

CA-IDMS[®]

Presspack User Guide
15.0



Computer Associates™

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

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

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

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

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

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

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

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

Second Edition, October 2001

© 2001 Computer Associates International, Inc.
All rights reserved.

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

Contents

How to Use This Manual	v
Chapter 1. Overview of CA-IDMS Presspack	1-1
1.1 About this chapter	1-4
1.2 About CA-IDMS Presspack	1-5
1.3 Guidelines for using CA-IDMS Presspack	1-7
1.4 Using IDMSPASS	1-9
Chapter 2. Enabling CA-IDMS Presspack Compression	2-1
2.1 About this chapter	2-4
2.2 Defining records and tables for compression	2-5
2.2.1 Using SCHEMA RECORD syntax	2-5
2.2.2 Using CREATE TABLE syntax	2-7
2.3 Converting schema records	2-8
2.4 Converting tables	2-11
2.5 CA-IDMS Presspack runtime execution	2-12
Chapter 3. Creating DCTs with IDMSPASS	3-1
3.1 About IDMSPASS	3-4
3.2 About this chapter	3-5
3.3 Running IDMSPASS	3-6
3.4 IDMSPASS syntax	3-8
3.4.1 Usage	3-10
3.4.2 Examples	3-12
3.5 What you can do with IDMSPASS	3-14
3.5.1 Generating a statistical report	3-14
3.5.2 Reading the IDMSPASS report	3-15
3.5.3 Creating DCTs	3-17
Appendix A. Reading Compressed Data From the Journal	A-1
Appendix B. Sample JCL	B-1
B.1 About this appendix	B-4
B.2 OS/390 JCL	B-5
B.2.1 Usage	B-6
B.3 VSE/ESA JCL	B-8
B.3.1 Usage	B-9
B.4 VM/ESA commands	B-10
B.4.1 Usage	B-11
B.5 BS2000/OSD JCL	B-12
B.5.1 Usage	B-13
Index	X-1

How to Use This Manual

What this manual is about

This manual contains information on CA-IDMS Presspack, a large-volume database compression and decompression tool for use with CA-IDMS/DB:

- **Chapter 1** provides a general description of CA-IDMS Presspack and its components, and gives guidelines for using the product
- **Chapter 2** shows how to convert your database records or tables to CA-IDMS Presspack compression
- **Chapter 3** documents the uses and syntax of IDMSPASS, the CA-IDMS Presspack utility that you use to:
 - Report on the records or tables that you have selected for possible compression
 - Create custom data characteristic tables (DCTs) for CA-IDMS Presspack to use in compression/decompression functions
- **Appendix A** shows how you can access compressed records that have been journaled
- **Appendix B** provides sample IDMSPASS JCL

Who should use this manual

- Database administrators
- Database designers
- Others responsible for implementing or designing a CA-IDMS/DB database

How to proceed

This manual is both a user guide and a reference tool. To obtain the best results from using the manual, you should:

- Read Chapters 1 through 3, familiarizing yourself with CA-IDMS Presspack and with what you have to do to use the product. In this first read-through, you can postpone concentrating on reference items, such as syntax, syntax rules, JCL, and error codes and messages.
- Return to the manual for specific information that you need to use the product. This may include reference items and procedural information.

Related documentation

Manuals that supplement information in this manual include:

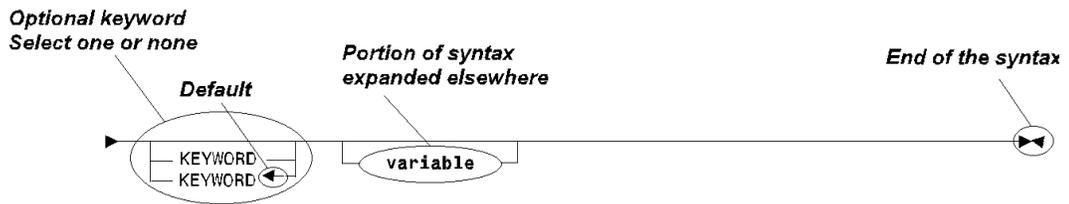
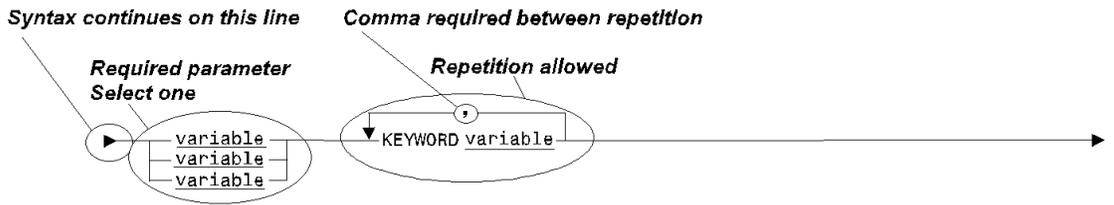
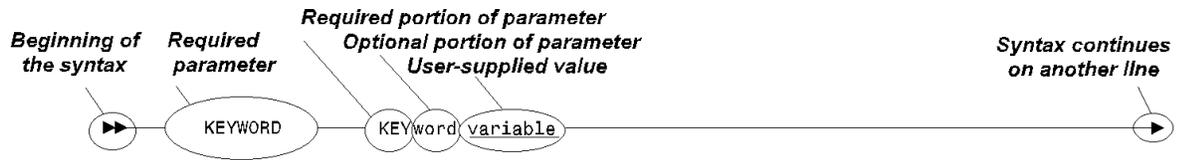
- *CA-IDMS Utilities*
- *CA-IDMS Command Facility*
- *CA-IDMS SQL Reference Guide*
- *CA-IDMS Database Design Guide*
- *CA-IDMS Database Administration*
- *CA-IDMS Navigational DML Programming*
- *CA-IDMS SQL Programming Guide*
- *CA-IDMS Messages and Codes*

Understanding Syntax Diagrams

Look at the list of notation conventions below to see how syntax is presented in this manual. The example following the list shows how the conventions are used.

UPPERCASE OR SPECIAL CHARACTERS	Represents a required keyword, partial keyword, character, or symbol that must be entered completely as shown.
lowercase	Represents an optional keyword or partial keyword that, if used, must be entered completely as shown.
<u>underlined lowercase</u>	Represents a value that you supply.
◀	Points to the default in a list of choices.
lowercase bold	Represents a portion of the syntax shown in greater detail at the end of the syntax or elsewhere in the document.
▶▶	Shows the beginning of a complete piece of syntax.
◀◀	Shows the end of a complete piece of syntax.
▶	Shows that the syntax continues on the next line.
▶	Shows that the syntax continues on this line.
▶	Shows that the parameter continues on the next line.
▶	Shows that a parameter continues on this line.
▶ parameter ▶	Shows a required parameter.
▶ parameter parameter ▶	Shows a choice of required parameters. You must select one.
▶ parameter ▶	Shows an optional parameter.
▶ parameter parameter ▶	Shows a choice of optional parameters. Select one or none.
▶ parameter ▶	Shows that you can repeat the parameter or specify more than one parameter.
▶ parameter , parameter ▶	Shows that you must enter a comma between repetitions of the parameter.

Sample Syntax Diagram



Chapter 1. Overview of CA-IDMS Presspack

- 1.1 About this chapter 1-4
- 1.2 About CA-IDMS Presspack 1-5
- 1.3 Guidelines for using CA-IDMS Presspack 1-7
- 1.4 Using IDMSPASS 1-9

1.1 About this chapter

This chapter provides a general description of CA-IDMS Presspack, a large-volume database compression and decompression tool for CA-IDMS/DB. It also gives guidelines for using this product and refers you to other chapters of this manual for details.

1.2 About CA-IDMS Presspack

What is CA-IDMS Presspack: CA-IDMS Presspack is a tool for compression and decompression of CA-IDMS/DB records or tables. Its compression efficiency makes it ideal for compressing large-volume databases.

CA-IDMS Presspack consists of:

- Runtime compression and decompression routines
- A built-in data characteristic table (DCT) for use in compression and decompression functions
- IDMSPASS, a utility that allows you to create custom DCTs and thus enhance the efficiency of CA-IDMS Presspack compression

Advantages of using CA-IDMS Presspack: Compressing your database records:

- Slows the demand for new DASD.
By compressing data you will be using your DASD more efficiently and as a result, will be able to defer the purchase of additional hardware.
- Alleviates physical space constraints in your data center.
Additional DASD requires physical space that may not be available.
- Helps I/O bound jobs.
Since more data is stored in the same amount of space, fewer I/Os will be required to retrieve the same amount of data.
- Reduces journaling activity.
Since the records that are written to the journals are compressed, more records fit onto a journal and, as a result, you do not have to offload the journals as often.
- Speeds backup and restoration of your database.
Since the same amount of data can be stored on fewer pages, the backup/restore utilities have to do fewer I/Os and, as a result, run faster.

What CA-IDMS Presspack compresses: CA-IDMS/DB records or tables may contain control field data and non-control field data. CA-IDMS Presspack compresses the data beyond the last control field.

How CA-IDMS Presspack compresses: CA-IDMS Presspack compresses database records by:

1. Eliminating repeating character strings
2. Using the Huffman Encoding Technique
3. Applying the concept of successor tables

The result is very efficient compression of your data.

Huffman Encoding Technique: CA-IDMS Presspack uses an algorithm called the Huffman Encoding Technique to assign bit strings of varying length to characters that are being compressed. Characters that occur most frequently are assigned the shortest bit strings and those that occur less frequently are assigned longer bit strings. The result is that instead of every character using a fixed eight bit code, some characters are represented by as few as two bits and a few characters require more than eight bits.

