement. The entity-namemask can be the name of any occurrence of the entity type, or mask using wild card characters for the entity type. Do not, however, use more than the maximum IDD characters.

This appendix provides descriptions and diagrams of the paths that CA IDMS Dictionary Migrator follows for these entities:

```
EXTRACT, DIALOG=dialog, LEVEL=DIALOG
EXTRACT, RECORD=record-name, LEVEL=DIALOG
EXTRACT, RECORD=record-name, LEVEL=ENTITY
EXTRACT, ATTRIBUTE=attribute-name, LEVEL=DIALOG
EXTRACT, ATTRIBUTE=attribute-name, LEVEL=ENTITY
EXTRACT, ATTRIBUTE=attribute-name, LEVEL=ONLY
EXTRACT, SUBSCHEMA=subschema-name, LEVEL=DIALOG
EXTRACT, SUBSCHEMA=subschema-name, LEVEL=ENTITY
EXTRACT, APPLICATION=application-name, LEVEL=DIALOG
```

Symbol	Definition
--(dotted line)	Path from Entry Point to Starting Point of Extraction.
_ (solid line)	Path from Starting Point to all extracted components.
*	Points to where the Extract Statement goes.

Extract Path for DIALOG

This section describes the EXTRACT path for DIALOG.

EXTRACT, LEVEL = DIALOG, DIALOG = dialog-name

The following is extracted:

For each dialog named in the EXTRACT statement:

1. Extract the dialog
 - Extract the corresponding dialog load module from DDLDCLOUD (load) area
2. For each process module connected to the dialog and for each included process module within the process module
 - Extract the process module source
 - For each message code in the process module source
 - Extract the message from DDLDCMSG (message) area
 - For each referenced dialog within the process module source
 - Extract the dialog and its components (start at step 1. above)
3. For each record connected to the dialog
 - Extract the record
 - For each element connected to the dialog record
 - Extract the dialog record element and all of its subordinate elements
 - For each IDD defined file connected to the record
 - Extract the file
4. For each file used by the dialog
 - Extract the file

5. For the subschema connected to the dialog
 - Extract the schema connected to the subschema
 - For each record connected to the schema
 - Extract the schema record
 - For each element connected to the schema record
 - Extract the schema record element and all of its subordinate elements
 - For each subschema connected to the schema
 - Extract the subschema
 - Extract the corresponding subschema load module from DDLDCLOAD (load) area
 - Extract each Logical Record Facility work record used by the subschema
 - For each record connected to the subschema
 - Extract the Record
 - Extract the subschema record element and all of its subordinate elements.
6. Extract the map connected to the dialog

Extract the corresponding load module map from DDLDCLOAD (load) area

For each message code in the map

 - Extract message from DDLDCMSG (message) area

For each edit and/or code table used in the map

 - Extract the table
 - Extract the corresponding table load module from DDLDCLOAD (load) area

For each work record used in the map

 - Extract the map record
 - For each element connected to the map record
 - Extract the map record element and all of its subordinate elements

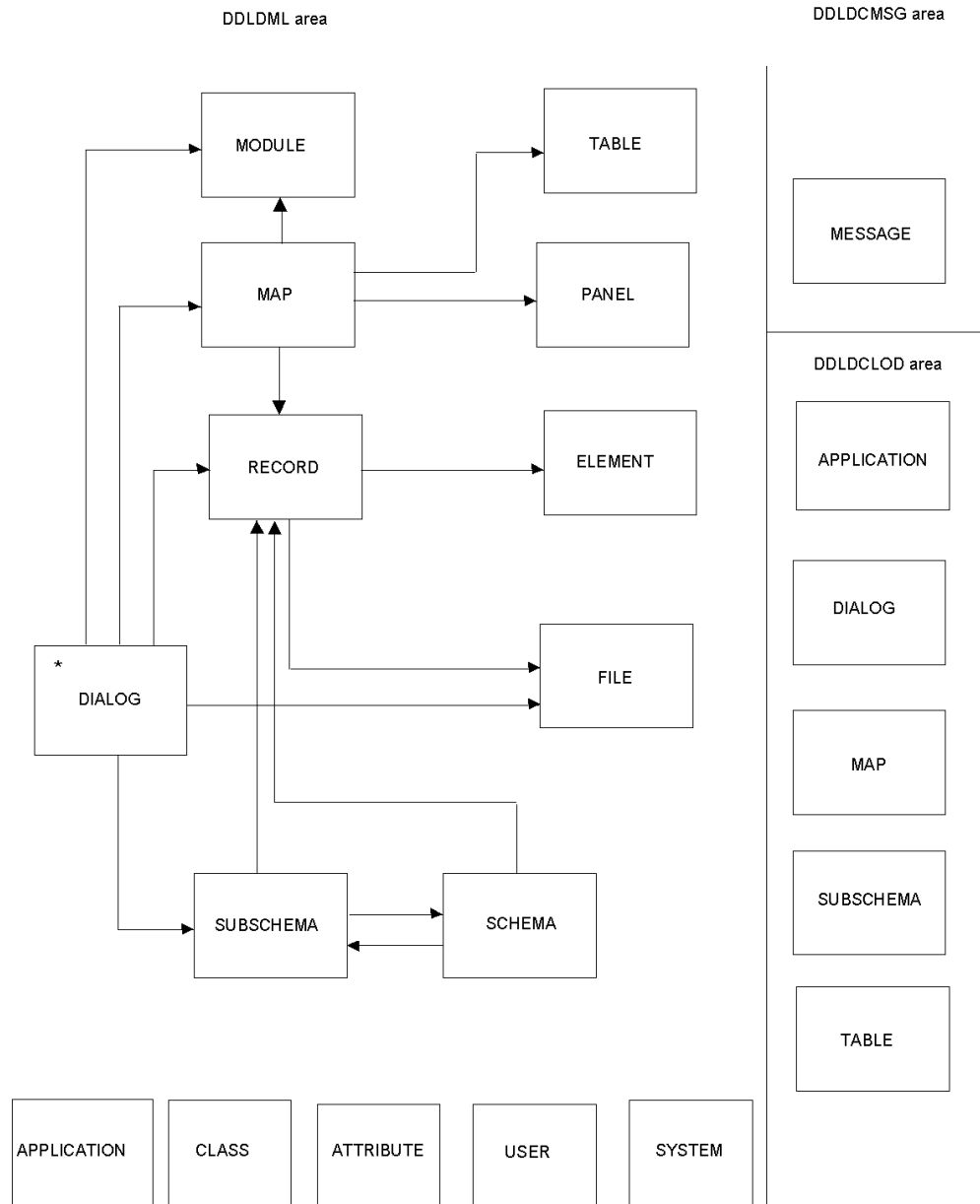
For the panel connected to the map

- Extract the panel

For each help module connected to the map

- Extract the module source
- Extract the map help load module

All attributes, systems and users associated with every entity are also extracted.



Extract Path for RECORD

This section describes the EXTRACT path for RECORD.

EXTRACT, LEVEL = DIALOG, RECORD = record-name

The following is extracted:

For each dialog that uses the specified record:

1. Extract the dialog

Extract the corresponding dialog load module from DDLDCLOAD (load) area

2. For each module connected to the dialog and for each included process module within the process module

- Extract the process module source
- For each message code in the process module source
 - Extract the message from DDLDCMSG (message) area

For each referenced dialog within the process module source

- Extract the dialog and its components (start at step 1. above)

3. For each record connected to the dialog

- Extract the record
- For each element connected to the dialog record
 - Extract the dialog record element and all of its subordinate elements

For each IDD defined file connected to the record

- Extract the file

4. For each file used by the dialog

- Extract the file

5. For the subschema connected to the dialog

- Extract the schema connected to the subschema
- For each record connected to the schema
 - Extract the schema record
 - For each element connected to the schema record

-Extract the schema record element and all of its subordinate elements

For each subschema connected to the schema

- Extract the subschema
- Extract the corresponding subschema load module from DDLDCLOAD (load) area
- Extract each Logical Record Facility work record used by the subschema
- For each record connected to the subschema
 - Extract the Record
 - Extract the subschema record element and all of its subordinate elements.

