CA IDMS - 19.0
Using the CA IDMS Dictionary Migrator
This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

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

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

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

The manufacturer of this Documentation is CA.

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

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

Introduction ........................................................................................................... 11

Migration Steps .................................................................................................... 11

Definition of Migration .......................................................................................... 12

When to Use CA IDMS Dictionary Migrator .......................................................... 12

Migrate Systems Quickly and Accurately ............................................................... 12

Take the Guesswork Out of Migration .................................................................... 12

CA IDMS Dictionary Migrator General Information ................................................. 13
CA IDMS Dictionary Migrator Is Easy to Use .......................................................... 14
Flexible Processing Capabilities ............................................................................... 14
  SQL Entity Migration ........................................................................................... 15
Improve Development Productivity ........................................................................... 16
  New Applications Environment ............................................................................ 16
  Existing Applications Environment ....................................................................... 17
Backup and Recovery ............................................................................................... 19
Configure the Migration to Meet Your Needs .......................................................... 19
  Migrate Entire Systems ....................................................................................... 19
    Migrate by Organizational Structure ................................................................ 19
    Migrate by Organizational Structure Without Regard to Dialog ....................... 20
    Migrate by Component Relationship .................................................................. 21
  Migrate a Single Dictionary Entity ........................................................................ 22
  Migrate in a Multiple-Machine Environment ....................................................... 22
    Single Job Execution--Using Node Communication ............................................ 23
    Two-Job Execution .............................................................................................. 23
Advantages of the Two-Job Execution ................................................................. 24
Extract Entities from Multiple Source Dictionaries ........................................... 24
CA IDMS Dictionary Migrator Information .......................................................... 24
Comprehensive Reports ....................................................................................... 25
Online Aid for Selecting Parameters ................................................................. 26
DMA Your CA IDMS Dictionary Migrator Connection ....................................... 26
Choose Parameters for Migration ...................................................................... 27
DMA Functions .................................................................................................... 27
User Responsibilities ......................................................................................... 27
CA IDMS Dictionary Migrator Functions ............................................................ 27
Use DMA to Set up a Migration Easily ............................................................... 27
Alternative to Learning CA IDMS Dictionary Migrator Parameters ................. 28
Menu-Driven Parameter Selection ...................................................................... 28
Parameter Storage ............................................................................................. 28
Online Help ......................................................................................................... 29
When to Review the Output ................................................................................ 29
Obtain Reports ..................................................................................................... 30
View Syntax for Migration .................................................................................. 31
View JCL for Migration ....................................................................................... 31

CA IDMS Dictionary Migrator Operations .......................................................... 31
Operational Considerations ................................................................................ 32
Local Mode Processing ....................................................................................... 33
Additional Tasks Performed by CA IDMS Dictionary Migrator ....................... 33
How CA IDMS Dictionary Migrator Operates ................................................... 33
Two Methods of Operation ................................................................................. 34
Allocating Space for Work Files and Syntax Files ............................................. 35
Changing Block Size of Work Files and Syntax Files in z/OS ......................... 36
z/VSE File Assignments .................................................................................... 36
z/VSE File Allocation Alternate Method ............................................................. 36
Running as a Single Job Execution Under Central Version .............................. 38
Migration Activities - Single Job Execution ....................................................... 38
Step 1 -- USMVSAM ........................................................................................ 39
Step 2--USMSYNTX -- z/OS Only .................................................................... 39
Step 3 -- USMXTRCT ....................................................................................... 39
Step 4 -- USMLOAD1 ...................................................................................... 39
Step 5 -- USMLOAD3 (Optional) .................................................................... 40
Step 6 -- USMSQLOD (Optional) .................................................................... 40
Using CREATESYNTAX Run Type ................................................................... 40
Running as a Two-Job Execution ..................................................................... 40
Migration Activities - Two-Job Execution ......................................................... 41
Dictionary Migrator Assistant ............................................................... 43

Introduction ....................................................................................... 46

DMA Requirements ........................................................................... 46

Installation ......................................................................................... 46

System Requirements ......................................................................... 46

Terminal Type .................................................................................... 46

Security ............................................................................................. 46

Storage Requirements ......................................................................... 47

Program Storage ................................................................................ 47

Working Storage ............................................................................... 47

Scratch Storage ............................................................................... 47

Database Storage ............................................................................. 47

User Authority .................................................................................. 47

Types of Messages ............................................................................ 48

Screen Messages .............................................................................. 48

Confirm Screens .............................................................................. 48

DMA Commands ............................................................................... 49

Enter a Command ............................................................................ 49

System Commands Menu .................................................................. 50

Multiple Commands ......................................................................... 51

Display PF Key Values ...................................................................... 51

Access the Values Screen ................................................................ 51

Change a PF Key Value .................................................................... 51