Since the most frequently occurring characters get the shortest bit strings, the result is that the set of data requires less space to be stored.

Successor tables: The successor table concept allows for the use of multiple Huffman conversion tables. CA-IDMS Presspack decides which table to use on the next character (that is, the "succeeding" character) on the basis of the type of character it expects to compress or decompress next. This technique is based on the premise that user data occurs in patterns and not in a random fashion.

By applying the concept of successor tables, CA-IDMS Presspack can achieve further compression by assigning the same short bit string to more than one frequently occurring character. For example, the most frequently occurring alphabetic character and the most frequently occurring digit could both be represented by the shortest bit string. The number of successor tables used is related to the degree of compression you specify when you create a custom DCT.

How you can optimize compression: You can optimize compression by passing selected records or tables through IDMSPASS, the compression optimization utility of CA-IDMS Presspack. IDMSPASS allows you to achieve the highest degree of compression possible with CA-IDMS Presspack.

►► For further information about IDMSPASS, see 1.4, "Using IDMSPASS" on page 1-9 later in this chapter.

1.3 Guidelines for using CA-IDMS Presspack

When to use CA-IDMS Presspack: Consider using CA-IDMS Presspack for:

- Fixed length records or tables that can be compressed by more than 11 bytes.
CA-IDMS Presspack converts fixed-length records to variable length records, which require 8 bytes of overhead. CA-IDMS Presspack also requires an additional 3 bytes per record in overhead.
- Infrequently updated records.
- Long-lived records.

Also consider using CA-IDMS Presspack to compress records that contain data in unusual formats, such as binary or packed decimal. However, to obtain the most efficient compression for these records, you should first pass them through the IDMSPASS utility.

►► For more information about IDMSPASS, see 1.4, “Using IDMSPASS” on page 1-9 later in this chapter.

When not to use CA-IDMS Presspack: You do not gain the advantages of using CA-IDMS Presspack if:

- CPU overhead (if any) exceeds the space savings that you can obtain for a record or table
- You cannot obtain more than 11 bytes in space savings for a fixed-length record, or 3 bytes for a variable length record or table
- The compressible portion of the record or table (that is, what follows the last index or CALC key field or column) is very small

Required steps: These are the required steps for implementing CA-IDMS Presspack:

1. Install CA-IDMS Presspack.
 - For information about how to do this, refer to the appropriate installation document supplied with the tape.
 - *CA-IDMS Installation and Maintenance Guide - OS/390*
 - *CA-IDMS Installation and Maintenance Guide - VSE/ESA*
 - *CA-IDMS Installation and Maintenance Guide - VM/ESA*
 - *CA-IDMS Installation and Maintenance Guide - BS2000/OSD*
 - *CA-IDMS Installation and Maintenance Guide - MSP*
2. Define or convert selected records or tables for CA-IDMS Presspack compression and decompression.

►► For more information, see Chapter 2, “Enabling CA-IDMS Presspack Compression” on page 2-1.

Optional steps: To maximize the advantage you can gain from using CA-IDMS Presspack, you can use the IDMSPASS utility to:

- Generate a statistical report that will tell you how much benefit you can get from using CA-IDMS Presspack
 - For more information, see 3.5.1, “Generating a statistical report” on page 3-14.

- Create one or more custom DCTs for use with IDMSPASS.

This step is optional because if you do not create a custom DCT, you can direct CA-IDMS Presspack to use BUILTIN, its generic DCT, when compressing and decompressing records or tables defined for CA-IDMS Presspack compression. However, a custom DCT assures more efficient compression than the BUILTIN DCT.

►► For information about how to create a DCT, see 3.5.3, “Creating DCTs” on page 3-17.

1.4 Using IDMSPASS

How IDMSPASS works: You use the IDMSPASS utility to sample selected records or tables to gather statistics on characteristics of the data. Using these statistics, IDMSPASS can then build a custom DCT.

At runtime, CA-IDMS Presspack uses the custom DCT to compress or decompress the data in your database. Since the custom DCT contains conversion information specific to characteristics of the data, CA-IDMS Presspack can compress or decompress the data very efficiently.

Using BUILTIN: You can compress data that has not been passed through the IDMSPASS utility by specifying BUILTIN, the generic DCT, in the record or table definition.

►► For more information about defining records or tables for compression, see Chapter 2, “Enabling CA-IDMS Presspack Compression” on page 2-1.

The information in BUILTIN consists of statistics that describe a general mix of data. This allows efficient compression at runtime, though less efficient than use of a custom DCT created with IDMSPASS.

Chapter 2. Enabling CA-IDMS Presspack Compression

- 2.1 About this chapter 2-4
- 2.2 Defining records and tables for compression 2-5
 - 2.2.1 Using SCHEMA RECORD syntax 2-5
 - 2.2.2 Using CREATE TABLE syntax 2-7
- 2.3 Converting schema records 2-8
- 2.4 Converting tables 2-11
- 2.5 CA-IDMS Presspack runtime execution 2-12

2.1 About this chapter

This chapter tells you how to define database records or tables for compression and decompression with CA-IDMS Presspack, including:

- How to use the syntax required to define records or tables for CA-IDMS Presspack compression
- How to convert existing records or tables to (or from) CA-IDMS Presspack compression
- How CA-IDMS Presspack executes at runtime

2.2 Defining records and tables for compression

Defining schema records for compression: You define schema records for CA-IDMS Presspack compression by specifying a DCTABLE parameter on the SCHEMA RECORD statement for each record that you select. In the DCTABLE parameter you can specify the data characteristic table to be used for compression and decompression.

►► For further information about the SCHEMA RECORD statement and the DCTABLE parameter, see 2.2.1, “Using SCHEMA RECORD syntax” later in this chapter.

Defining tables for compression: You identify SQL-defined tables to be compressed by specifying the COMPRESS parameter on the CREATE TABLE statement for each table that you select to compress with CA-IDMS Presspack. Following the COMPRESS parameter, you can specify in the USING parameter the data characteristic table to be used for compression and decompression.

►► For further information about the CREATE TABLE statement and its compression parameters, see 2.2.2, “Using CREATE TABLE syntax” on page 2-7 later in this chapter.

Converting records and tables: To specify compression for records or tables that have already been defined in a CA-IDMS/DB database, you perform a conversion procedure to make the specified data available for use by CA-IDMS Presspack.

►► For information about how to convert schema records, see 2.3, “Converting schema records” on page 2-8 later in this chapter.

For information about how to convert SQL-defined tables, see 2.4, “Converting tables” on page 2-11 later in this chapter.

2.2.1 Using SCHEMA RECORD syntax

Specifying the DCTABLE parameter: You specify the DCTABLE parameter of the SCHEMA RECORD statement to enable CA-IDMS Presspack to compress and decompress the record at runtime.

►► For complete syntax documentation of the SCHEMA RECORD statement, refer to *CA-IDMS Database Administration*.

These considerations apply to the DCTABLE parameter:

- Specify *dctable-name* if you want CA-IDMS Presspack to use a custom DCT for the schema record, or BUILTIN if you want CA-IDMS Presspack to use its generic DCT

You can specify *dctable-name* before you create the DCT. However, you cannot specify *dctable-name* before you install CA-IDMS Presspack (that is, before CA-IDMS Presspack module IDDPST is linked with the schema compiler).

- Specify IS USED FOR COMPRESSION or IS USED FOR DECOMPRESSION only if you need to *convert* existing schema records to or from CA-IDMS Presspack compression; otherwise, specify IS USED FOR BOTH

►► For information about converting records, see 2.3, “Converting schema records” on page 2-8 later in this chapter.

DCTABLE parameter example: In this example, the DCTABLE parameter of a MODIFY statement names JOBDCT as the DCT used both to compress and decompress all JOB records:

```
MODIFY RECORD NAME IS JOB
      DCTABLE NAME JOBDCT IS USED FOR BOTH.
```

Associated defaults for other parameters: The DCTABLE parameter has associated defaults for the MINIMUM ROOT LENGTH and MINIMUM FRAGMENT LENGTH parameters of the SCHEMA RECORD statement. These defaults are:

- MINIMUM ROOT IS CONTROL LENGTH.
- MINIMUM FRAGMENT LENGTH IS
 - *Fragment-length* for fixed compressed records whose fragmentable length is less than or equal to 40 bytes (a record's fragmentable length is the record's total length less its control length)
 - 40, for fixed compressed records whose fragmentable length is greater than 40 bytes (a record's fragmentable length is the record's total length less its control length)
 - 4, for variable compressed records

The schema compiler automatically uses these parameter values when it processes a DCTABLE parameter.

Overriding the defaults: You can override the defaults by specifying your own values for the MINIMUM ROOT LENGTH and MINIMUM FRAGMENT LENGTH parameters of the SCHEMA RECORD statement.

For example, to prevent fragmentation, you can specify RECORD LENGTH for both MINIMUM ROOT LENGTH and MINIMUM FRAGMENT LENGTH in the SCHEMA RECORD statement. Fragmentation is valuable as an alternative to compression, not in place of compression.

2.2.2 Using CREATE TABLE syntax

Using the COMPRESS parameter: You specify the COMPRESS parameter of the CREATE TABLE statement to enable CA-IDMS Presspack to compress and decompress a table at runtime.

►► For complete documentation of CREATE TABLE statement syntax, refer to *CA-IDMS SQL Reference Guide*.