6. Extract the map connected to the dialog

Extract the corresponding load module map from DDLDCLOUD (load) area

For each message code in the map

- Extract message from DDLDCMSG (message) area

For each edit and/or code table used in the map

- Extract the table
- Extract the corresponding table load module from DDLDCLOUD (load) area

For each work record used in the map

- Extract the map record
- For each element connected to the map record
 - Extract the map record element and all of its subordinate elements

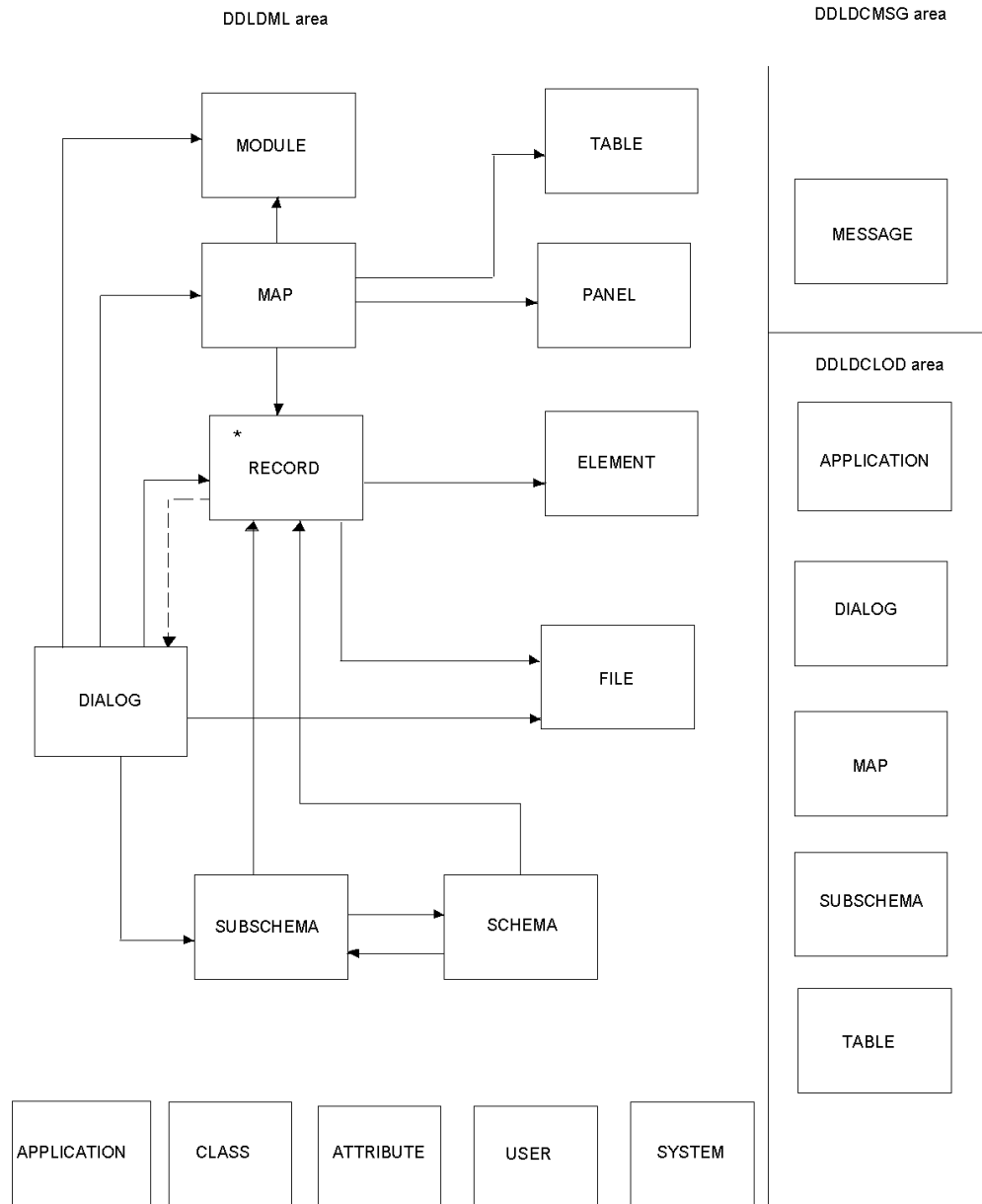
For the panel connected to the map

- Extract the panel

For each help module connected to the map

- Extract the module source
- Extract the map help load module

All attributes, systems and users associated with every entity are also extracted.



Extract Path for RECORD when LEVEL = ENTITY

This section describes the EXTRACT path for RECORD, when LEVEL = ENTITY..

EXTRACT, RECORD = record-name, LEVEL = ENTITY

For the record specified in the EXTRACT statement:

1. Extract the record

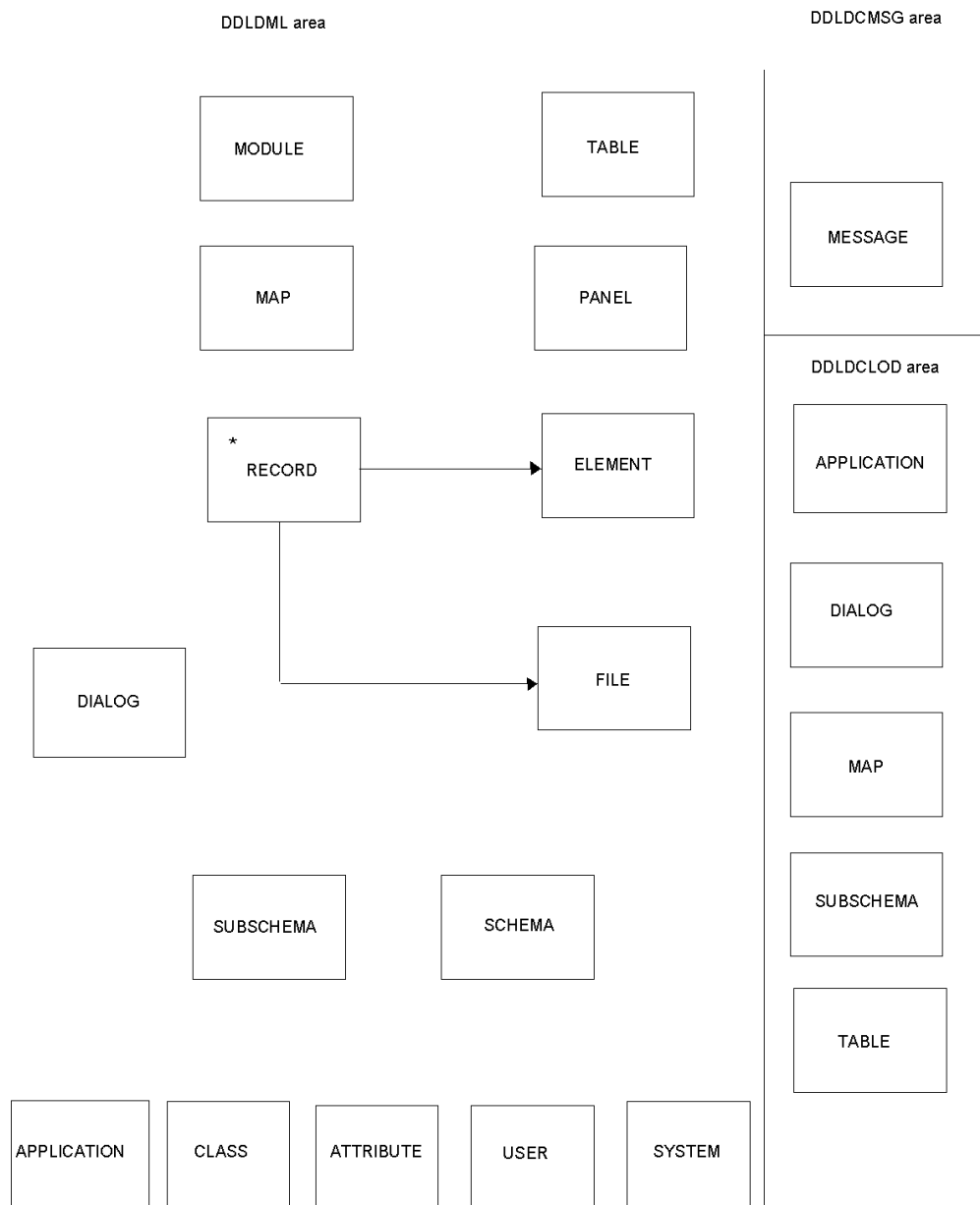
For each element connected to the record

- Extract the record element and all of its subordinate elements

For each IDD defined file connected to the record

- Extract the file

All attributes, systems and users associated with the record are also extracted.



Extract Path for ATTRIBUTE when LEVEL = DIALOG

This section describes the EXTRACT path for ATTRIBUTE when LEVEL = DIALOG.