Which PF Key Values Apply? .............................................................. 51

Questions and Answers About Using DMA ...................................... 52

What are the PF Key Values? .............................................................. 52

What are the System Commands? ..................................................... 52

What Should I Enter On a DMA Screen? .......................................... 53

Can I Perform DMA Activities? ......................................................... 53

How Will A Migration Affect The Target Dictionary? ....................... 53
How Do I Use the Dictionary? ................................................................. 53
First Steps in Using DMA ....................................................................... 53
   Access DMA ..................................................................................... 53
   Select DMA Activity ........................................................................ 54
Reset and Exit DMA ............................................................................. 54
   Reset DMA ....................................................................................... 54
   Exit DMA ........................................................................................ 55
Common Fields on DMA Screens .......................................................... 56
DMA Activity Paths ................................................................................ 56
Define a New Migration Sequence of Screens ....................................... 57
   1. Specify Overrides of DMA-Supplied Defaults ............................... 59
   2. Accept DMA-Supplied Defaults .................................................... 59
Define a New Migration Overview ....................................................... 59
Complete First Steps Using DMA ......................................................... 59
   1. Describe the Migration Environment ........................................... 60
      Establish a Setup ........................................................................... 60
      Transfer to the Next Screen ......................................................... 60
   2. Specify a New Starting Point ......................................................... 60
      Transfer to the Next Screen ......................................................... 61
      View a List of Starting Points ....................................................... 61
   3. Accept or Override Defaults ......................................................... 61
      Transfer to the Next Screen ......................................................... 62
   4. Specify Starting Point Overrides .................................................. 62
      Transfer to the Next Screen ......................................................... 62
      Parameter Reference ..................................................................... 62
   5. Include/Exclude Source Types (Current Starting Point) ................ 62
      Transfer to the Next Screen ......................................................... 62
      Parameter Reference ..................................................................... 63
   6. Include/Exclude Load Types (Current Starting Point) .................... 63
      Transfer to the Next Screen ......................................................... 63
      Parameter Reference ..................................................................... 63
   7. Obtain CA IDMS Dictionary Migrator Reports ............................... 63
      Specify Migration Output .............................................................. 64
      Transfer to the Next Screen ......................................................... 64
      Parameter Reference ..................................................................... 64
   8. Store Setup into a PARMFILE ....................................................... 64
      What You Can Store ..................................................................... 64
      How to Store ................................................................................ 65
      Transfer to the Next Screen ......................................................... 65
   9. Continue to Use DMA .................................................................. 65
Global Overrides ................................................................................................................. 65
  DMA Provides a Default Starting Point ........................................................................... 67
Use a Series of Commands from the Menu ......................................................................... 67
Specify Migration Overrides ................................................................................................. 67
  Transfer to the Next Screen ......................................................................................... 67
  Parameter Reference ................................................................................................. 68
Limit Entities for Overall Migration .................................................................................... 68
  Include/Exclude SOURCE Types (Overall Migration) .................................................... 68
  Use the Command SWAP ......................................................................................... 68
  Transfer to the Next Screen ......................................................................................... 68
  Parameter Reference ................................................................................................. 68
  Include/Exclude LOAD Types (Overall Migration) ....................................................... 69
  Use the Command SWAP ......................................................................................... 69
  Transfer to the Next Screen ......................................................................................... 69
  Parameter Reference ................................................................................................. 69
  View the Parameter Statements ................................................................................... 69
  STORE Before Submitting ......................................................................................... 70
Select a DMA Parameter File .............................................................................................. 70
  Select Multiple DMA Parameter Files ......................................................................... 70
Specify a DMA Parameter File .......................................................................................... 70
Modify a DMA Parameter File .......................................................................................... 71
  Steps to Modify a DMA Parameter File ....................................................................... 71
  Modify Dictionary/Starting Point Values ....................................................................... 71
  Modify Parameters ....................................................................................................... 71
  Modify JCL .................................................................................................................. 72
    Display JCL ............................................................................................................... 72
    Save Modifications ..................................................................................................... 72
    Change the Default JCL ............................................................................................ 73
Copy a DMA Parameter File ............................................................................................... 73
Submit Parameters and JCL Online .................................................................................... 74
  File Allocations .......................................................................................................... 74
Submit Parameters and JCL from the Batch Environment ................................................ 74
Copy CA IDMS/Dictionary Migrator JCL to DMA ............................................................. 75