What COMPRESS means: When you specify COMPRESS on a CREATE TABLE statement, you direct the DBMS to call CA-IDMS Presspack both to compress and to decompress data in the table when DML functions are performed.

If you do not specify the USING parameter to name a DCT in conjunction with the COMPRESS parameter, CA-IDMS Presspack will use BUILTIN at runtime.

Changing the compression specification: You cannot change the compression specification with an ALTER TABLE statement. To change the compression specification, you convert the table.

►► For information about table conversion, see 2.4, “Converting tables” on page 2-11 later in this chapter.

Specifying a DCT: Specify USING *dctable-name* in conjunction with the COMPRESS parameter to direct CA-IDMS Presspack to use a custom DCT at runtime. You create a custom DCT for the table using the IDMSPASS utility.

COMPRESS parameter example: In this example, the COMPRESS parameter of a CREATE TABLE statement names JOBDCT as the DCT used both to compress and decompress the JOB table:

```
CREATE TABLE DEMOEMPL.JOB
.
.
.
IN DEMODB.EMPAREA
COMPRESS USING JOBDCT;
```

2.3 Converting schema records

What you can do: You can convert your database records by changing them from non-CA-IDMS Presspack use to CA-IDMS Presspack use, or by removing them from CA-IDMS Presspack use.

Converting records to CA-IDMS Presspack use: If you just installed CA-IDMS Presspack, the schema records in your existing CA-IDMS/DB database may be:

- Uncompressed records, either fixed-length or variable-length
- Records compressed with IDMSCOMP and decompressed with IDMSDCOM

In either case, you can convert records that you select for CA-IDMS Presspack compression and decompression.

Conversion methods: The method to use for converting existing database records to CA-IDMS Presspack depends on your objective.

To reduce area size when converting to CA-IDMS Presspack:

1. Unload the database using the UNLOAD utility
2. Modify the compression specifications of the schema records
3. Reload the database using the RELOAD utility

Note: When using UNLOAD and RELOAD to convert records to CA-IDMS Presspack, be sure that the DCTABLE specification in the new schema record definition includes the parameter IS USED FOR BOTH.

►► For information about the UNLOAD and RELOAD utilities and about the unload/reload process, refer to *CA-IDMS Utilities*.

To convert from fixed-length, uncompressed records to CA-IDMS Presspack without reducing the size of the area:

1. Change the DCTABLE specification in the schema records
2. Use the RESTRUCTURE utility on the area or areas containing the records

►► For information about the RESTRUCTURE utility and about the restructure process, refer to *CA-IDMS Utilities*.

To convert from variable-length, uncompressed records without reducing the size of the area:

1. Specify the DCTABLE parameter for each record, including IS USED FOR COMPRESSION.

2. Run a database sweep program, performing the OBTAIN and MODIFY functions for each record.

Note: If you perform a database sweep to convert your records, you should put normal restart and recovery procedures into effect. This will help preserve the integrity of your database if problems arise during conversion.

3. Change the DCTABLE parameter for each record, specifying that the named DCT is used for both compression and decompression.

Converting from IDMSCOMP: Your existing database records may be compressed with calls to IDMSCOMP. To convert IDMSCOMP-compressed records to CA-IDMS Presspack:

1. For each record's SCHEMA RECORD statement:
 - Specify the DCTABLE parameter, including IS USED FOR COMPRESSION.
 - Code CALL IDMSDCOM AFTER GET.
2. Run a database sweep program, performing the OBTAIN and MODIFY functions for each record.
3. Change the DCTABLE parameter for each record, specifying that the named DCT is used for both compression and decompression.

Changing the DCT: You can change the DCT that CA-IDMS Presspack uses to compress and decompress a record:

1. Specify two DCTABLE parameters in the SCHEMA RECORD statement for each record:
 - In one DCTABLE parameter, specify that the old DCT named is used for decompression.
 - In the other DCTABLE parameter, specify that the new DCT named is used for compression.
2. Run a database sweep program, performing the OBTAIN and MODIFY functions for each record.
3. Alter the DCTABLE parameter for each record, specifying that the new DCT is used for both compression and decompression.

Ways of converting from CA-IDMS Presspack: After your records have been converted to CA-IDMS Presspack use, you may occasionally want to remove selected records from CA-IDMS Presspack use. This involves converting the records from CA-IDMS Presspack compression and decompression to uncompressed format, either fixed-length or variable-length.

To convert from CA-IDMS Presspack **to variable-length, uncompressed records:**

1. Specify the DCTABLE parameter for each record, including the *dctable-name* of the record's existing DCT and IS USED FOR DECOMPRESSION.

2. Run your database sweep program, performing the OBTAIN and MODIFY functions for each record.
3. Change the SCHEMA RECORD statement for each record, specifying:
 - Appropriate values for the MINIMUM ROOT and MINIMUM FRAGMENT parameters
 - EXCLUDE ALL CALLS

To convert from CA-IDMS Presspack **to fixed-length, uncompressed records:**

1. Decompress each record and use the RESTRUCTURE utility to store each record as fixed-length without compression.
2. Change the SCHEMA RECORD statement for each record, specifying:
 - MINIMUM ROOT NULL
 - MINIMUM FRAGMENT NULL
 - EXCLUDE ALL CALLS

2.4 Converting tables

What you can do: You can convert existing SQL-defined tables to or from CA-IDMS Presspack use. The basic steps are the same for each conversion.

How to convert tables to CA-IDMS Presspack: You can specify compression for a table only in the CREATE TABLE statement. To specify compression for a table that contains data:

1. Create an interim table with same characteristics as the table to be converted but with a different name.
2. Copy the data from the table to be converted to the interim table, using an INSERT statement.

For example, if the table to be converted is DEMOEMPL.JOB and the interim table is DEMOEMPL.TMPJOB, you submit a form of this statement:

```
insert into demoempl.tmpjob
  values (select * from demoempl.job);
```

3. Delete DEMOEMPL.JOB with a DROP TABLE statement.
4. Re-create the DEMOEMPL.JOB table using the COMPRESS parameter:

```
create table demoempl.job
.
.
.
  compress using job_dct;
```

5. Copy the data from DEMOEMPL.TMPJOB into the newly-defined DEMOEMPL.JOB table using an INSERT statement:

```
insert into demoempl.job
  values (select * from demoempl.tmpjob);
```

6. After verifying successful execution of the previous step, drop the DEMOEMPL.TMPJOB table.

Tip: If you perform the conversion in a single SQL transaction, consider using the CREATE TEMPORARY TABLE statement to create the interim table.

►► For documentation of all CA-IDMS/DB SQL statements refer to *CA-IDMS SQL Reference Guide*.

How to convert tables to a new DCT: To convert tables to a new DCT, follow the steps for converting to CA-IDMS Presspack. Do not include a COMPRESS parameter when creating the interim table.

How to convert tables from CA-IDMS Presspack: To convert tables from CA-IDMS Presspack, follow the steps for converting to CA-IDMS Presspack, but omit the COMPRESS parameter when you re-create the table you are converting.

2.5 CA-IDMS Presspack runtime execution

When CA-IDMS Presspack routines are executed: After you implement CA-IDMS Presspack, data compression and decompression take place automatically when the application program performs data manipulation language (DML) functions against the CA-IDMS/DB database at runtime. The DBMS executes CA-IDMS Presspack's runtime routines to:

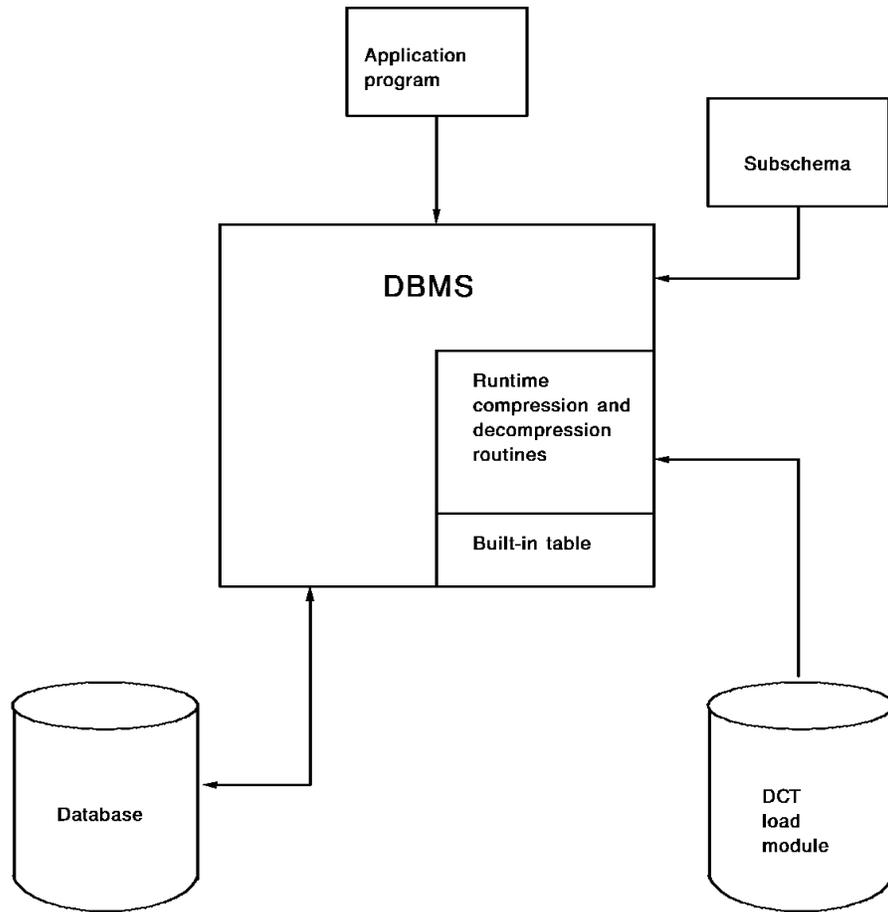
- Compress a record or table on one of these update functions:
 - STORE (navigational DML)
 - MODIFY (navigational DML)
 - ERASE (navigational DML)
 - INSERT (SQL DML)
 - UPDATE (SQL DML)
 - DELETE (SQL DML)
- Decompress a record or table on one of these retrieval functions:
 - GET (navigational DML)
 - OBTAIN (navigational DML)
 - SELECT (SQL DML)

►► For information about DML functions in application programs, refer to *CA-IDMS Navigational DML Programming* or *CA-IDMS SQL Programming Guide*.