EXTRACT, ATTRIBUTE = attribute-name, LEVEL = DIALOG

The following are extracted:

For each dialog connected to the attribute specified in the EXTRACT statement, and for each dialog connected to an attribute of the attribute specified in the extract statement:

1. Extract the dialog
 - Extract the corresponding dialog load module from DDLDCLOAD (load) area
2. For each module connected to the dialog and for each included process module within the process module
 - Extract the process module source
 - For each message code in the process module source
 - Extract the message from DDLDCMSG (message) area
 - For each referenced dialog within the process module source
 - Extract the dialog and its components (start at step 1. above)
3. For each record connected to the dialog
 - Extract the record
 - For each element connected to the dialog record
 - Extract the dialog record element and all of its subordinate elements

For each IDD defined file connected to the record

 - Extract the file
4. For each file used by the dialog
 - Extract the file

5. For the subschema connected to the dialog
 - Extract the schema connected to the subschema
 - For each record connected to the schema
 - Extract the schema record
 - For each element connected to the schema record
 - Extract the schema record element and all of its subordinate elements
 - For each subschema connected to the schema
 - Extract the subschema
 - Extract the corresponding subschema load module from DDLDCLOUD (load) area
 - Extract each Logical Record Facility work record used by the subschema
 - For each record connected to the subschema
 - Extract the Record
 - Extract the subschema record element and all of its subordinate elements
6. Extract the map connected to the dialog
 - Extract the corresponding load module map from DDLDCLOUD (load) area
 - For each message code in the map
 - Extract message from DDLDCMSG (message) area
 - For each edit and/or code table used in the map
 - Extract the table
 - Extract the corresponding table load module from DDDLDCLOUD (load) area
 - For each work record used in the map
 - Extract the map record
 - For each element connected to the map record
 - Extract the map record element and all of its subordinate elements

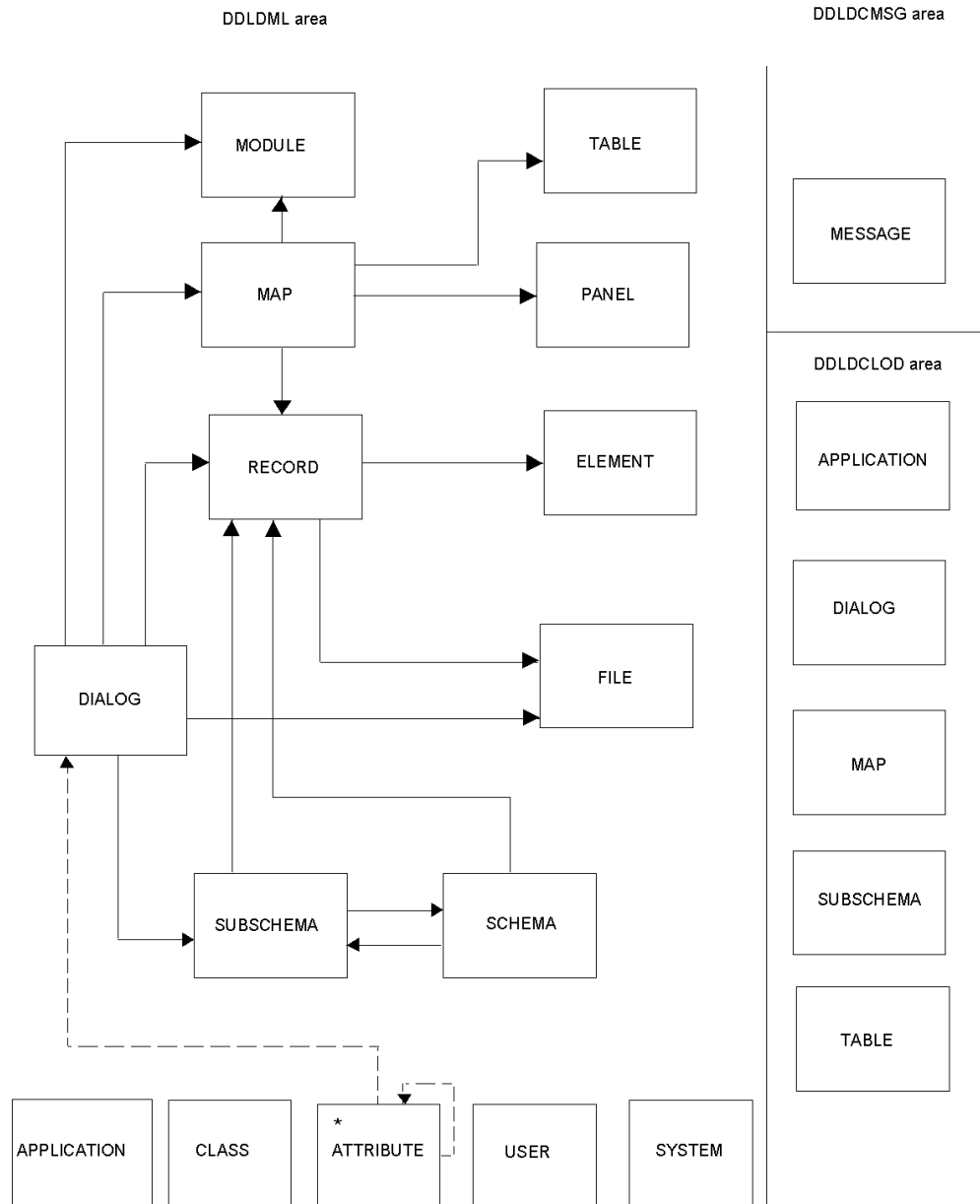
For each panel connected to the map

- Extract the panel

For the help module connected to the map

- Extract the module source
- Extract the map help load module

All attributes, systems and users associated with every entity are also extracted.



EXTRACT Path for ATTRIBUTE when LEVEL = ENTITY

This section describes the EXTRACT path for ATTRIBUTE when LEVEL = ENTITY.

EXTRACT, ATTRIBUTE=attribute-name, LEVEL=ENTITY

For the attribute specified in the EXTRACT statement:

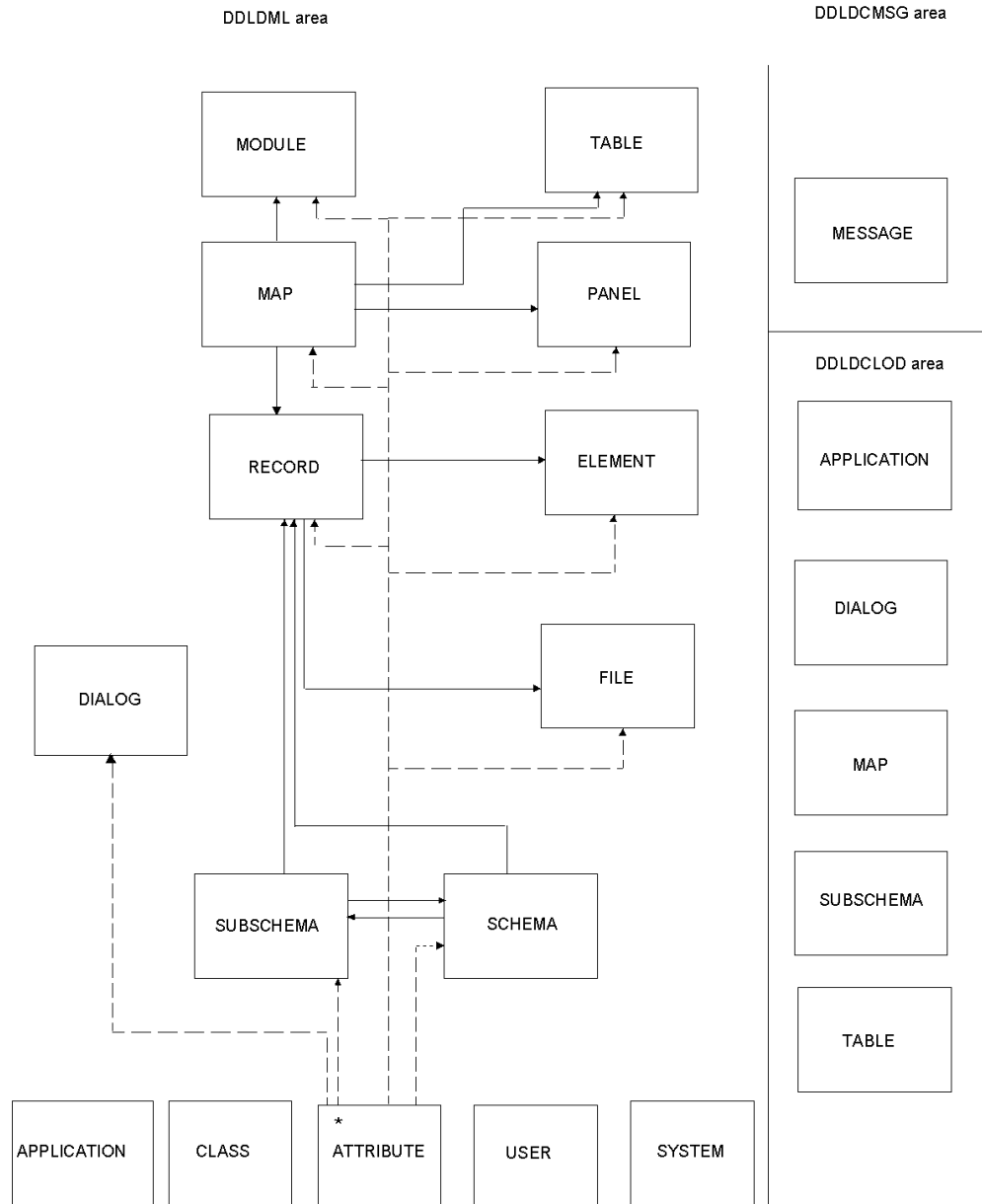
Find each entity connected to the named attribute, extract these entities and extract components of all these entities:

1. Extract all schemas connected to the attribute
 - For each record connected to the schema
 - Extract the schema record
 - For each element connected to the schema record
 - Extract the schema record element and all of its subordinate elements
2. Extract all subschemas connected to the attribute
 - Extract the subschema
 - Extract the corresponding subschema from DDLDCLD (load) area
 - For each record connected to the subschema
 - Extract the subschema record
 - For each element connected to the subschema record
 - Extract the subschema record element and all of its subordinate elements
3. Extract all records connected to the attribute
 - Extract the record
 - For each element connected to the record
 - Extract the record element and all of its subordinate elements
 - For each IDD defined file connected to the record
 - Extract the file
4. Extract all elements connected to the attribute
 - For each element
 - Extract all of its subordinate elements
5. Extract all modules connected to the attribute
 - Extract the module source
6. Extract all programs connected to the attribute
 - For each program connected to the attribute
 - Extract the programs PROG-051 record
 - Extract the programs load module from DDLDCLD (load) area

7. Extract all maps connected to the attribute
 - Extract the corresponding map from DDLDCLOUD (load) area
 - For each message code in the map
 - Extract message from DDLDCMSG (message) area
 - For each edit and/or code table used in the map
 - Extract the edit and/or code table used in the map
 - Extract the edit and/or code table from DDLDCLOUD (load) area
 - For each record connected to the map
 - Extract the map record
 - For each element connected to the map record
 - Extract the map record element and all of its subordinate elements
 - For the panel connected to the map
 - Extract the panel
 - For each help module connected to the map
 - Extract the module source
 - Extract the map help load module

8. Extract all the panels connected to the attribute
 9. Extract the class connected to the attribute
- Extract the attribute

All attributes, systems and users associated with every entity are also extracted.



Extract Path for ATTRIBUTE when LEVEL = ONLY

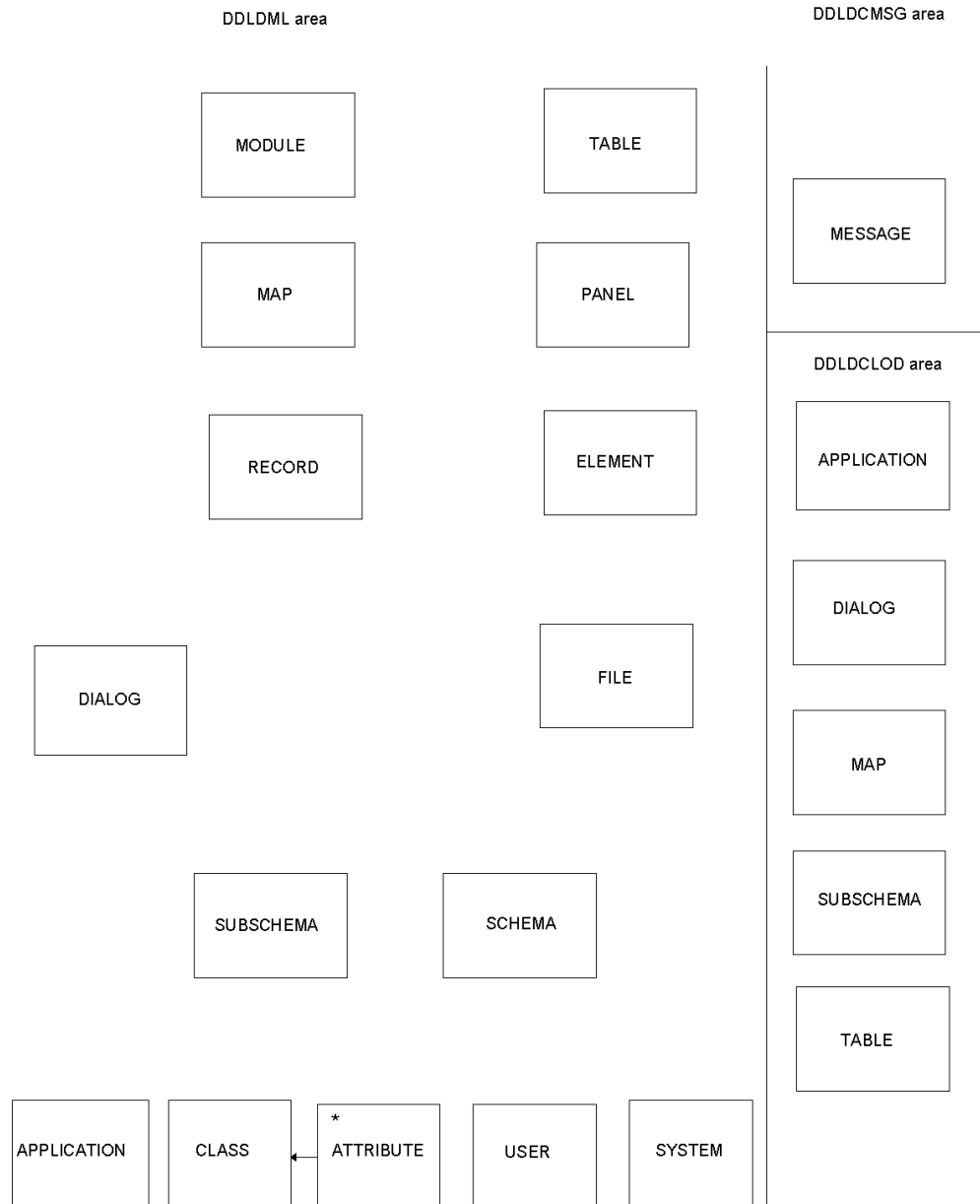
This section describes the EXTRACT path for ATTRIBUTE, when LEVEL = ONLY.

EXTRACT, ATTRIBUTE = attribute-name, LEVEL = ONLY

For the attribute specified in the EXTRACT Statement

1. Extract the class connected to the attribute
Extract the attribute

All users associated with the attribute are also extracted.



Extract Path for SUBSCHEMA when LEVEL = DIALOG

This section describes the EXTRACT path for SUBSCHEMA when LEVEL = DIALOG.

EXTRACT, SUBSCHEMA = subschema-name, LEVEL = DIALOG

For each dialog that uses the specified subschema:

1. Extract the dialog

Extract the corresponding dialog load module from DDLDCLOUD (load) area

2. For each process module connected to the dialog and for each included process module within the process module

- Extract the process module source
- For each message code in the process module source
 - Extract the message from DDLDCMSG (message) area
- For each referenced dialog within the process module source
 - Extract the dialog and its components (start at step 1. above)

3. For each record connected to the dialog

Extract the record

For each element connected to the dialog record

- Extract the dialog record element and all of its subordinate elements

For each IDD defined file connected to the record

- Extract the file

4. For each file used by the dialog

- Extract the file

5. For the subschema connected to the dialog

- Extract the schema connected to the subschema
- For each record connected to the schema
 - Extract the schema record
 - For each element connected to the schema record
 - Extract the schema record element and all of its subordinate elements
- For each subschema connected to the schema
 - Extract the subschema
 - Extract the corresponding subschema load module from DDLDCLOUD (load) area
- Extract each Logical Record Facility work record used by the subschema
- For each record connected to the subschema
 - Extract the Record
 - Extract the subschema record element and all of its subordinate elements

6. Extract the map connected to the dialog

Extract the corresponding load module map from DDLDCLOUD (load) area

For each message code in the map

- Extract message from DDLDCMSG (message) area

For each edit and/or code table used in the map

- Extract the table
- Extract the corresponding table load module from DDLDCLOUD (load) area

For each work record used in the map

- Extract the map record
- For each element connected to the map record
 - Extract the map record element and all of its subordinate elements