Extract Paths ...................................................................................................................... 75
Extract Path for DIALOG ................................................................................................. 76
  EXTRACT, LEVEL = DIALOG, DIALOG = dialog-name ............................................... 76
Extract Path for RECORD ................................................................................................. 79
  EXTRACT, LEVEL = DIALOG, RECORD = record-name ............................................. 80
Extract Path for RECORD when LEVEL = ENTITY ......................................................... 82
EXTRACT, RECORD = record-name, LEVEL = ENTITY ......................................................... 83
Extract Path for ATTRIBUTE when LEVEL = DIALOG ...................................................... 84
   EXTRACT, ATTRIBUTE = attribute-name, LEVEL = DIALOG ........................................ 85
EXTRACT Path for ATTRIBUTE when LEVEL = ENTITY ...................................................... 88
   EXTRACT, ATTRIBUTE = attribute-name, LEVEL = ENTITY .......................................... 88
Extract Path for ATTRIBUTE when LEVEL = ONLY ......................................................... 90
   EXTRACT, ATTRIBUTE = attribute-name, LEVEL = ONLY ........................................... 91
Extract Path for SUBSCHEMA when LEVEL = DIALOG ................................................... 93
   EXTRACT, SUBSCHEMA = subschema-name, LEVEL = DIALOG ................................... 93
Extract Path for SUBSCHEMA when LEVEL = ENTITY .................................................... 96
   EXTRACT, SUBSCHEMA = subschema-name, LEVEL = ENTITY .................................... 96
Extract Path for APPLICATION when LEVEL = DIALOG ................................................ 97

Customization .................................................................................................................. 100
Introduction ..................................................................................................................... 101
Installation Parameters .................................................................................................. 102
Message Severity ............................................................................................................ 102
   Customization Instructions ......................................................................................... 102
      General .................................................................................................................... 102
      Message Severity Level Table Generation Steps ...................................................... 102
      Message Severity Level Table Customization ....................................................... 102
      Assembly and Linkage ............................................................................................ 104
Entity Exclusion Table .................................................................................................. 104
   Customization Instructions ......................................................................................... 104
      General .................................................................................................................... 104
      Entity Exclusion Table Generation Steps ................................................................. 104
      Entity Exclusion Table Customization .................................................................... 105
      Assembly and Linkage ............................................................................................ 106
      Default Entity Exclusion Table (table name USMEXCLT) ........................................ 106
Manipulation of the Extract File ..................................................................................... 106
   Extract Record ............................................................................................................. 106
   Field Definitions and Values ........................................................................................ 107
      Values for Utility Types ............................................................................................ 109
      Value for Entity Type Codes .................................................................................... 109
Change only ..................................................................................................................... 111
Entry point ....................................................................................................................... 111
Extract path ..................................................................................................................... 112
Migration .......................................................................................................................... 112
Object dictionary ............................................................................................................. 112
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source dictionary</td>
<td>112</td>
</tr>
<tr>
<td>Starting point</td>
<td>112</td>
</tr>
<tr>
<td>Syntax files</td>
<td>113</td>
</tr>
<tr>
<td>Trace</td>
<td>113</td>
</tr>
</tbody>
</table>
Using the CA IDMS Dictionary Migrator

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

- Introduction (see page 11)
- Migration Steps (see page 11)
- Definition of Migration (see page 12)
- When to Use CA IDMS Dictionary Migrator (see page 12)
- Migrate Systems Quickly and Accurately (see page 12)
- Take the Guesswork Out of Migration (see page 12)

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 section 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 sectionally, or select them using DMA.
5. Run 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
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 section will further explain performing migration.

To configure the migration you can migrate by PROGRAM and DIALOG. PROGRAM indicates that the specified program is to 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 sectionly 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 CA ADS 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.

For more information, see the following topics:
- CA IDMS Dictionary Migrator General Information (see page 13)
- CA IDMS Dictionary Migrator Operations (see page 31)
- Dictionary Migrator Assistant (see page 43)
- Extract Paths (see page 75)
- Customization (see page 100)

CA IDMS Dictionary Migrator General Information

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

- CA IDMS Dictionary Migrator Is Easy to Use (see page 14)
- Flexible Processing Capabilities (see page 14)
  - SQL Entity Migration (see page 15)
- Improve Development Productivity (see page 16)
  - New Applications Environment (see page 16)
  - Existing Applications Environment (see page 17)
- Backup and Recovery (see page 19)
- Configure the Migration to Meet Your Needs (see page 19)
  - Migrate Entire Systems (see page 19)
    - Migrate by Organizational Structure (see page 19)
    - Migrate by Organizational Structure Without Regard to Dialog (see page 20)
    - Migrate by Component Relationship (see page 21)
  - Migrate a Single Dictionary Entity (see page 22)
  - Migrate in a Multiple-Machine Environment (see page 22)
    - Single Job Execution--Using Node Communication (see page 23)
    - Two-Job Execution (see page 23)
    - Advantages of the Two-Job Execution (see page 24)
  - Extract Entities from Multiple Source Dictionaries (see page 24)
CA IDMS Dictionary Migrator Is Easy to Use

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
CA IDMS - 19.0

- Single components with or without system relationships
- Source components from the DDLDML area
- Load components from the DDLDCLOD area
- Selected messages from the DDLDCMSG area
- Changed entities only.
- SQL catalog definitions from the DDLCAT 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 USMCOPY. 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.