Use of the DCT: CA-IDMS Presspack performs compression and decompression functions on record or table data using either BUILTIN or a custom DCT to translate the compressed data to decompressed data and to translate decompressed data into compressed data.

CA-IDMS Presspack must use the same DCT to decompress a given record or table as it did to compress the record or table.

Runtime processing: The figure below shows how CA-IDMS Presspack performs compression and decompression at runtime.



Chapter 3. Creating DCTs with IDMSPASS

- 3.1 About IDMSPASS 3-4
- 3.2 About this chapter 3-5
- 3.3 Running IDMSPASS 3-6
- 3.4 IDMSPASS syntax 3-8
 - 3.4.1 Usage 3-10
 - 3.4.2 Examples 3-12
- 3.5 What you can do with IDMSPASS 3-14
 - 3.5.1 Generating a statistical report 3-14
 - 3.5.2 Reading the IDMSPASS report 3-15
 - 3.5.3 Creating DCTs 3-17

3.1 About IDMSPASS

IDMSPASS is a utility provided with CA-IDMS Presspack. With IDMSPASS you can accumulate database record or table statistics, create custom DCTs, and print out a report containing compression information about your database records or tables.

3.2 About this chapter

This chapter describes IDMSPASS, including:

- Running IDMSPASS
- What you can do with IDMSPASS
- IDMSPASS input parameters
- A sample IDMSPASS report

3.3 Running IDMSPASS

IDMSPASS execution environment

- **Accessing a non-SQL defined database:**

You can run IDMSPASS against a non-SQL defined database as a batch job either under the CA-IDMS/DC central version or in local mode.

If you run IDMSPASS under the central version, the DBMS loads a subschema into the local region to make it available for extracting information. Make sure that this subschema is the same as the subschema used by your central version run unit. This will prevent discrepancies that may result from using different versions of a subschema.

- **Accessing an SQL-defined database:**

You can run IDMSPASS against an SQL-defined database in local mode. You specify the dictionary containing the SQL definitions in a SYSIDMS DBNAME= parameter.

IDMSPASS execution: At runtime, IDMSPASS samples the records or tables that you specify in IDMSPASS input parameters. To sample records, you should begin the input parameter stream with PROCESS SUBSCHEMA. To sample records, you should begin the input parameter stream with PROCESS SQL.

IDMSPASS uses statistical information in its sample to build a DCT and/or print the IDMSPASS report. You can save the statistics in a SYS002 file for use in a subsequent run of IDMSPASS.

►► For documentation of IDMSPASS input parameters, see 3.4, “IDMSPASS syntax” on page 3-8 later in this chapter.

For an example of specifying a SYS002 file in IDMSPASS JCL, see Appendix B, “Sample JCL” on page B-1.

How IDMSPASS creates a DCT: To create a DCT, IDMSPASS uses sampling statistics to create assembler source code for the DCT and punches the code into SYSPCH. You can then assemble and link this code to produce a DCT load module.

Remember: Always save a copy of the source code for each DCT load module. For more information, see 3.5.3, “Creating DCTs” on page 3-17 later in this chapter.

After you specify CA-IDMS Presspack compression using a SCHEMA RECORD or CREATE TABLE statement, the DBMS can use at application program runtime the custom DCTs that you have created with IDMSPASS.

Summary of IDMSPASS execution: The flow chart below summarizes IDMSPASS execution.

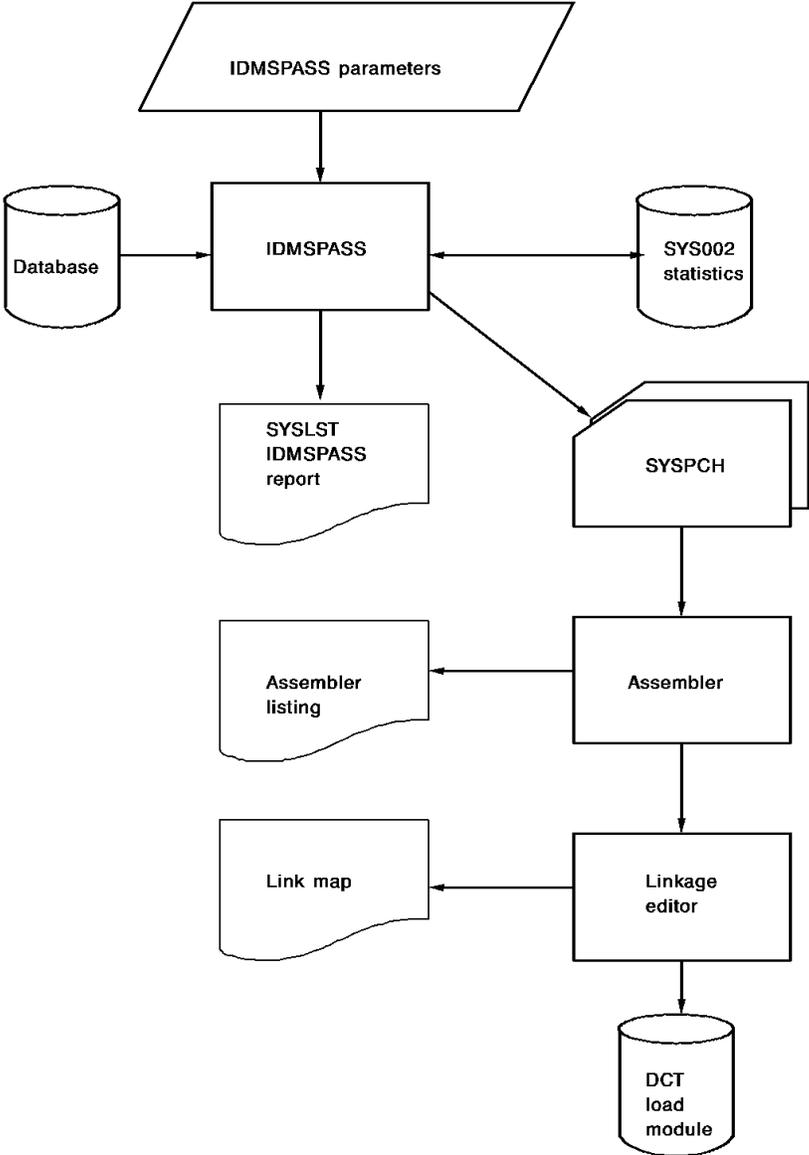


TABLE *schema-name.table-identifier*

Specifies a table for IDMPASS to sample.

Table-identifier identifies a table created with a CREATE TABLE statement.

Schema-name specifies the schema with which the table is associated.

AREA *segment-name.area-name*

Specifies an area that contains tables for IDMPASS to process.

IDMPASS will process all user-defined tables in the area.

SUBSCHEMA *subschema-name*

Names the subschema for IDMPASS to use for accessing the records specified in AREA or RECORD parameters that follow.

If you specify SUBSCHEMA, you must specify at least one AREA or RECORD parameter following it.

DBNAME *database-name*

Specifies the database to which *subschema-name* allows access.

The value in this parameter overrides the SYSIDMS DBNAME= parameter (if any).

If you do not specify DBNAME in the PROCESS statement, you must specify it in a DBNAME= SYSIDMS parameter.

►► For documentation of SYSIDMS parameters, refer to *CA-IDMS Database Administration*.

RECORD *record-name*

Specifies a record for IDMPASS to sample.

AREA *area-name*

Specifies an area that contains records for IDMPASS to process.

IDMPASS will process all user-defined records in the area.

DCT *dctable-name*

Names a data characteristic table.

Dctable-name must contain 1 to 8 alphanumeric characters. It cannot begin with the characters *BUILTIN*.

This parameter is required if you specify a value greater than 0 in the DEGREE parameter.

START *start-number*

Specifies the record occurrence or table row that IDMPASS will use to start sampling the record or table.

Start-number must be an integer between 1 and the number of record occurrences or table rows that exist when you run IDMPASS.

If you omit the START parameter, IDMPASS will start sampling with the first record occurrence or table row.

SKIP *skip-number*

Specifies the number of record occurrences or table rows that IDMPASS skips in sampling.

If you omit the SKIP parameter, IDMPASS skips no records.

NRECS *sample-number*

Specifies the maximum number of record occurrences or table rows that IDMPASS will sample in the current run.

Sample-number must be an integer.

If you specify NRECS=0, IDMPASS will not sample any record occurrences.

If you omit the NRECS parameter, IDMPASS continues sampling until it processes either all record occurrences or table rows.

DEGREE *compression-degree-number*

Specifies the degree to which CA-IDMS Presspack will compress records using the DCT created in the current run of IDMPASS.

