CA IDMS™ Presspack

User Guide

Release 18.5.00

CA Technologies Product References

This document seeences the following CA product:

- CA IDMS™ /DC

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Contents

Chapter 1: Introduction

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

Chapter 2: Overview of CA IDMS Presspack

This section contains the following topics:

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 sees you to other chapters of this manual for details.

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 database 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. This component 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 Using IDMSPASS (see page 14).

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 Using IDMSPASS (see page 14).

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, see to the CA IDMS installation guide for your operating system.
2. Define or convert selected records or tables for compression and decompression.

   For more information, see Chapter 2, “Overview of CA IDMS Presspack”.

Optional Steps

To maximize the advantages gained 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 "Generating a Statistical Report" in Chapter 4.
- 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 specify to use the generic DCT, BUILTIN, when compressing and decompressing records or tables. However, a custom DCT ensures more efficient compression than the BUILTIN DCT. For information about how to create a DCT, see "Creating DCTs" in Chapter 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, the custom DCT is used to compress or decompress the data in your database. Data compression and decompression are far more efficient with a custom DCT than with the generic BUILTIN DCT because the data characteristics used are specific to the data being compressed.

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”.

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 3: Enabling CA IDMS Presspack Compression

This section contains the following topics:

About This Chapter

This chapter tells you how to define database records or tables for compression and decompression, including:

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

Defining Records and Tables for Compression

Defining Schema Records for Compression

You define schema records for 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 more information about the SCHEMA RECORD statement and the DCTABLE parameter, see "Using SCHEMA RECORD Syntax."

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. Following the COMPRESS parameter, you can specify in the USING parameter the data characteristic table to be used for compression and decompression.

For more information about the CREATE TABLE statement and its compression parameters, see "Using CREATE TABLE Syntax."

Converting Records and Tables

To specify compression for records or tables that have already been defined in a CA IDMS™ /DB, 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 "Converting Schema Records."

For information about how to convert SQL-defined tables, see “Converting Tables”.

Using SCHEMA RECORD Syntax

Specifying the DCTABLE Parameter

You specify the DCTABLE parameter of the SCHEMA RECORD statement to compress and decompress the record at runtime.

For more information about complete syntax documentation of the SCHEMA RECORD statement, see to *CA IDMS Database Administration Guide*.

These considerations apply to the DCTABLE parameter:

- Specify *dctable-name* if you want to use a custom DCT for the schema record, or BUILTIN if you want to use the 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 IDDPRST 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 more information about converting records, see Converting Schema Records.

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.

Using CREATE TABLE Syntax

Using the COMPRESS Parameter

You specify the COMPRESS parameter of the CREATE TABLE statement to compress and decompress a table at runtime.

More Information: For complete documentation of CREATE TABLE statement syntax, see to *CA IDMS SQL 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, BUILTIN is used 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 more information about table conversion, see 23 (see page 21).