CA IDMS Dictionary Migrator Information

This section provides the following information:

- Comprehensive Reports (see page 25)
• Online Aid for Selecting Parameters (see page 26)
  • DMA Your CA IDMS Dictionary Migrator Connection (see page 26)
    • Establish Several Migration Setups in One Session (see page 26)
• Choose Parameters for Migration (see page 27)
  • DMA Functions (see page 27)
  • User Responsibilities (see page 27)
  • CA IDMS Dictionary Migrator Functions (see page 27)
• Use DMA to Set up a Migration Easily (see page 27)
  • Alternative to Learning CA IDMS Dictionary Migrator Parameters (see page 28)
• Menu-Driven Parameter Selection (see page 28)
• Parameter Storage (see page 28)
• Online Help (see page 29)
  • When to Review the Output (see page 29)
    • Method One (see page 29)
    • Method Two (see page 29)
    • Method Three (see page 30)
• Obtain Reports (see page 30)
• View Syntax for Migration (see page 31)
• View JCL for Migration (see page 31)

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 section.
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 on 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

15-Jan-2018
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 on 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 CREATESYNTAX
- 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 Section 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
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 section "Dictionary Migrator Assistant."

CA IDMS Dictionary Migrator Operations

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

Contents

- Operational Considerations (see page 32)
  - Local Mode Processing (see page 33)
- Additional Tasks Performed by CA IDMS Dictionary Migrator (see page 33)
- How CA IDMS Dictionary Migrator Operates (see page 33)
  - Two Methods of Operation (see page 34)
- Allocating Space for Work Files and Syntax Files (see page 35)
  - Changing Block Size of Work Files and Syntax Files in z/OS (see page 36)
  - z/VSE File Assignments (see page 36)
  - z/VSE File Allocation Alternate Method (see page 36)
- Running as a Single Job Execution Under Central Version (see page 38)
  - Migration Activities - Single Job Execution (see page 38)
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 Section.

- 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, DDLDLCMSG 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 on the necessary parameters, see the *CA IDMS Common Facilities Section*. 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).
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 on 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.

<table>
<thead>
<tr>
<th>File Name (z/VSE Logical Unit)</th>
<th>Required Type</th>
<th>LRECL Suggested Space</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIGPARM</td>
<td>yes</td>
<td>disk 80, tape 1, card 1</td>
<td>z/OS and z/VM--Input file for CA IDMS Dictionary Migrator parameter statement</td>
</tr>
<tr>
<td>SYSDPC (SYS018)</td>
<td>yes</td>
<td>disk 80, 30,60</td>
<td>Input to CA IDMS IDMSDDDL, IDMSSUBC, RHDCMPUT, and IDMSCHEM utilities</td>
</tr>
<tr>
<td>SYSIPT (SYS012)</td>
<td>yes</td>
<td>disk 80, 150,60</td>
<td>Output from IDMSDDDL, IDMSSUBC, RHDCMPUT, and IDMSCHEM utilities</td>
</tr>
<tr>
<td>SELECT (SYS014)</td>
<td>**</td>
<td>disk 176, 2,1</td>
<td>Contains encoded extract statements</td>
</tr>
</tbody>
</table>
**Exhibit 5.2: Work Files Table**

<table>
<thead>
<tr>
<th>Step</th>
<th>Syntax File (z Required Type /VSE Logical Unit)</th>
<th>LRECL Suggested Space</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EXTRACT</td>
<td>yes disk 228 150,60</td>
<td>Contains entities from source dictionary as they are extracted</td>
</tr>
<tr>
<td></td>
<td>VSAMEXT</td>
<td>** VSAM KSDS 228 60,30</td>
<td>Contains extracted entities for dictionary comparisons</td>
</tr>
<tr>
<td></td>
<td>VALDRPT</td>
<td>** disk 138 5,2</td>
<td>Sorts and controls information from dictionary comparisons for reports</td>
</tr>
<tr>
<td></td>
<td>WORKFIL</td>
<td>** disk 80 60,30</td>
<td>Temporary holding file for syntax-creation or modification</td>
</tr>
<tr>
<td></td>
<td>WORKFL2 (SYS034)</td>
<td>** disk 228 60,30</td>
<td>Required to construct combination file for entities extracted from the object dictionary</td>
</tr>
<tr>
<td></td>
<td>SYSCTL</td>
<td>*** disk 20 pre-allocated share</td>
<td>Defines access to the central version which will be the gateway CV if a node is specified.</td>
</tr>
<tr>
<td></td>
<td>SYSIDMS</td>
<td>*** card 80 pre-allocated share</td>
<td>See the DBA Reference. Never specify DBNAME, NODENAME, DICTNODE, or DICTNAME. Use DMCL=dmclname, LOCAL=ON, and JOURNAL=OFF to run local (minicv) mode.</td>
</tr>
</tbody>
</table>