Compression-degree-number must be an integer in the range of 0 to 8. If the value is 0, IDMPASS will not create a DCT.

If you omit the DEGREE parameter, IDMPASS assumes a default value of 4 for *compression-degree-number*. Degree 4 yields good compression and a medium-sized (13K) DCT.

USCNTS

Specifies whether IDMPASS merges statistics from the current run with statistics from the previous run, if any. Y and N are the only valid values you can enter following the USCNTS keyword.

Y specifies that statistics from the current run (if requested) will be *merged* with statistics in the SYS002 file.

N specifies that only statistics from the current run will be used and will *replace* statistics in the SYS002 file, if specified in the JCL.

If you omit the USCNTS parameter, only statistics calculated in the current run will be used.

COMPARE

Specifies that the Compression Summary of the IDMPASS report will include a column that shows the actual savings, if any, of using the BUILTIN DCT for compressing the selected records.

3.4.1 Usage

Data that can be accessed in one run: IDMPASS can access in one run any data that is defined in:

- One subschema
- The catalog component of one dictionary

You can build a DCT that samples data defined in multiple subschemas or in catalog components of multiple dictionaries, or a combination of both, with multiple runs of IDMPASS specifying USCNTS Y.

Limiting the sample: After a sampling of 64K bytes of data (not including keys), the rate at which the sample improves begins to diminish.

You can use any one or more of the NRECS, SKIP, and START parameters to limit sample size.

Considerations in sampling an area: Before specifying an AREA parameter to IDMPASS, be aware of these considerations:

- A custom DCT for data with similar physical characteristics is desirable; an area is usually defined for data with similar logical characteristics
- It is redundant to specify a RECORD parameter for a record in the same area
- Specifying two areas and limiting the sampling with an NRECS parameter may bias the sampling toward the data in first area IDMPASS processes because the NRECS limit may be reached before some (or any) of the second area is processed

Specifying NRECS, DEGREE, and USCNTS: Your specifications for NRECS, DEGREE, and USCNTS determine whether IDMPASS:

- Creates a statistical report only, or a report and a DCT
- Uses statistics from only the current sampling
- Uses statistics from only the previous sampling
- Merges statistics from the previous sampling with statistics from the current sampling

This table shows the values required for each parameter depending on what you want IDMPASS to do:

IDMPASS functions	NRECS	DEGREE	USCNTS
Report only, using statistics from currently sampled database	> 0	0	N
Report only, using statistics from previous sampling merged with current sampling	> 0	0	Y
Report and DCT, using statistics from currently sampled database	> 0	> 0	N
Report and DCT, using statistics from previous sampling merged with current sampling	> 0	> 0	Y
Report, using statistics from previous sampling (report file unchanged)	0	0	Y
Report and DCT, using statistics from previous sampling (report file unchanged)	0	> 0	Y

Invalid parameter combinations: The combination of USCNTS N and NRECS 0 is invalid because it excludes statistics from both a previous sampling (USCNTS N) and the current sampling (NRECS 0).

3.4.2 Examples

IDMPASS run against records: In this example, the input parameters direct IDMPASS to:

- Access the database using the EMPSS01 subschema
- Sample records of the record type named JOB
- Create a DCT named EMPDCT
- Start sampling with the first record (this is the default)
- Skip four record occurrences after each one sampled
- Sample no more than 1000 record occurrences
- Create a DCT that will cause CA-IDMS Presspack to compress records by degree 6
- Use only statistics calculated in the current run to build a DCT (this is the default)

```
PROCESS SUBSCHEMA EMPSS01
RECORD JOB
DCT EMPDCT
START 1
SKIP 4
NRECS 1000
DEGREE 6
USCNTS N
```

IDMSPASS run against tables: This example is comparable to the previous example except that the accessed data is in an SQL-defined database:

```
PROCESS SQL
  TABLE DEMOEMPL.JOB
  DCT EMP_DCT
  START 1
  SKIP 4
  NRECS 1000
  DEGREE 6
  USCNTS N
```

3.5 What you can do with IDMSPASS

Making decisions about compression: You can use IDMSPASS to help you make these decisions about how to use CA-IDMS Presspack:

- Which records or tables in your database to compress —
 - Using BUILTIN
 - Using a custom DCT
- How many DCTs to create
- What degree of compression to specify for each DCT

Options for an IDMSPASS run: When you run IDMSPASS, you specify at least one record or table and you request that IDMSPASS generate one or both of these:

- A statistical report on the possible degrees of compression that can be attained (you specify a value greater than 0 in the NRECS parameter)
- A DCT (you specify a value greater than 0 in the DEGREE parameter)

►► For complete documentation of IDMSPASS syntax and usage, see 3.4, “IDMSPASS syntax” on page 3-8 earlier this chapter.

3.5.1 Generating a statistical report

What can be reported: IDMSPASS allows you to generate an IDMSPASS report that provides information about the prepassed records or tables, without creating a DCT.

This information includes projections about how much your records could be compressed at each of eight degrees of compression.

You can specify two or more records or tables to pass through IDMSPASS. Statistics on the specified records or tables are merged.

Using data on degrees of compression: You can use the data on projected degrees of compression in an IDMSPASS report to:

- Select a degree of compression for your records
- Decide whether to:
 - Compress records or tables you selected
 - Use the BUILTIN DCT provided with CA-IDMS Presspack for compression
 - Use a custom DCT for compression

Contents of an IDMPASS report: An IDMPASS report, part of the IDMPASS output, contains:

- The IDMPASS input parameters that you specified for the run
- Detailed information about the areas that IDMPASS accessed during the run
- Detailed information about the records or tables that IDMPASS accessed during the run
- Run totals
- Projected percentage of record compression for each of eight degrees of compression

3.5.2 Reading the IDMPASS report

The beginning section: The beginning section:

- Echoes input parameters
- Displays a message confirming the binding of the run unit
- Displays a message for each area accessed

```
IDMPASS CAGJFO          Database Prepass and DCT generate utility          06/21/96 125424
PAGE          1

                P R E S S T O  -  C O P Y R I G H T  ( C )  1 9 8 5  B Y  I R O N W O O D  S O F T W A R E ,  I N C

SUBSCHEMA=EMPSS01,
AREA=INS-DEMO-REGION,
RECORD=DEPARTMENT,
RECORD=OFFICE,
DCT=DEMODCT,START=1,SKIP=0,DEGREE=4,USCNTS=N

UT000018 RUN UNIT HAS BEEN BOUND
SUBSCHEMA NAME ----- EMPSS01
COMPILE DATE ----- 96-05-15
COMPILE TIME ----- 08.54.12
SCHEMA VERSION ----- 1100
SUBSCHEMA VERSION --- 1200

UT002017 Starting Scan - Area EMPDEMO.INS-DEMO-REGION
UT002017 Starting Scan - Area EMPDEMO.ORG-DEMO-REGION
```

►► For documentation of all messages issued by IDMPASS, refer to *CA-IDMS Messages and Codes*.

Summary of the records or tables sampled: The summary of records or tables sampled consists of this information:

- Record/Table NAME — Name of the record type or table sampled
- F/V — Record or table type (fixed-length/variable-length)
- Avg. Length — Average length of record occurrences or table rows, including noncompressible portion
- Number Sampled — Number of record occurrences or rows sampled
- Percent Compressible — The percentage of a record occurrence or table row that is eligible for compression
- Number of Bytes Compressible — The actual number of bytes of a record occurrence or table row that can be compressed

3.5 What you can do with IDMSPASS

Summary of Records Sampled					
Record Name	F/V	Avg. Length	Number Sampled	Percent Compressible	Number of Bytes Compressible
OFFICE	F	100	5	73.0%	73
NON-HOSP-CLAIM	V	824	2	30.0%	248
INSURANCE-PLAN	F	148	4	87.1%	129
HOSPITAL-CLAIM	F	308	2	94.8%	292
DEPARTMENT	F	80	9	65.0%	52
DENTAL-CLAIM	V	824	2	15.7%	130
COVERAGE	F	44	74	36.3%	16

Total Bytes Sampled 4,265

The Compression Summary: The Compression Summary *estimates* the average length of the record occurrence or table row after compression and the percentage of the original length that this figure represents. This information is provided for each of the eight possible degrees of compression you can choose.

If you specify the COMPARE parameter to IDMSPASS, the Compression Summary also gives the *actual* savings, if any, of using the BUILTIN DCT for compression.

►► For more information about IDMSPASS syntax, see 3.4, “IDMSPASS syntax” on page 3-8.

'NO COMP' indicates that no gain would result from compression because the reduction would be less than the addition of 8 bytes of overhead associated with variable-length records.

Compression Summary														
Record Name	Average Length After Compression and Percentage of Original Size by Degree													
	1	2	3	4	5	6	7	8						
OFFICE	72	72%	68	68%	64	64%	64	64%	64	64%	64	64%	64	64%
NON-HOSP-CLAIM	136	16%	124	15%	116	14%	116	14%	116	14%	116	14%	116	14%
INSURANCE-PLAN	88	59%	80	54%	76	51%	76	51%	76	51%	76	51%	76	51%
HOSPITAL-CLAIM	156	50%	140	45%	132	42%	132	42%	132	42%	132	42%	132	42%
DEPARTMENT	64	80%	60	75%	60	75%	60	75%	60	75%	60	75%	60	75%
DENTAL-CLAIM	84	10%	76	9%	76	9%	72	8%	76	9%	76	9%	72	8%
COVERAGE	NO COMP.		NO COMP.		NO COMP.		NO COMP.		NO COMP.		NO COMP.		NO COMP.	