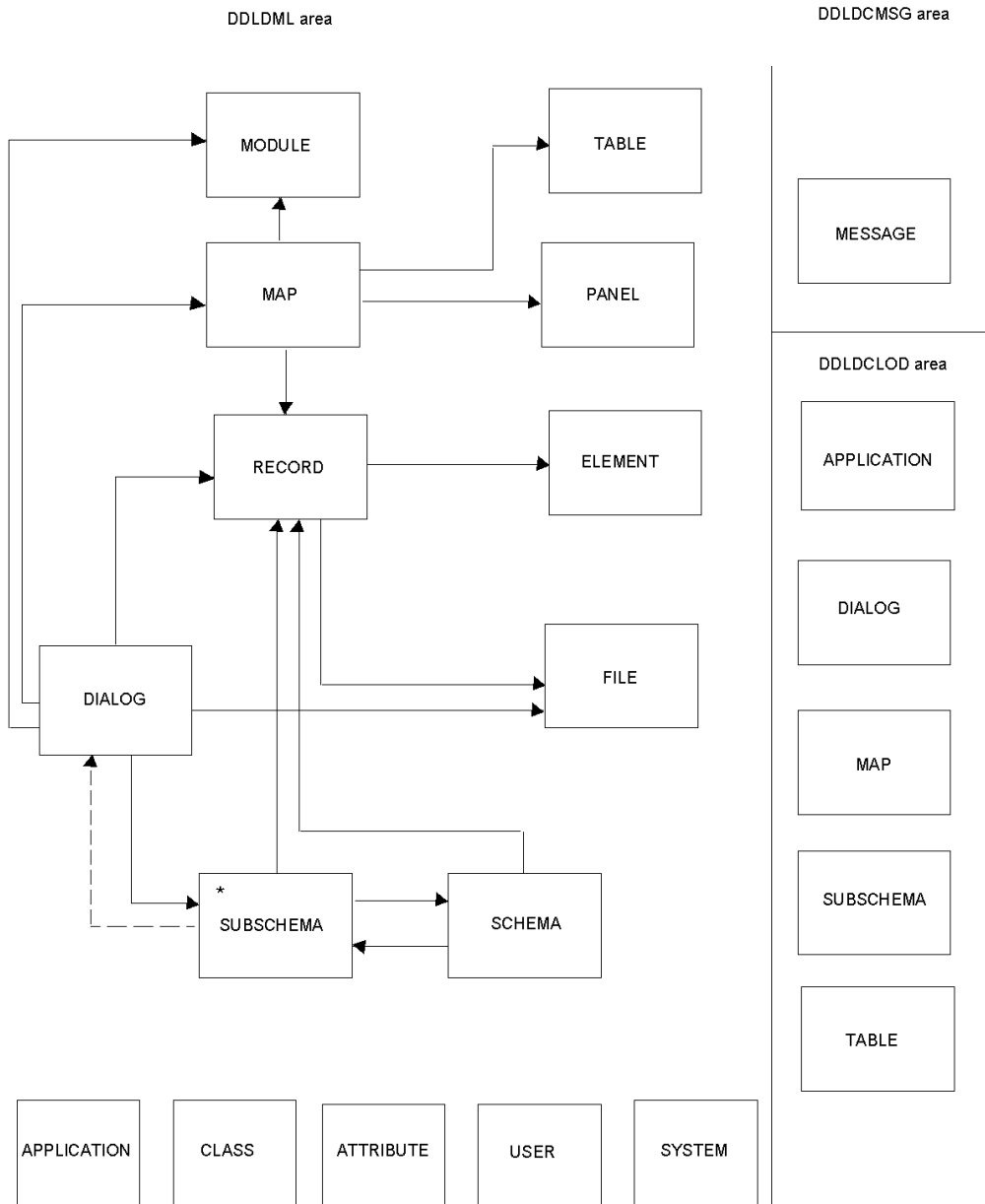
For each panel connected to the map

- Extract the panel

For the help module connected to the map

- Extract the module source
- Extract the map help load module

All attributes, systems, and users associated with every entity are also extracted.



Extract Path for SUBSCHEMA when LEVEL = ENTITY

This section describes the EXTRACT path for SUBSCHEMA when LEVEL = ENTITY.

EXTRACT, SUBSCHEMA=subschema-name, LEVEL=ENTITY

For the subschema specified in the EXTRACT statement:

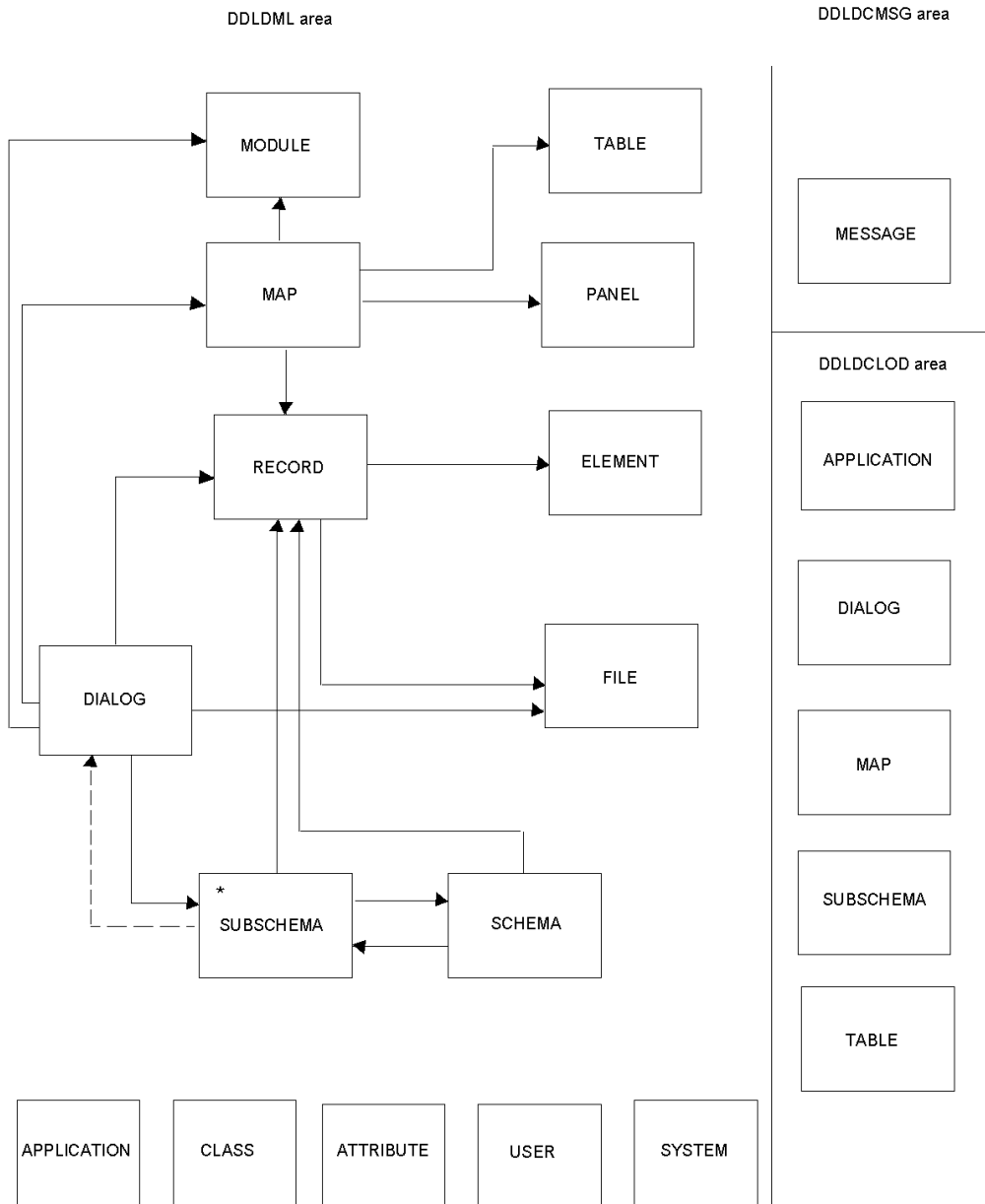
1. Extract the subschema

Extract the corresponding subschema load module from the DDLDCLOAD (load) area

For each record connected to the subschema

- Extract the subschema record
- For each element connected to the subschema record
 - Extract the subschema record element and all of its subordinate elements
- Extract all files associated with record

All attributes, systems, and users associated with every entity are also extracted.