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

**File requirement depends on the run type.**

***File requirement depends on environment.**

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

**File requirement depends on the run type.**

***File requirement depends on environment.**
### Migration Activities - Single Job Execution

<table>
<thead>
<tr>
<th>Step</th>
<th>Job Name (SYSON)</th>
<th>Call Level</th>
<th>Activity Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>DDDLLOD (SYS025)</td>
<td>5,2</td>
<td>ADD/Modify of load modules for subschemas, maps, map editing tables, and dialog processes.</td>
</tr>
<tr>
<td>9</td>
<td>SCHMUPD (SYS026)</td>
<td>5,2</td>
<td>Upload of schema modifications -- source statements written in schema DDL and any data copied from the dictionary to add schemas.</td>
</tr>
<tr>
<td>10</td>
<td>SUBSUPD (SYS028)</td>
<td>5,2</td>
<td>Upload of subschemas.</td>
</tr>
<tr>
<td>11</td>
<td>SUBSLOD (SYS029)</td>
<td>5,2</td>
<td>Generates subschema load modules in DDLDCLOD area.</td>
</tr>
<tr>
<td>12</td>
<td>RHDCUPD (SYS030)</td>
<td>5,2</td>
<td>ADD/MODIFY of panels and maps.</td>
</tr>
<tr>
<td>13</td>
<td>RHDCLOD (SYS031)</td>
<td>60,30</td>
<td>Compiles maps and stores map load modules in DDLDCLOD area.</td>
</tr>
<tr>
<td>14</td>
<td>ADSOBOGN (SYS032)</td>
<td>1,2</td>
<td>Generates dialogs. This step can only be executed if dialog load modules were migrated.</td>
</tr>
<tr>
<td>15</td>
<td>DDDLPGM (SYS033)</td>
<td>5,2</td>
<td>ADD/MODIFY of program statements.</td>
</tr>
<tr>
<td>16</td>
<td>ADSBTAT (SYS037)</td>
<td>1,1</td>
<td>Updates Task Application Table (revised ADSA applications only).</td>
</tr>
<tr>
<td>17</td>
<td>USERUPD (SYS038)</td>
<td>1,1</td>
<td>Updates passwords on migrated user records to reflect source database.</td>
</tr>
</tbody>
</table>

*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--USMSYNTX -- z/OS Only

The JCL model contained in the distribution SAMPJCL library member USMSYNTX (z/OS only) allocates the syntax files. The key to the model JCL is also contained in USMSYNTX. 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 -- USMSYNTX--z/OS Only

The JCL model contained in the distribution SAMPJCL library member USMSYNTX (z/OS) allocates the syntax files. The key to the model JCL is also contained in USMSYNTX. 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 USMIMPRT 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.