You chose Degree 4

SQL example: The IDMSPASS report on an SQL-defined database is similar to a report on a non-SQL defined database, as in this example:

```

IDMSPASS CAGJF0          Database Prepass and DCT generate utility          06/21/96 130045
PAGE          1

                                P R E S S T O  -  COPYRIGHT (C) 1985 BY IRONWOOD SOFTWARE, INC

PROCESS SQL
AREA SQLDEMO.EMPLAREA
DCT=DEMOCT,START=1,SKIP=0,DEGREE=4,USCNTS=N,COMPARE

UT00018 RUN UNIT HAS BEEN BOUND
      SUBSCHEMA NAME ----- PASSSS01
      COMPILE DATE ----- 96-06-21
      COMPILE TIME ----- 13.00.50
      SCHEMA VERSION ----- 1100
      SUBSCHEMA VERSION --- 1200

UT002017 Starting Scan - Area SQLDEMO.EMPLAREA

                                Summary of Tables Sampled

Table Name      F/V      Avg.      Number      Percent      Number of Bytes
                F/V      Length     Sampled     Compressible Compressible
-----
BENEFITS        F      132        200         78.7%         104
COVERAGE        F      48         71          41.6%          20

Total Bytes Sampled 22,220

                                Compression Summary

Table Name      1      2      3      4      5      6      7      8      BUILTIN
-----
BENEFITS        84 63%  80 60%  80 60%  76 57%  76 57%  76 57%  72 54%  72 54%  108 81%
COVERAGE        NO COMP.  NO COMP.

You chose Degree 4
    
```

3.5.3 Creating DCTs

Selecting a DCT for compression: A DCT is a conversion table built by CA-IDMS Presspack on the basis of the data sampled by IDMSPASS. It is used at runtime to perform compression and decompression functions.

Part of the decision to compress a record or table is to select a DCT for CA-IDMS Presspack to use. Your choices are:

- Use the generic DCT (BUILTIN) provided with CA-IDMS Presspack
- Create a custom DCT for the selected records or tables

How to create a DCT: To create a DCT, you specify these IDMSPASS parameters:

- For **tables**:
 - The keyword SQL
 - TABLE *table-name* for each selected table (or AREA *segment-name.area-name* for all tables in each selected area)
- For **records**:
 - SUBSCHEMA *subschema-name*
 - RECORD *record-name* for each selected record (or AREA *area-name* for all records in each selected area)
- DCT, to name the DCT
- DEGREE, with a value greater than 0, to specify the degree of compression

With additional optional parameters, you can make specifications about how IDMSPASS takes the sampling that it uses to create the DCT.

►► For complete documentation of IDMSPASS syntax and usage, see 3.4, “IDMSPASS syntax” on page 3-8 earlier in this chapter.

Effect of degree: You specify an integer in the range 1 through 8 in the DEGREE parameter to create a DCT. DEGREE determines the number of successor tables used in the sampling process. The higher the degree value, the more finely tuned the DCT is to the data sampled.

The size (in bytes) of the DCT that IDMSPASS creates will be 3.3K multiplied by the degree that you specify. In other words, the higher the degree of compression that you select, the larger the DCT created by IDMSPASS.

A higher degree of compression does not mean compression routines do more work; it means that they have more successor tables to work with. There is no extra processing.

The higher the degree of compression, the greater the compression efficiency, but on a diminishing scale. Thus, compression efficiency increases the most from 1 to 2, less from 2 to 3, and so on. It increases the least from 7 to 8.

Creating custom DCTs: If you specify one record or table when you run IDMSPASS, IDMSPASS creates a DCT for just that record or table. If you pass two records or tables when you run IDMSPASS, IDMSPASS creates a DCT for both. In other words, IDMSPASS creates one DCT for just what you select to pass in an IDMSPASS run.

The more specific that a DCT is to a record or table, the more effective the compression. One DCT for each record or table achieves the most effective compression for the record or table.

Deciding how many DCTs: To decide how many DCTs to create, balance the high degree of efficiency obtained from using many custom DCTs with the storage and loading requirements of many DCTs. Also consider whether the record or table is worth the effort to compress — a few hundred record occurrences or table rows probably are not worth the effort.

A rule of thumb is to create one DCT for all records that contain similar data (for example, one DCT for records or tables with text data, one for predominantly packed data, one for mixed data, and so on).

Saving DCTs: Once you compress a record with a DCT, you can use *only* that DCT to decompress the record. Therefore, be sure that the source code module and the load module for each DCT are backed up in secure fashion to safeguard against losing access to compressed data.

Using saved source code or a backup copy of the load module is the only way to recreate a DCT because IDMSPASS does not create a duplicate DCT unless you use it with exactly the same sampling.

Generating a DCT load module: You generate the DCT load module by assembling the DCT source code that IDMSPASS creates and writes to the SYSPCH file.

►► For information about incorporating DCT load module generation in the JCL stream for IDMSPASS, see Appendix B, “Sample JCL” on page B-1.

Appendix A. Reading Compressed Data From the Journal

How database changes are journaled: Before and after images of compressed data are journaled in compressed format. Therefore, journaled data must be decompressed to be readable.

For example, a program (such as an audit program) that reads compressed data in the journal must be able to decompress the data.

Calling IDMSUNPS: To decompress journal images of records that were compressed with CA-IDMS Presspack, a program must call IDMSUNPS.

Note: IDMSUNPS mimics a database procedure, but it is not a database procedure and cannot be called as such.

Prerequisite for a call to IDMSUNPS: To use IDMSUNPS, the user-written program must read the journal, deblock individual journal records, and piece together any spanned journal images. IDMSUNPS should be called only when the program has identified a record to be decompressed.

Call procedure: The program that needs the journal image decompressed calls UNPSEP1, the entry point to IDMSUNPS, using standard calling IBM conventions.

The IDMSUNPS module requires five parameters to be passed:

Parameter	Usage	Length	Description
Procedure control block	Alphanumeric	1 byte	(Filler — unused)
Application control block	Alphanumeric	1 byte	(Filler — unused)
Application program information block	Alphanumeric	8 bytes	Name of the DCT to be used for compression ¹
Record control block	Alphanumeric	40 bytes	(Filler — unused)
	Binary	2 bytes	Control length — length of the data portion of the record (including the RDW field) up to and including the last CALC or sort-control field ¹
	Binary	2 bytes	Maximum length of the variable length record (decompressed size) including the RDW ¹
Record occurrence block	Record occurrence (compressed and decompressed — see the discussion below)		

¹ This information can be found in the dictionary or a subschema, or can be coded in the calling program.

The Record Occurrence block: Be aware of these considerations for the Record Occurrence block:

- The program passes the compressed record in this block, and IDMSUNPS returns the decompressed record to it. Therefore, the area must be large enough to hold the decompressed record
- A prefix-only journal record should not be passed
- The information passed in this block must be the portion of the compressed journal record that begins with the record's RDW.

This example shows where the starting point is in a typical journal record:

RDW of journal record
Journal type (BFOR/AFTR)
Journal sequence number
Run unit ID
Area type
Verb number
Page group
Dbkey format word
Dbkey
Record ID
Displacement on page
Record length
Prefix length
Displacement in DBMS of journal call
Offset
Prefix points
Start of record for IDMSUNPS → RDW of data record
Data of data record

IDMSUNPS return codes: IDMSUNPS may return to the calling program these values (to Register 15 for an Assembler program or RETURN-CODE for a COBOL program):

- 00 - Decompression successful
- 04 - GETSTG failure
- 08 - Call from system mode module; probable cause is
trying to call IDMSUNPS as a database procedure
- 12 - DCT load failure
- 16 - DCT not valid
- 20 - Record or table not compressed by CA-IDMS Presspack
- 24 - Load of IDMPRES failed
- >100 - Error during decompression
(most likely, the wrong DCT was specified);
PRESSTO return code = return code minus 100

Appendix B. Sample JCL

- B.1 About this appendix B-4
- B.2 OS/390 JCL B-5
 - B.2.1 Usage B-6
- B.3 VSE/ESA JCL B-8
 - B.3.1 Usage B-9
- B.4 VM/ESA commands B-10
 - B.4.1 Usage B-11
- B.5 BS2000/OSD JCL B-12
 - B.5.1 Usage B-13

B.1 About this appendix

In this appendix you will find sample JCL or commands for executing IDMSPASS on four operating systems: OS/390, VSE/ESA, VM/ESA, and BS2000/OSD.