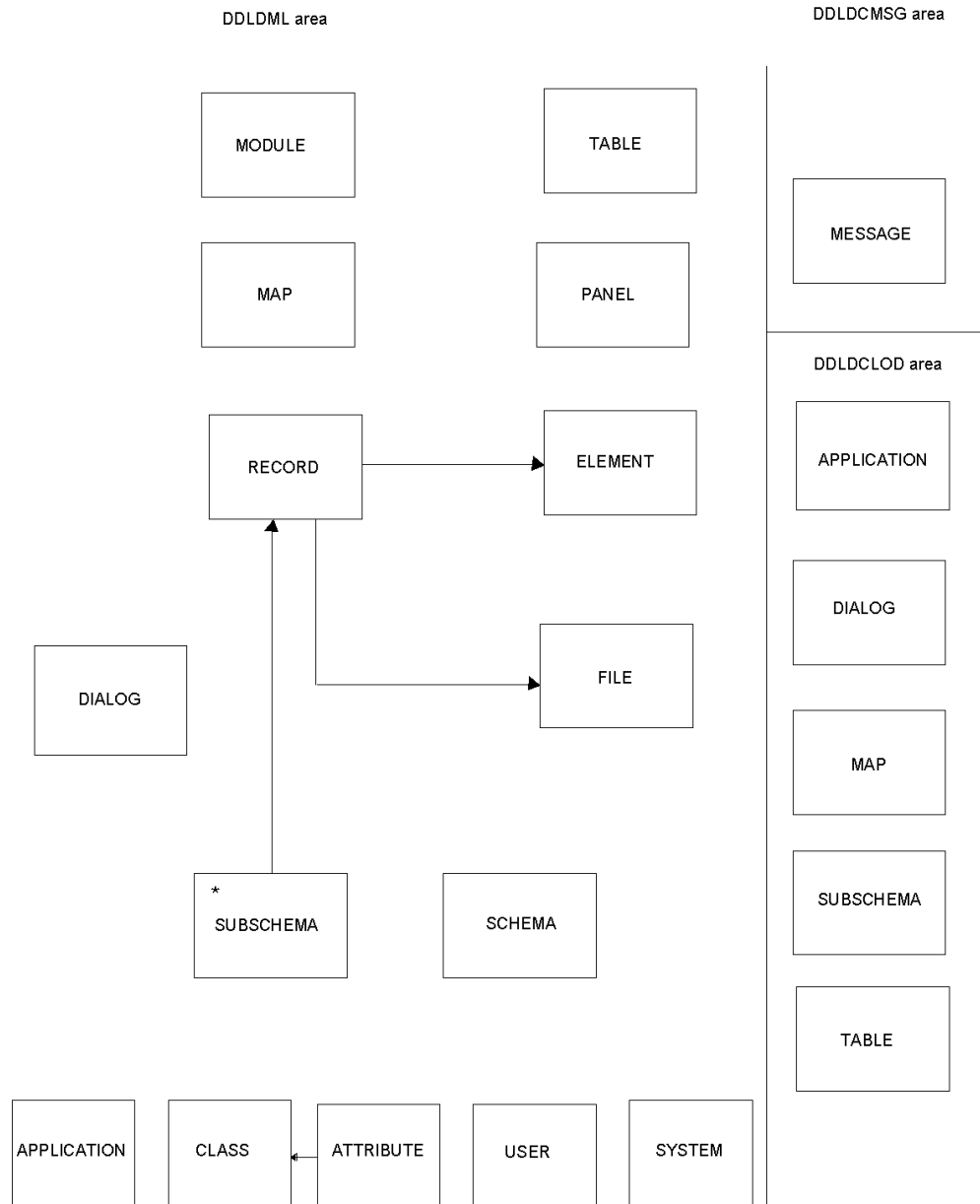


Extract Path for APPLICATION when LEVEL = DIALOG

1. For each record defined as a global record of the application
 - Extract the record
 - For each element connected to the global record
 - Extract the global record element and all of its subordinate elements
 - For each IDD defined file connected to the record
 - Extract the file
2. For each dialog named in a function of the application
 - Extract the dialog
 - Extract the corresponding dialog load module from DDLDCLOD (load) area
3. For each process module connected to the dialog and for each included process module within the process module
 - Extract the process module source
 - For each message code in the process module source
 - Extract the message from DDLDCMSG (message) area
 - For each referenced dialog within the process module source
 - Extract the dialog and its components (start at step 2. above)
4. For each record connected to the dialog
 - Extract the record
 - For each element connected to the dialog record
 - Extract the dialog record element and all of its subordinate elements
 - For each IDD defined file connected to the record
 - Extract the file
5. For each file used by the dialog
 - Extract the file
6. For the subschema connected to the dialog
 - Extract the schema connected to the subschema
 - For each record connected to the schema
 - Extract the schema record
 - For each element connected to the schema record
 - Extract the schema record element and all of its subordinate elements
 - For each subschema connected to the schema
 - Extract the subschema

- Extract the corresponding subschema load module from DDLDCLOAD (load) area
 - Extract each Logical Record Facility work record used by the subschema
 - For each record connected to the subschema
 - Extract the Record
 - Extract the subschema record element and all of its subordinate elements
- 7. Extract the map connected to the dialog
 - Extract the corresponding map from DDLDCLOAD (load) area
 - For each message code in the map
 - Extract message from DDLDCMSG (message) area
 - For each edit and/or code table used in the map
 - Extract the table
 - Extract the corresponding table load module from DDLDCLOAD (load) area
 - For each work record used in the map
 - Extract the map record
 - For each element connected to the map record
 - Extract the map record element and all of its subordinate elements
 - For the panel connected to the map
 - Extract the panel
 - For each help module connected to the map
 - Extract the module source
 - Extract the map help load module

All attributes, systems, and users associated with every entity are also extracted.



Appendix C: Customization

This appendix contains the following topics:

This section contains the following topics:

[Introduction](#) (see page 327)

[Installation Parameters](#) (see page 327)

[Message Severity](#) (see page 327)

[Entity Exclusion Table](#) (see page 330)

[Manipulation of the Extract File](#) (see page 332)

Introduction

CA IDMS Dictionary Migrator can be customized in the following general areas:

Area	Customization
Installation Parameters	A set of options that are made to non-standard processing to meet special situations described in the CA IDMS installation guides.
Message Severity	The severity of some messages can be changed to correspond to an installation's standards. Messages can be omitted from reports or to stop processing.
Entity Exclusion Table	The exclusion table can define any entity occurrences that you do not want migrated.
Manipulation of the Extract File	The format of the extract file is described to help you when you want to use the extract file for any further processing.

Installation Parameters

Note: For more information about these parameters, see the CA IDMS installation guides.

Message Severity

This section contains information about message severity.

Customization Instructions

Dictionary Migrator provides a facility where the severity level of messages can be changed.

Note: You do not need to use this facility for the proper execution of CA IDMS Dictionary Migrator. The facility is provided for users whose shop standards mandate some deviation from the basic migration strategy. This option is not used in default execution mode of CA IDMS Dictionary Migrator. Therefore, no load module is provided as part of the standard installation process.

General

This facility allows the CA IDMS Dictionary Migrator user to generate a table of specific messages along with their required severity levels.

Message Severity Level Table Generation Steps

The following are Message Severity Level Table Generation Steps:

1. Customize the source member USMSGSVT from the SRCLIB library.
2. Assemble and link edit this member into your CA IDMS Dictionary Migrator Loadlib.

Message Severity Level Table Customization

Edit the source member USMSGSVT as follows:

First Line:

```
#EXCLTAB TBLNAME=USMSGSVT
```

For each Message that has a severity level that needs to be changed, add a line that invokes the #MSGSEV macro with the required parameters.

```
#MSGSEV MSG=messageid,SEV=severitylevel
```

where:

messageid - is a 7 Character Message ID whose first four characters correspond to one of the following:

- EXTR
- ES00
- ES11
- VALD

The messages with the first four characters that do not correspond to one of the above may not have their severity levels altered.

severitylevel - is a single character either:

- X
- W
- E
- C

These characters may be interpreted as:

- **X** - NO PRINT - Occurrences of this message do not generate a printed message on any report.
- **W** - WARNING - Warning Message printed.
- **E** - ERROR - Error Message printed.
- **C** - CRITICAL - Critical Error printed. This error type causes CA IDMS Dictionary Migrator to terminate processing at the end of the current processing stage.

Last Lines:

```
#EXCLEND  
END
```

Sample Table:

```
#EXCLTAB TBLNAME=USMSGSVT  
#MSGSEV MSG=ES00515,SEV=X  
#MSGSEV MSG=VALD219,SEV=C  
#EXCLEND
```

Assembly and Linkage

Any level of IBM assembler and linkage editor can be used to create the Message Severity Level table.