---

**Dictionary Migrator Assistant**

**Contents**
- Introduction (see page 46)
- DMA Requirements (see page 46)
  - Installation (see page 46)
  - System Requirements (see page 46)
- Terminal Type (see page 46)
- Security (see page 46)
- Storage Requirements (see page 47)
  - Program Storage (see page 47)
  - Working Storage (see page 47)
  - Scratch Storage (see page 47)
  - Database Storage (see page 47)
- User Authority (see page 47)
- Types of Messages (see page 48)
  - Screen Messages (see page 48)
  - Confirm Screens (see page 48)
- DMA Commands (see page 49)
  - Enter a Command (see page 49)
  - System Commands Menu (see page 50)
    - Multiple Commands (see page 51)
- Display PF Key Values (see page 51)
  - Access the Values Screen (see page 51)
  - Change a PF Key Value (see page 51)
  - Which PF Key Values Apply? (see page 51)
- Questions and Answers About Using DMA (see page 52)
  - What are the PF Key Values? (see page 52)
  - What are the System Commands? (see page 52)
  - What Should I Enter On a DMA Screen? (see page 53)
  - Can I Perform DMA Activities? (see page 53)
  - How Will A Migration Affect The Target Dictionary? (see page 53)
  - How Do I Use the Dictionary? (see page 53)
- First Steps in Using DMA (see page 53)
  - Access DMA (see page 53)
  - Select DMA Activity (see page 54)
- Reset and Exit DMA (see page 54)
  - Reset DMA (see page 54)
  - Exit DMA (see page 55)
- Common Fields on DMA Screens (see page 56)
- DMA Activity Paths (see page 56)
- Define a New Migration Sequence of Screens (see page 57)
  - 1. Specify Overrides of DMA-Supplied Defaults (see page 59)
  - 2. Accept DMA-Supplied Defaults (see page 59)
- Define a New Migration Overview (see page 59)
- Complete First Steps Using DMA (see page 59)
  - 1. Describe the Migration Environment (see page 60)
    - Establish a Setup (see page 60)
    - Transfer to the Next Screen (see page 60)
  - 2. Specify a New Starting Point (see page 60)
- Transfer to the Next Screen (see page 61)
- View a List of Starting Points (see page 61)
- 3. Accept or Override Defaults (see page 61)
  - Transfer to the Next Screen (see page 62)
- 4. Specify Starting Point Overrides (see page 62)
  - Transfer to the Next Screen (see page 62)
  - Parameter Reference (see page 62)
- 5. Include/Exclude Source Types (Current Starting Point) (see page 62)
  - Transfer to the Next Screen (see page 62)
  - Parameter Reference (see page 63)
- 6. Include/Exclude Load Types (Current Starting Point) (see page 63)
  - Transfer to the Next Screen (see page 63)
  - Parameter Reference (see page 63)
- 7. Obtain CA IDMS Dictionary Migrator Reports (see page 63)
  - Specify Migration Output (see page 64)
  - Transfer to the Next Screen (see page 64)
  - Parameter Reference (see page 64)
- 8. Store Setup into a PARMFILE (see page 64)
  - What You Can Store (see page 64)
  - How to Store (see page 65)
  - Transfer to the Next Screen (see page 65)
- 9. Continue to Use DMA (see page 65)
- Global Overrides (see page 65)
  - DMA Provides a Default Starting Point (see page 67)
- Use a Series of Commands from the Menu (see page 67)
  - Specify Migration Overrides (see page 67)
    - Transfer to the Next Screen (see page 67)
    - Parameter Reference (see page 68)
- Limit Entities for Overall Migration (see page 68)
  - Include/Exclude SOURCE Types (Overall Migration) (see page 68)
  - Use the Command SWAP (see page 68)
  - Transfer to the Next Screen (see page 68)
  - Parameter Reference (see page 68)
  - Include/Exclude LOAD Types (Overall Migration) (see page 69)
  - Use the Command SWAP (see page 69)
  - Transfer to the Next Screen (see page 69)
  - Parameter Reference (see page 69)
  - View the Parameter Statements (see page 69)
  - STORE Before Submitting (see page 70)
- Select a DMA Parameter File (see page 70)
  - Select Multiple DMA Parameter Files (see page 70)
- Specify a DMA Parameter File (see page 70)
- Modify a DMA Parameter File (see page 71)
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 Section.

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 section "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
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 Confirm Session 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.
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

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.

(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 - Display 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.

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).

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 sections.

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 task code 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.

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 section "Messages".

- **Screen name**—the top right of the screen (XMANNU). Unique identifier for screen.

CA IDMS/MA Rnn.nn --- System Commands ---------------------------------------- hh:mm mm/dd/yy

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  - Display 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
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?
IDMSDB--Defining a New Migration
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 section 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.
- 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?
- 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?
- 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 on 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.
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.

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 on the requested entities located in the source and target dictionaries.

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

⚠️ Note: For more information on CA IDMS Dictionary Migrator reports, see the section "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 on 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 on 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.
IDMSDB--Using Global Overrides--DIA
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.

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 on 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/Exclude SOURCE Types screen for the OVERALL Migration are retained as default values unless you enter other values on the Include/Exclude SOURCE 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 on the Include/Exclude SOURCE 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 on 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 71).
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 File Name?

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.
Modify JCL

Before submitting JCL and parameters, you should modify the CA IDMS Dictionary Migrator JCL to reflect conditions at your site. See 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.
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 . 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
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 XDMBJCL.S (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 //MIGPARM DD*. The JCL must not contain any migrator parameter statements.

Extract Paths

This section 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 section contains the following topics:

- Extract Path for DIALOG (see page 76)
  - EXTRACT, LEVEL = DIALOG, DIALOG = dialog-name (see page 76)
- Extract Path for RECORD (see page 79)
  - EXTRACT, LEVEL = DIALOG, RECORD = record-name (see page 80)
- Extract Path for RECORD when LEVEL = ENTITY (see page 82)
  - EXTRACT, RECORD = record-name, LEVEL = ENTITY (see page 83)
- Extract Path for ATTRIBUTE when LEVEL = DIALOG (see page 84)
  - EXTRACT, ATTRIBUTE = attribute-name, LEVEL = DIALOG (see page 85)
- EXTRACT Path for ATTRIBUTE when LEVEL = ENTITY (see page 88)
  - EXTRACT, ATTRIBUTE = attribute-name, LEVEL = ENTITY (see page 88)
- Extract Path for ATTRIBUTE when LEVEL = ONLY (see page 90)
  - EXTRACT, ATTRIBUTE = attribute-name, LEVEL = ONLY (see page 91)
- Extract Path for SUBSCHEMA when LEVEL = DIALOG (see page 93)
  - EXTRACT, SUBSCHEMA = subschema-name, LEVEL = DIALOG (see page 93)
- Extract Path for SUBSCHEMA when LEVEL = ENTITY (see page 96)
  - EXTRACT, SUBSCHEMA = subschema-name, LEVEL = ENTITY (see page 96)
- Extract Path for APPLICATION when LEVEL = DIALOG (see page 97)