B.2 OS/390 JCL

IDMSPASS (OS/390)

```
//PASS      EXEC PGM=IDMSPASS,REGION=1024K
//STEPLIB  DD DSN=idms.dba.loadlib,DISP=SHR
//          DD DSN=idms.loadlib,DISP=SHR
//userdb   DD DSN=user.userdb,DISP=SHR
//dcmmsg   DD DSN=idms.sysmsg.ddldcmmsg,DISP=SHR
//sysjrn1  DD DUMMY
//SYS002   DD DSN=idms.statfile,DISP=(NEW,CATLG,DELETE),
//          UNIT=disk,VOL=SER=vvvvvy,SPACE=(TRK,(10,5)),
//          DCB=(RECFM=FB,LRECL=1032,BLKSIZE=bbbbbb)
//SYSPCH   DD DSN=&&dctsrc,DISP=(NEW,PASS),
//          UNIT=disk,SPACE=(TRK,(10,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSLST   DD SYSOUT=A
//SYSIDMS  DD *
DMCL=dmcl-name
Additional SYSIDMS parameters, as appropriate
/*
//SYSIPT   DD *
IDMSPASS PROCESS statement
/*
//*
//ASM      EXEC PGM=assembler,REGION=1024K,PARM='DECK,RENT'
//SYSPRINT DD SYSOUT=A
//SYSLIB   DD DSN=sys1.maclib,DISP=SHR
//          DD DSN=idms.maclib,DISP=SHR
//SYSUT1   DD UNIT=disk,SPACE=(CYL,(3,2))
//SYSUT2   DD UNIT=disk,SPACE=(CYL,(3,2))
//SYSUT3   DD UNIT=disk,SPACE=(CYL,(3,2))
//SYSPUNCH DD DSN=&&dctobj,DISP=(NEW,PASS),
//          UNIT=disk,SPACE=(TRK,(10,5)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSIN    DD DSN=&&dctsrc,DISP=(OLD,DELETE)
//*
//LINK     EXEC PGM=IEWL,REGION=512K,PARM='LET,LIST,NCAL,RENT,XREF'
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD UNIT=disk,SPACE=(CYL,(3,2))
//SYSLMOD  DD DSN=user.loadlib,DISP=SHR
//SYSLIN   DD DSN=&&dctobj,DISP=(OLD,DELETE)
//          DD *
ENTRY dctable-name
NAME dctable-name(R)
/*
//*
```

<u>assembler</u>	Name of the assembler program
<u>bbbb</u>	Block size of the statistics file; must be a multiple of 1032
<u>dcmsg</u>	DDname of the system message (DDLDCMSG) area
<u>dctable-name</u>	Name of the DCT
<u>dmcl-name</u>	Name of the DMCL
<u>dictionary-name</u>	Dictionary containing SQL catalog component
<u>disk</u>	Symbolic device name for work files
<u>idms.dba.loadlib</u>	Data set name of the load library containing the DMCL and database name table load modules
<u>idms.loadlib</u>	Data set name of the load library containing the CA-IDMS executable modules
<u>idms.maclib</u>	Data set name of the CA-IDMS macro library
<u>idms.srclib</u>	Library containing member #CRQID
<u>idms.statfile</u>	File containing statistics gathered by IDMSPASS on sampled records
<u>idms.sysmsg.ddldcmsg</u>	Data set name of the system message (DDLDCMSG) area
<u>sysjrn1</u>	DDname of the dummy journal file
<u>sys1.maclib</u>	Data set name of the operating system macro library
<u>user.loadlib</u>	Data set name of the load library containing the user-defined executable modules
<u>userdb</u>	DDname of the user database file
<u>user.userdb</u>	Data set name of the user database file
<u>vvvvv</u>	Volume serial identifier
<u>&&dctobj</u>	Data set name of the temporary file containing the assembler output
<u>&&dctsrc</u>	Data set name of the temporary file containing the assembler source that becomes the DCT

B.2.1 Usage

Loading the subschema from the load area: If the subschema named in the PROCESS statement resides in a load area, include this card to identify the ddname and data set name of the DDLDCLOD area the application dictionary:

```
//dclod DD DSN=idms.system.ddldclod,DISP=SHR
```

Accessing tables: To access tables, identify the ddnames and data set names of the DDLCAT and DDLCATX areas of the application dictionary where the table definitions are stored:

```
//dcat DD DSN=idms.syssql.ddlcat,DISP=SHR  
//dcatx DD DSN=idms.syssql.ddlcatx,DISP=SHR
```

Also specify the name of the application dictionary in a SYSIDMS parm (you do not need to specify the database name):

```
DBNAME=dictionary-name
```

Making the DCT available to CA-IDMS Presspack: If you store the executable DCT load module in *user.loadlib* as shown in the sample, *user.loadlib* must be defined in the startup JCL for the system for CA-IDMS Presspack to access it.

Merging statistics: If you use multiple runs to merge statistics, change the DISP= specification for *idms.statfile* in the sample for each subsequent run.

Reassembling CA-IDMS modules: Any modifications to the CA-IDMS load libraries should be applied by SMP/E for OS/390. For instructions on how to assemble and link-edit a module using SMP/E, refer to "System Modification Procedures" in *CA-IDMS Installation and Maintenance Guide - OS/390*.

B.3 VSE/ESA JCL

IDMSPASS (VSE/ESA)

```
// EXEC PROC=IDMSLBLS
// EXTENT SYSCLB,nnnnnn,,ssss,1111
// ASSGN SYSCLB,DISK,VOL=nnnnnn,SHR
// TLBL V002,'idms.statfile',,nnnnnn,,f
// ASSGN SYS002,TAPE,VOL=nnnnnn
// DLBL IDMSPCH,'temp.dct',0
// EXTENT sys020,nnnnnn,,ssss,1111
// ASSGN sys020,DISK,VOL=nnnnnn,SHR
// DLBL userdb,'user.userdb',,DA
// EXTENT sys010,nnnnnn
// ASSGN sys010,DISK,VOL=nnnnnn,SHR
Additional database file assignments, as required
// ASSGN SYS009,IGN
// EXEC IDMSPASS
DMCL=dmcl-name
DBNAME=dictionary-name
Additional SYSIDMS parameters, as appropriate
/*
IDMSPASS PROCESS statement
/*
// DLBL IJSYSIN,'temp.dct'
// EXTENT SYSIPT,nnnnnn
// ASSGN SYSIPT,DISK,VOL=nnnnnn,SHR
// OPTION CATAL,NODECK,NOSYM
// PHASE dct,*
// EXEC ASSEMBLY
// EXEC LNKEDT
```

<u>dct</u>	Name of the IDMSPASS module; must be the same as DEVICE-MEDIA NAME in source code
<u>dmcl-name</u>	DMCL name
<u>dictionary-name</u>	Dictionary containing SQL catalog component
<u>f</u>	File number of the statistics file
<u>idms.statfile</u>	File ID of file that contains statistics gathered by IDMSPASS; this file can go to either disk or tape
<u>l111</u>	Number of tracks (CKD) or blocks (FBA) of disk extent
<u>nnnnnn</u>	Volume serial number
<u>ssss</u>	Starting track (CKD) or block (FBA) of disk extent
<u>sys009</u>	Logical unit assignment of the dummy journal file
<u>sys010</u>	Logical unit assignment of the user database file
<u>sys020</u>	Logical unit assignment of the punched output
<u>temp.dct</u>	File ID of the temporary data set containing the assembler source that becomes the DCT

B.3.1 Usage

Reassembling CA-IDMS modules: Any modifications to the CA-IDMS load libraries should be applied by MSHP for VSE/ESA. For instructions on how to assemble and link-edit a module using MSHP, refer to "System Modification Procedures" in *CA-IDMS Installation and Maintenance Guide - VSE/ESA*.

B.4 VM/ESA commands

IDMSPASS (VM/ESA)

```

*
*   IDMSPASS STEP
*
FILEDEF SYSPCH DISK dct ASSEMBLE a (RECFM FB LRECL bbbb2
FILEDEF SYS002 DISK idmpass stats a (RECFM FB LRECL 1032 BLKSIZE bbbb1
FILEDEF userdb DISK user userdb a (RECFM FB LRECL ppp BLKSIZE ppp
XTENT nnn
Additional database file assignments, as required
FILEDEF SYSIPT DISK idmpass input a (RECFM F LRECL lll BLKSIZE bbb
FILEDEF SYSIDMS DISK sysidms parms a (RECFM F LRECL lll BLKSIZE bbb
EXEC IDMSFD
OSRUN IDMSPASS
*
*   ASSEMBLE STEP
*
FILEDEF TEXT DISK dct TEXT A
EXEC IDMSFD
ASSEMBLE dct (NODECK OBJECT PRINT NOTERM
*
*   LINK EDIT STEP
*
EXEC IDMSFD
LKED dct (LET LIST XREF RENT

```

<u>bbb</u>	Block size of the user input data file
<u>bbbb1</u>	Must be a multiple of 1032
<u>bbbb2</u>	Must be a multiple of 80
<u>dct</u>	File name of the data characteristic table
<u>dct</u> ASSEMBLE <u>a</u>	File ID of the file containing the assembler source that becomes a DCT
<u>idmpass input a</u>	File ID of the file containing the IDMSPASS input parameters
<u>idmpass stats a</u>	File ID of the file containing statistics gathered by IDMSPASS on sampled records
<u>lll</u>	Logical record length of the user input data file
<u>nnn</u>	Number of pages in the user database file
<u>ppp</u>	Page size of the user database file
<u>sysidms parms a</u>	File ID of the file containing SYSIDMS parameters
<u>userdb</u>	DDname of the user database file
<u>user userdb a</u>	File ID of the user database file

B.4.1 Usage

Local mode: To specify that IDMPASS is executing in local mode, perform one of the following:

- Link IDMPASS with an IDMSOPTI program that specifies local execution mode
- Specify *LOCAL* as the first input parameter of the filename, type and mode identified by idmpass input a in the IDMSFD exec.
- Modify the OSRUN statement:

```
OSRUN IDMPASS PARM='*LOCAL*'
```

Note: This option is valid only if the OSRUN command is issued from a System Product interpreter or an EXEC2 file.