1. MACLIB of the assembler step: Add the SRCLIB from installation to the files named in MACLIB.
2. SYSLMOD of the link edit step: Use the same LOADLIB from which CA IDMS Dictionary Migrator is executed.

Note: No default Message Severity Level table is provided. If you wish to generate this table see the SRCLIB member USMSGVT.

Entity Exclusion Table

The following instructions explain how to make use of the entity exclusion facility and detail what entity types may be excluded.

Customization Instructions

CA IDMS Dictionary Migrator provides a facility where specific entity occurrences can be excluded from the migration process.

Note: You do not need to use this facility for the proper execution of CA IDMS Dictionary Migrator. The facility is provided for users whose shop standards mandate some deviation from the basic migration strategy. A default version of this module with no entities excluded is provided in load module form as part of the standard installation process.

General

This facility allows the CA IDMS Dictionary Migrator user to generate a table of specific entities that are to be excluded from the migration process.

Entity Exclusion Table Generation Steps

1. Customize the source member USMEXCLD from the SRCLIB library.
2. Assemble and linkedit this member into your CA IDMS Dictionary Migrator Loadlib.
3. At Execution time, define the Entity Exclusion Table to be used to CA IDMS Dictionary Migrator through the EXCLTAB parameter. This step is only necessary if a table name other than the default USMEXCLT is to be used.

Entity Exclusion Table Customization:

Edit the source member USMEXCLD as follows :

FirstLine:

```
#EXCLTAB TBLNAME=yourname.
```

Use any name up to 8 characters following standard naming rules.

Default name=USMEXCLT.

For each entity occurrence to be excluded, add a source line that invokes the #EXCLEN macro with the required parameters:

```
#EXCLEN TYPE=entitytype,  
NAME=entityname,  
VERS=version, (if applicable)  
QUAL=qualifier (if applicable)
```

where :

entitytype (Required)—One of the following entity types that must be specified in uppercase and can be abbreviated to the first two characters.

AT (TRIBUTE)
 CL (ASS)
 DI (ALOG)
 EL (EMENT)
 FI (LE)
 MA (P)
 MO (DULE)
 PR (OGRAM)
 RE (CORD)
 SC (HEMA)
 SU (BSCHEMA)
 SY (STEM)
 US (ER)

entityname (Required)—Entity OccurrenceName in uppercase.

version (Optional)—Defaults to version number 1. Version number or ALL (for all versions). Version numbers apply to all entity types except from Attribute, Class, and User.

qualifier—Applies to the following entity types (must be in uppercase):

- Attribute -- qualifier is class name (required).
- Module -- qualifier is language name (optional). Language name should be specified if the module is stored in the Data Dictionary with a LANG-067 field other than spaces. See Note below.
- Subschema -- qualifier is schema name (required).

Note: ADS process modules are stored in the Data Dictionary with a language name = PROCESS. If you want to exclude an ADS module from the migration, you should specify QUAL=PROCESS on the #EXCLENT line entry for the module to be excluded.

Last Lines:

```
#EXCLEND
END
```

Sample Table:

```
#EXCLTAB TBLNAME=EXCLTAB
#EXCLENT TYPE=RECORD, NAME=TESTREC, VERS=100
#EXCLENT TYPE=ATTRIBUTE, NAME=WORK, QUAL=STATUS
#EXCLEND
END
```

Assembly and Linkage

Any level of IBM assembler and linkage editor can be used to create the Entity Exclusion Table. As indicated above, the default Entity Exclusion Table name is USMEXCLT. If you want to utilize a table with a different name, remember to indicate the named table on the EXCLTAB CA IDMS Dictionary Migrator parameter. For example, EXCLTAB=mytable.

1. MACLIB of the assembler step: Add the product installation MACLIB to the files named in MACLIB.
2. SYSLMOD of the link edit step: Use the same LOADLIB from which CA IDMS Dictionary Migrator is executed.

Default Entity Exclusion Table (table name USMEXCLT)

```
#EXCLTAB  
#EXCLEND
```

```
END
```

Manipulation of the Extract File

The extract file is a set of extract records that can be manipulated.

Extract Record

The following is a COBOL definition of the extract record. A description of the various fields follows the definition.

```

01  MIGRATOR-EXTRACT
    05  EXTR-CO-LOC.
        10  EXTR-CHANGEONLY          PIC  X(01).
        10  EXTR-EXTRACT-LEVEL       PIC  X(01).
    05  EXTR-VSAMEXT-KEY.
        10  EXTR-ENTITY-TYPE         PIC  S(4) USAGE IS COMP.
        10  EXTR-ENTITY-NAME         PIC  X(40).
        10  EXTR-SELECT-VERSION      PIC  S9(4) USAGE IS COMP.
        10  EXTR-ALT-ENTITY-NAME     PIC  X(40).
    05  EXTR-ENTITY-LEVEL            PIC  X(2).
    05  EXTR-ENTITY-UTILITY          PIC  X(2).
    05  EXTR-SELECT-DICTIONARY       PIC  X(8).
    05  EXTR-SELECT-NODE             PIC  X(8).
    05  EXTR-SOURCE-USER             PIC  X(32).
    05  EXTR-SOURCE-PASSWORD         PIC  X(8).
    05  EXTR-COBOL-FORMAT            PIC  X(01).
    05  EXTR-MOVESTAMP              PIC  X(01).
    05  EXTR-SELECT-COMMENT-SOURCE   PIC  X(01).
    05  EXTR-SELECT-DEFINITION-SOURCE PIC  X(01).
    05  EXTR-GENERATE                PIC  X(01).
    05  EXTR-RESERVED-1              PIC  X(14).
    05  EXTR-LOAD-TYPE              PIC  S9(4) USAGE IS COMP.
    05  EXTR-RESP-OVERRIDE           PIC  X(01).
    05  EXTR-COMPARE-DATE.
        10  EXTR-COMPARE-MN          PIC  X(02).
        10  FILLER                   PIC  X(01).
        10  EXTR-COMPARE-DD          PIC  X(02).
        10  FILLER                   PIC  X(01).
        10  EXTR-COMPARE-YY          PIC  X(02).
    05  EXTR-COMPARE-TIME.
        10  EXTR-COMPARE-TIME-6.
            15  EXTR-COMPARE-HH      PIC  X(02).
            15  EXTR-COMPARE-MN      PIC  X(02).
            15  EXTR-COMPARE-SS      PIC  X(02).
        10  EXTR-COMPARE-HN          PIC  X(02).
05  EXTR-RESERVED-2                  PIC  X(44).

```

Field Definitions and Values

- **EXTR-CHANGEONLY**—CHANGEONLY Status of extract record. See ADDONLY, CHANGEONLY process parameters and USMTPARM DELADDS option.
- 'D'—Entity is dropped
- 'Y'—Entity is selected for modify to object dictionary
- 'N'—Entity is selected for ADD to object dictionary
- **EXTR-EXTRACT-LEVEL** — Extract Level of migration. See LEVEL process and extract parameters.
 - 'S'—"Dialog" Level
 - 'E'—"Entity" Level
 - 'O'—"Only" Level
- **EXTR-VSAMEXT-KEY**—An 84 byte key used to create the VSAM work file that contains extracted entities for dictionary comparisons.
- **EXTR-ENTITY-TYPE**—2 byte numeric field. Encoded type of entity. See entity type definitions table.
- **EXTR-ENTITY-NAME**—40 byte field name. It contains the entity occurrence name.
- **EXTR-SELECT-VERSION**—2 byte numeric version number of extracted entity. If the entity type does not have version number, set to 0. If null, new version processing is used it contains -2.
- **EXTR-ALT-ENTITY-NAME**—40 byte field used to qualify certain entity types. Contains the following:
 - Class name for Attributes
 - Schema name for Subschema
 - Language name (if any) for Modules
 - Element details for Element definitions
- **EXTR-ENTITY-LEVEL**—2 byte character field. Relative level number for entities that have nest relationships. Used to ensure that correct ordering of entity levels, for example elements, are maintained.
- **EXTR-ENTITY-UTILITY**—2 byte character field. Indicates which CAIDMS utility is used to process the extracted entity. See utility type definitions table below.
- **EXTR-SELECT-DICTIONARY**—8 byte character field specifying the source dictionary name. See DICTIONARY extract or process parameter.
- **EXTR-SELECT-NODE**—8 byte character field specifying the source database node name. See NODE extract or process parameter.
- **EXTR-SELECT-USER**—32 byte character source dictionary user name. See IDSOURCE, EXTRACT, or PROCESS parameter.