This section 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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>--(dotted line)</td>
<td>Path from Entry Point to Starting Point of Extraction.</td>
</tr>
<tr>
<td>_ (solid line)</td>
<td>Path from Starting Point to all extracted components.</td>
</tr>
<tr>
<td>*</td>
<td>Points to where the Extract Statement goes.</td>
</tr>
</tbody>
</table>

**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 DDLDCLOD (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 DDLDCLOD (load) area
       - Extract each Logical Record Facility work record used by the subschema
       - For each record connected to 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 DDLDCLOD (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 DDDLCLCLOD (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 DDLDCLOD (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 DDLDCLOD (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 DDLDCLOD (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 DDDLCLOD (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 DDLDCLOD (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 DDLDCLOD (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 DDLDCLOD (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 DDDLCLOD (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.
IDMSDB--Attribute--DIA
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 DDLDCLOD (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 DDLDCLOD (load) area

7. Extract all maps connected to the attribute
- Extract the corresponding map from DDLDCLOD (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 DDLDCLOD (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 DDLDCLOD (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 DDLDCLOD (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 DDLDCLOD (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 DDDLCLOD (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 DDLDCLOD (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 DDLCLOD (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 DDLCMSG (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 DDLDCLOD (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 DDLDCLOD (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 DDDLCLOD (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.
Customization

Contents
This appendix contains the following topics:

Introduction

CA IDMS Dictionary Migrator can be customized in the following general areas:

<table>
<thead>
<tr>
<th>Area</th>
<th>Customization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation Parameters</td>
<td>A set of options that are made to non-standard processing to meet special situations described in the CA IDMS installation sections.</td>
</tr>
<tr>
<td>Message Severity</td>
<td>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.</td>
</tr>
<tr>
<td>Entity Exclusion Table</td>
<td>The exclusion table can define any entity occurrences that you do not want migrated.</td>
</tr>
</tbody>
</table>
1. Area Customization

<table>
<thead>
<tr>
<th>Area</th>
<th>Customization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manipulation of the Extract File</td>
<td>The format of the extract file is described to help you when you want to use the extract file for any further processing.</td>
</tr>
</tbody>
</table>

Installation Parameters

⚠️ **Note:** For more information on these parameters, see the CA IDMS installation sections.

Message Severity

This section contains information on 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 link edit 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:

First Line:

`#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 #EXCLENT macro with the required parameters:

`#EXCLENT 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 Occurrence Name 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:
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).
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 CA IDMS 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-DEFINITION-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

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special Record</td>
<td>value '01'</td>
</tr>
<tr>
<td>IDMSQDDL</td>
<td>value '06'</td>
</tr>
<tr>
<td>Idmsdddl</td>
<td>value '10'</td>
</tr>
<tr>
<td>Tablddl</td>
<td>value '12' IDMSQDDL for tables only</td>
</tr>
<tr>
<td>Loadddl</td>
<td>value '15' IDMSQDDL for load modules only</td>
</tr>
<tr>
<td>reserved</td>
<td>value '20'</td>
</tr>
<tr>
<td>Idmschem</td>
<td>value '25'</td>
</tr>
<tr>
<td>Objdchem</td>
<td>value '26' IDMSCHM from object dictionary</td>
</tr>
<tr>
<td>reserved</td>
<td>value '28'</td>
</tr>
<tr>
<td>Idmsubsc</td>
<td>value '30'</td>
</tr>
<tr>
<td>Objdubc</td>
<td>value '31' IDMSUBSC from object dictionary</td>
</tr>
<tr>
<td>Rhdcmput</td>
<td>value '40'</td>
</tr>
<tr>
<td>Objdmput</td>
<td>value '41' RHDCMPUT from object dictionary</td>
</tr>
<tr>
<td>Progdddl</td>
<td>value '50' IDMSQDDL for programs only; also creates ADSOBCOM statements for dialogs</td>
</tr>
<tr>
<td>Objpdddl</td>
<td>value '51' ADSOBCOM statement from object dictionary.</td>
</tr>
<tr>
<td>reserved</td>
<td>value '90'</td>
</tr>
<tr>
<td>reserved</td>
<td>value '95'</td>
</tr>
</tbody>
</table>