SYSIPT file: To create the SYSIPT file of IDMPASS input parameters:

1. On the VM/ESA command line, type **XEDIT sysipt data a (NOPROF** and press [Enter]
2. On the XEDIT command line, type **INPUT** and press [Enter]
3. In input mode, type in the IDMPASS input parameters
4. Press [Enter] to exit input mode
5. On the XEDIT command line, type **FILE** and press [Enter]

SYSIDMS file: To run IDMPASS, you should include these SYSIDMS parameters:

- **DMCL=dmcl-name**, to identify the DMCL
- If you are running IDMPASS against an SQL-defined database, **DBNAME=dictionary-name**, to identify the dictionary whose catalog component contains the database definitions

To create the SYSIDMS file of SYSIDMS parameters:

1. On the VM/ESA command line, type **XEDIT sysidms data a (NOPROF** and press [Enter]
2. On the XEDIT command line, type **INPUT** and press [Enter]
3. In input mode, type in the SYSIDMS parameters
4. Press [Enter] to exit input mode
5. On the XEDIT command line, type **FILE** and press [Enter]

Note: For documentation of SYSIDMS parameters, refer to *CA-IDMS Database Administration*.

B.5 BS2000/OSD JCL

IDMSPASS (BS2000/OSD)

```

/ADD-FILE-LINK L-NAME=CDMSLIB,F-NAME=idms.dba.loadlib
/ADD-FILE-LINK L-NAME=CDMSLIB1,F-NAME=idms.loadlib
/ADD-FILE-LINK L-NAME=CDMSLODR,F-NAME=idms.loadlib
/CREATE-FILE F-NAME=idms.statfile, (SPACE=(primary,secondary)),
/  SUPPRESS-ERR=*FILE-EXIST
/ADD-FILE-LINK L-NAME=SYS002,F-NAME=idms.statfile
/ADD-FILE-LINK L-NAME=userdb,F-NAME=user.userdb,SHARED-UPD=*YES
/ADD-FILE-LINK L-NAME=jljrn1,F-NAME=*DUMMY
/ADD-FILE-LINK L-NAME=sysmsg,F-NAME=idms.sysmsg.ddldcmsg,SHARED-UPD=*YES
/ADD-FILE-LINK L-NAME=SYSIDMS,F-NAME=idms.sysidms
/ASSIGN-SYSOPT TO=temp.dct
/ASSIGN-SYSOPT TO=*SYSCMD
/START-PROG *MOD(ELEM=IDMSPASS,LIB=idms.loadlib,RUN-MODE=*ADV)
Input IDMSPASS parameters
/ASSIGN-SYSOPT TO=*PRIMARY
/START-ASSEMBH
//COMPILE SOURCE=temp.dct
//      ,COMP-ACT=MOD-GEN(MODULE-FORMAT=LLM)
//      ,MOD-LIB=idms.dba.loadlib(ELEM=dct)
//      ,COMPILER-TERMINATION=(MAX-ERROR-NUMBER=0)
//END
/DELETE-FILE F-NAME=temp.dct

```

<u>dcmsg</u>	Linkname of the system message (DDLDCMSG) area
<u>dct</u>	Name of the DCT object and load module
<u>dictionary-name</u>	Dictionary containing SQL catalog component
<u>dmc1-name</u>	Name of the DMCL
<u>idms.dba.loadlib</u>	File name of user load library
<u>idms.loadlib</u>	File name of CA-IDMS load library
<u>idms.jljrn1</u>	Linkname of the tape journal file
<u>idms.statfile</u>	File name of the file containing statistics gathered by IDMSPASS on sample records
<u>idms.sysidms</u>	File name of the file containing SYSIDMS parameters
<u>idms.sysmsg.ddldcmsg</u>	File name of the system message (DDLDCMSG) area
<u>primary,secondary</u>	Space allocation for the statistics file
<u>temp.dct</u>	File name of temporary work file
<u>userdb</u>	Linkname of the user database file
<u>user.userdb</u>	File name of the user database file

B.5.1 Usage

SYSIDMS DMCL= parameter: In the *idms.sysidms* file, include a SYSIDMS parameter to identify the DMCL:

DMCL=dmc1-name

Loading the subschema from the load area: If the subschema named in the PROCESS statement resides in a load area, include this card to identify the linkname and file name of the DDLDCLOD area the application dictionary:

/FILE LINK=dc1od,idms.system.dd1dc1od,SHARUPD=YES

Accessing tables: To access tables, identify the ddnames and data set names of the DDLCAT and DDLCATX areas of the application dictionary where the table definitions are stored:

/FILE LINK=dccat,idms.syssql.dd1cat,SHARUPD=YES

/FILE LINK=dccatx,idms.syssql.dd1catx,SHARUPD=YES

Also specify the name of the application dictionary in a SYSIDMS parameter (you do not need to specify the database name):

DBNAME=dictionary-name

Index

A

application programs
performing DML functions 2-12

B

BS2000/OSD JCL B-12
BUILTIN DCT
and compression decision-making 3-14
contents of 1-9

C

CA-IDMS Presspack
See also BUILTIN DCT
See also IDMSPASS utility
benefits of use 1-5
BUILTIN DCT 1-5
definition 1-5
IDMSPASS utility 1-5
runtime routines 1-5
when not to use 1-7
when to use 1-7
CA-IDMS/DC 3-6
See also IDMSPASS utility, execution environment
central version
See CA-IDMS/DC
coding considerations
for COMPRESS parameter 2-7
for DCTABLE parameter 2-5—2-6
COMPRESS parameter
See also CREATE TABLE statement
in table conversion 2-11
CREATE TABLE statement
coding the CREATE TABLE statement 2-5
using the COMPRESS parameter 2-7
custom DCT
and compression decision-making 3-14
definition 1-9
optional step in implementing CA-IDMS
Presspack 1-8
saving DCT load modules 3-18
saving DCT source code 3-18
using IDMSPASS to create 3-18

D

data characteristic table (DCT)
See BUILTIN DCT
See custom DCT
data compression
See also degrees of compression
compressing binary data 1-7
compressing packed decimal data 1-7
data conversion
See record conversion
data manipulation language (DML) functions
coding the MODIFY function as part of a database
sweep 2-8
coding the OBTAIN function as part of a database
sweep 2-8
DCT
See data characteristic table (DCT)
DCTABLE parameter
See SCHEMA RECORD statement
degrees of compression
See also IDMSPASS report
and compression efficiency 3-18
and the DEGREE parameter 3-18
relation to custom DCT size 3-18
used for compression decision-making 3-14
using IDMSPASS to project 3-14
DML functions
See data manipulation language (DML) functions
See record conversion

G

generic DCT
See also BUILTIN DCT
optimizing compression with 1-9
using 1-9

I

IDMSCOMP compression utility 2-8
IDMSDCOM decompression utility 2-8
IDMSPASS input parameters
See also syntax
AREA parameter 3-9
DCT parameter 3-9
DEGREE parameter 3-10
NRECS parameter 3-10
RECORD parameter 3-9

IDMSPASS input parameters (*continued*)

- SKIP parameter 3-10
- START parameter 3-9
- SUBSCHEMA parameter 3-9
- USCNTS parameter 3-10

IDMSPASS JCL

- BS2000/OSD JCL B-12
- OS/390 JCL B-5
- VM/ESA commands B-10
- VSE/ESA JCL B-8

IDMSPASS report

- See also* IDMSPASS input parameters
- contents of 3-14
- details of 3-15

IDMSPASS syntax

- Usage notes 3-10

IDMSPASS utility

- See also* IDMSPASS input parameters
- See also* IDMSPASS JCL
- See also* IDMSPASS report
- compressing with IDMSPASS 1-8
- creating custom DCTs 1-9, 3-18
- execution environment 3-6
- how IDMSPASS works 1-8
- optimizing data compression 1-9
- output 3-6
- running IDMSPASS 3-6
- summary of uses 3-4
- used to accumulate record statistics 3-14
- used to optimize compression 1-6

IDMSUNPS

- calling A-3
- parameters A-3
- return codes A-5

J

- journal records, decompressing A-3

L

local mode

- See* IDMSPASS utility, execution environment

M

MINIMUM FRAGMENT LENGTH parameter

- See also* SCHEMA RECORD statement
- default with DCTABLE parameter 2-6

MINIMUM ROOT LENGTH parameter

- See also* SCHEMA RECORD statement
- default with DCTABLE parameter 2-6

O

- OS/390 JCL B-5

P

PROCESS statement

- See also* IDMSPASS input parameters
- function at runtime 3-8
- syntax for IDMSPASS input parameters 3-8

R

record conversion

- converting from CA-IDMS Presspack 2-9
- converting to CA-IDMS Presspack use 2-8, 2-11
- RELOAD utility 2-8
- types of 2-8
- UNLOAD utility 2-8
- using database sweeps 2-8
- using unload/reload 2-8

RELOAD utility

- See* record conversion

restructure

- converting from fixed-length, uncompressed format 2-8
- converting to fixed-length, uncompressed format 2-8
- definition 2-8

RESTRUCTURE utility 2-8

- See also* record conversion
- See also* restructure

S

SCHEMA RECORD statement

- coding the SCHEMA RECORD statement 2-5
- MINIMUM FRAGMENT LENGTH parameter 2-6
- MINIMUM ROOT LENGTH parameter 2-6
- specifying the DCTABLE parameter 2-5

SQL

- compressing tables 2-5
- converting tables to Presspack 2-11
- database, accessing with IDMSPASS 3-6
- IDMSPASS input parameter 3-8
- IDMSPASS report output 3-16

syntax

- for IDMSPASS input parameters 3-7—3-8

syntax rules

- for IDMSPASS input parameters 3-8—3-10

T

table conversion

from Presspack 2-11

to a new DCT 2-11

to CA-IDMS Presspack 2-11

U

UNLOAD utility

See record conversion

V

VM/ESA commands B-10

VSE/ESA JCL B-8