- **EXTR-SELECT-PASSWORD**—8 byte character source dictionary password. See IDSOURCE extract or process parameter.
- **EXTR-COBOL-FORMAT**—1 byte character field. See COBOLFORMAT process parameter. Valid only for Record type.
 - 'Y'—Generate COBOL like syntax
 - 'N'—Do not generate COBOL like syntax
- **EXTR-MOVESTAMP**—1 byte character field. See MOVESTAMP process parameter. Valid only for Map type.
 - 'Y'—Decompile map with DATETIME=YES
 - 'N'—Do not decompile with DATETIME=YES
- **EXTR-SELECT-COMMENT-SOURCE**—1 byte character field. See: EXSOURCE=COMMENT PROCESS, EXTRACT parameter
- **EXTR-SELECT-DEFINITION-SOURCE**—Determines whether or not definitions are extracted.
 - "Y"—Definitions are extracted
 - "N"—Definitions are not extracted
- **EXTR-GENERATE**—Determines whether or not the entity needs to be generated.
 - "Y"—Entity needs to be generated
 - "N"—Entity does not need to be generated
- **EXTR-RESERVED-1**—Reserved for future use.
- **EXTR-LOAD-TYPE**—2 byte numeric field. The source type of entity that the load module represents. See entity type definitions table.
- **EXTR-RESP-OVERRIDE**—determines whether or not a registration override statement should be used.
 - "Y"—Override should be used
 - "N"—Override should not be used

Values for Utility Types

Item	Value
Special Record	value '01'
IDMSQDDL	value '06'
IdmsdddI	value '10'
Tablddl	value '12' IDMSDDDL for tables only
Loadddl	value '15' IDMSDDDL for load modules only

Item	Value
reserved	value '20'
Idmschem	value '25'
Objdchem	value '26' IDMSCHEM from object dictionary
reserved	value '28'
Idmsubsc	value '30'
Objdubc	value '31' IDMSUBSC from object dictionary
Rhdcmput	value '40'
Objdmput	value '41' RHDCMPUT from object dictionary
Progddd	value '50' IDMSDDDL for programs only; also creates ADSOBCOM statements for dialogs
Objpddd	value '51' ADSOBCOM statement from object dictionary.
reserved	value '90'
reserved	value '95'

Value for Entity Type Codes

Item	Value
Special weight (used for CA IDMS Dictionary Migrator control)	value +01
Message delete	value +03
Load delete	value +05
Dialog delete	value +10
Program delete	value +11
Subschema delete	value +20
Reserved	value +23
Schema delete	value +25
Map delete	value +28
Table delete	value +29
Process delete	value +30
Qfile delete	value +31
Reserved	value +33

Item	Value
Module delete	value+35
File delete	value+36
Panel delete	value+37
Record delete	value+39
Report delete	value+41
Transaction delete	value+43
Element delete	value+45
System delete	value+46
Attribute delete	value+47
User delete	value+48
Class delete	value+49
Class	value+50
Module	value+51
User (pre)	value+52
Attribute	value+54
System	value+56
Application	value+57
Element	value+60
Reserved	value+62
Qfile	value+63
Process	value+64
Table	value+66
File	value+68
Record	value+70
Report	value+71
Transaction	value+72
Panel	value+73
Schema	value+75
Reserved	value+85
User (post)	value+88

Item	Value
Map	value+78
Subschema	value+79
Program	value+80
Dialog	value+82
Reserved	value+86
Reserved	value+87
Load Module	value+90
SQL SCHEMA	value+91
Reserved	value+92
SQL TABLE	value+93
SQL VIEW	value+94
Message	value+95
Reserved	value+96
CONSTRAINT	value+97
TABLE PROCEDURE	value+98
SQL KEY	value+99

Glossary

Change only

The method of CA IDMS Dictionary Migrator execution which limits selection to changed entities only.

Entry point

The entity that serves as a gateway (for CA IDMS Dictionary Migrator) to the entities that are to be extracted from the source dictionary. You specify the entry point in the EXTRACT statement. For example, in the statement

```
EXTRACT, SUBSCHEMA=subschema-name, LEVEL=DIALOG
```

the subschema that has been specified is the entry point.

Extract path

The route CA IDMS Dictionary Migrator follows through the dictionary's set connections from the entity specified in the EXTRACT statement to all of its related components. What you include in the EXTRACT statement determines the beginning and direction of this route. For example, if you enter this statement:

```
EXTRACT, SUBSCHEMA=subschema-name, LEVEL=ENTITY
```

CA IDMS Dictionary Migrator will follow the dictionary set connections from the subschema that you specify to all related records and all of the elements related to the records.

Migration

The movement of entities from one or more source dictionaries to an object dictionary. Migration is comprised of extracting entities from one or more source dictionaries, checking the effect of source dictionary entities upon the object dictionary, creating the syntax necessary for input to the CA IDMS compilers, and uploading the entities into the object dictionary.

Object dictionary

A dictionary which acts as the destination for entities that are moved during the migration process. Your object dictionary is populated by entities that are copied from your source dictionary(s). Only one object dictionary can be used during each migration.

Source dictionary

The dictionary which acts as a reservoir for entities that are available for migration into another (object) dictionary. Your source dictionary is not changed during the migration process. Multiple source dictionaries can be used during each migration. And, if you desire, you can use your object dictionary as a source dictionary.

Starting point

This entity and all of its related entities are copied into your object dictionary. You determine what the starting point is by specifying the entry point and whatever option you choose to include in the LEVEL parameter. For example, in the statement

EXTRACT, SUBSCHEMA=subschema-name, LEVEL=DIALOG

the dialog (or dialogs) related to the subschema that you specified is the starting point.

Syntax files

A collection of files that contain source statements used as input to the CA IDMS utilities and compilers. The syntax files permit completion of the movement of entities and entity components from one or more source dictionaries to the object dictionary.

Trace

To identify an entity, its related entities, its related components, and all of the connections among these entities.

Index

1

- 1. Describe the Migration Environment • 132

2

- 2. Specify a New Starting Point • 132

3

- 3. Accept or Override Defaults • 133

4

- 4. Specify Starting Point Overrides • 134

5

- 5. Include/Exclude Source Types (Current Starting Point) • 135

6

- 6. Include/Exclude Load Types (Current Starting Point) • 135

7

- 7. Obtain CA IDMS Dictionary Migrator Reports • 136

8

- 8. Store Setup into a PARMFILE • 137

9

- 9. Continue to Use DMA • 138

A

Additional Tasks Performed by CA IDMS Dictionary Migrator • 102

C

- CA IDMS Dictionary Migrator is easy to use • 18
- CA IDMS Dictionary Migrator Parameter Statements • 42
- CA IDMS Dictionary Migrator Reports • 153
- CHANGE Statement • 84
- Choose Parameters for Migration • 32
- Common Fields on DMA Screens • 128

- Comparison Reports • 95
- Complete First Steps Using DMA • 131
- Comprehensive Reports • 31
- Configure the Migration to Meet Your Needs • 24
- Copying a DMA Parameter File • 147
- Copying CA IDMS/Dictionary Migrator JCL to DMA • 148

D

- Defining a New Migration • 129, 131
 - Overview • 131
 - Sequence of Screens • 129
- Definition of Migration • 16
- Display PF Key Values • 122
- DMA Activity Paths • 129
- DMA Commands • 119
- DMA Requirements • 116

E

- EXTRACT Statement • 67
- Extract Summary and Extract Detail Reports • 91

F

- First Steps in Using DMA • 125
- Flexible Processing Capabilities • 19

H

How CA IDMS Dictionary Migrator Operates • 102

I

Improve Development Productivity • 20

L

Limiting Entities for Overall Migration • 141

M

- Menu-Driven Parameter Selection • 34
- Migrate Systems Quickly and Accurately • 17
- Migration Steps • 16
- Modifying a DMA Parameter File • 144

O

Obtain Reports • 37

Online Aid for Selecting Parameters • 32
Online Help • 35

P

Parameter Storage • 35

Q

Questions and Answers About Using DMA • 123

R

Resetting and Exiting DMA • 125
Running as a Single Job Execution Under Central
Version • 108
Running as a Two-Job Execution • 110

S

Selecting a DMA Parameter File • 143
Selecting Parameters and Limiting Migration • 40
Specify Migration Overrides • 140
Specifying a DMA Parameter File • 143
Submitting Parameters and JCL from the Batch
Environment • 148
Submitting Parameters and JCL Online • 147

T

Take the Guesswork Out of Migration • 17
Types of Messages • 117

U

Use DMA to Set up a Migration Easily • 33
User Authority • 117
Using a Series of Commands from the Menu • 140
Using Global Overrides • 138

V

View JCL for Migration • 37
View Syntax for Migration • 37

W

When to Use CA IDMS Dictionary Migrator • 16