### Value for Entity Type Codes

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special weight (used for CA IDMS Dictionary Migrator control)</td>
<td>value +01</td>
</tr>
<tr>
<td>Message delete</td>
<td>value +03</td>
</tr>
<tr>
<td>Load delete</td>
<td>value +05</td>
</tr>
<tr>
<td>Dialog delete</td>
<td>value +10</td>
</tr>
<tr>
<td>Program delete</td>
<td>value +11</td>
</tr>
<tr>
<td>Delete Type</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Subschema delete</td>
<td>+20</td>
</tr>
<tr>
<td>Reserved</td>
<td>+23</td>
</tr>
<tr>
<td>Schema delete</td>
<td>+25</td>
</tr>
<tr>
<td>Map delete</td>
<td>+28</td>
</tr>
<tr>
<td>Table delete</td>
<td>+29</td>
</tr>
<tr>
<td>Process delete</td>
<td>+30</td>
</tr>
<tr>
<td>Qfile delete</td>
<td>+31</td>
</tr>
<tr>
<td>Reserved</td>
<td>+33</td>
</tr>
<tr>
<td>Module delete</td>
<td>+35</td>
</tr>
<tr>
<td>File delete</td>
<td>+36</td>
</tr>
<tr>
<td>Panel delete</td>
<td>+37</td>
</tr>
<tr>
<td>Record delete</td>
<td>+39</td>
</tr>
<tr>
<td>Report delete</td>
<td>+41</td>
</tr>
<tr>
<td>Transaction delete</td>
<td>+43</td>
</tr>
<tr>
<td>Element delete</td>
<td>+45</td>
</tr>
<tr>
<td>System delete</td>
<td>+46</td>
</tr>
<tr>
<td>Attribute delete</td>
<td>+47</td>
</tr>
<tr>
<td>User delete</td>
<td>+48</td>
</tr>
<tr>
<td>Class delete</td>
<td>+49</td>
</tr>
<tr>
<td>Class</td>
<td>+50</td>
</tr>
<tr>
<td>Module</td>
<td>+51</td>
</tr>
<tr>
<td>User (pre)</td>
<td>+52</td>
</tr>
<tr>
<td>Attribute</td>
<td>+54</td>
</tr>
<tr>
<td>System</td>
<td>+56</td>
</tr>
<tr>
<td>Application</td>
<td>+57</td>
</tr>
<tr>
<td>Element</td>
<td>+60</td>
</tr>
<tr>
<td>Reserved</td>
<td>+62</td>
</tr>
<tr>
<td>Qfile</td>
<td>+63</td>
</tr>
<tr>
<td>Process</td>
<td>+64</td>
</tr>
<tr>
<td>Table</td>
<td>+66</td>
</tr>
<tr>
<td>File</td>
<td>+68</td>
</tr>
<tr>
<td>Record</td>
<td>+70</td>
</tr>
<tr>
<td>Report</td>
<td>+71</td>
</tr>
<tr>
<td>Transaction</td>
<td>+72</td>
</tr>
<tr>
<td>Panel</td>
<td>+73</td>
</tr>
<tr>
<td>Schema</td>
<td>+75</td>
</tr>
<tr>
<td>Item</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
</tr>
<tr>
<td>Reserved</td>
<td>value +85</td>
</tr>
<tr>
<td>User (post)</td>
<td>value +88</td>
</tr>
<tr>
<td>Map</td>
<td>value +78</td>
</tr>
<tr>
<td>Subschema</td>
<td>value +79</td>
</tr>
<tr>
<td>Program</td>
<td>value +80</td>
</tr>
<tr>
<td>Dialog</td>
<td>value +82</td>
</tr>
<tr>
<td>Reserved</td>
<td>value +86</td>
</tr>
<tr>
<td>Reserved</td>
<td>value +87</td>
</tr>
<tr>
<td>Load Module</td>
<td>value +90</td>
</tr>
<tr>
<td>SQL SCHEMA</td>
<td>value +91</td>
</tr>
<tr>
<td>Reserved</td>
<td>value +92</td>
</tr>
<tr>
<td>SQL TABLE</td>
<td>value +93</td>
</tr>
<tr>
<td>SQL VIEW</td>
<td>value +94</td>
</tr>
<tr>
<td>Message</td>
<td>value +95</td>
</tr>
<tr>
<td>Reserved</td>
<td>value +96</td>
</tr>
<tr>
<td>CONSTRAINT</td>
<td>value +97</td>
</tr>
<tr>
<td>TABLE PROCEDURE</td>
<td>value +98</td>
</tr>
<tr>
<td>SQL KEY</td>
<td>value +99</td>
</tr>
</tbody>
</table>

**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.