Specifying a DCT

Specify USING *dctable-name* in conjunction with the COMPRESS parameter 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;
```

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 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 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 more information about the UNLOAD and RELOAD utilities and about the unload/reload process, see to *CA IDMS Utilities Guide*.

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 more information about the RESTRUCTURE utility and about the restructure process, see to *CA IDMS Utilities Guide*.

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 is used to compress and decompress a record using the following steps:

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

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.

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

More Information: For documentation of all CA IDMS™ SQL statements see to *CA IDMS SQL seeence 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.

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 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 more information about DML functions in application programs, see to *CA IDMS Navigational DML Programming Guide* or *CA IDMS SQL Programming Guide*.

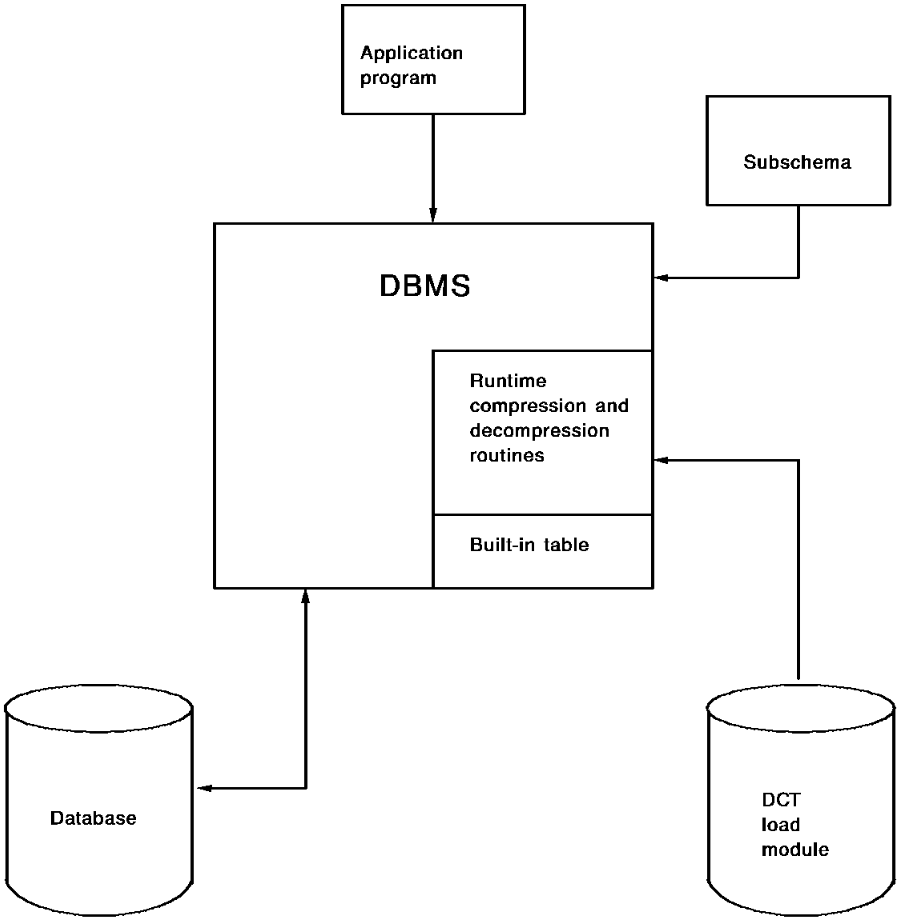
Use of the DCT

Compression and decompression functions are performed 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.

The same DCT must be used to decompress a given record or table as it did to compress the record or table.

Runtime Processing

The following figure shows how compression and decompression are performed at runtime.



Chapter 4: Creating DCTs with IDMSPASS

This section contains the following topics:

About This Chapter

This chapter describes IDMSPASS, including:

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

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.

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 must begin the input parameter stream with PROCESS SUBSCHEMA. To sample tables, you must 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 IDMSPASS Syntax.

For an example of specifying a SYS002 file in IDMSPASS JCL, see Sample JCL.

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.

You can modify the generated assembler source code to set AMODE=31 and RMODE=ANY. This eliminates warnings during the linkedit about mismatched AMODEs.

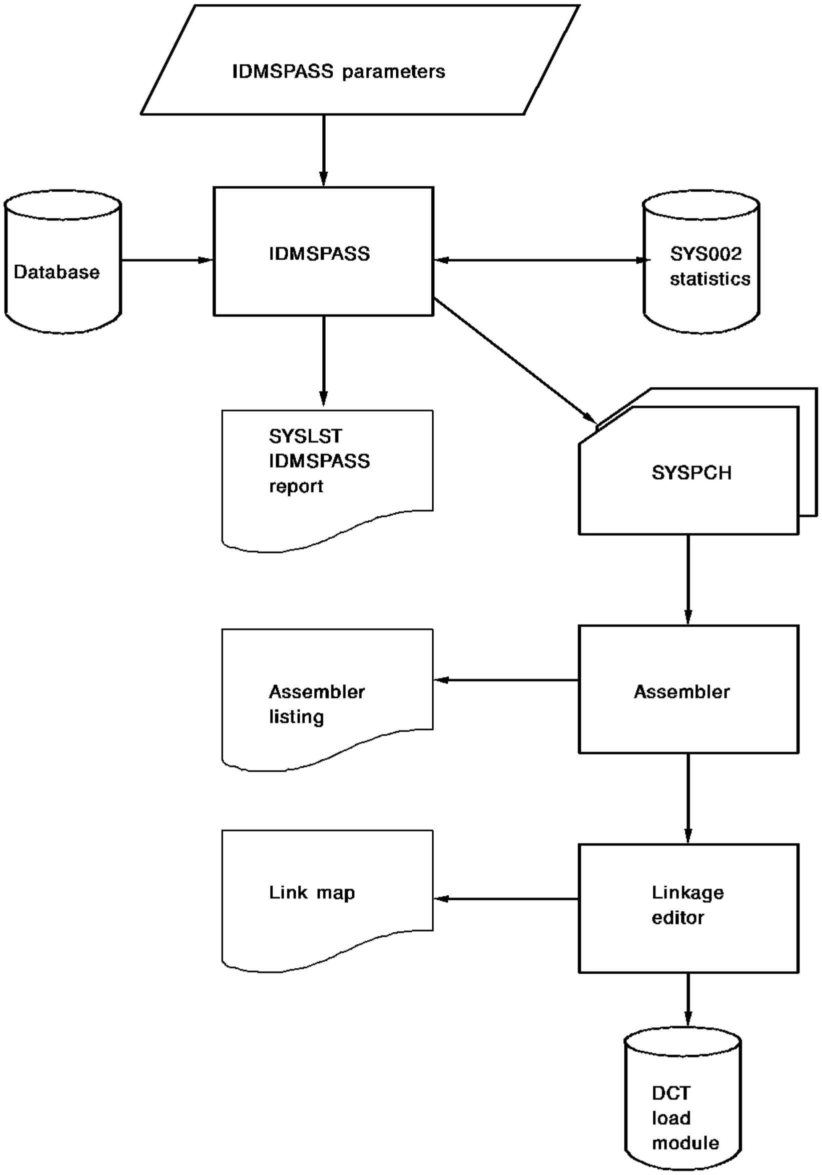
You must define the DCT to the CV just like a program.

Note: Always save a copy of the source code for each DCT load module. For more information, see Creating DCTs.

After you specify compression using a SCHEMA RECORD or CREATE TABLE statement, the DBMS uses the custom DCTs that you have created with IDMSPASS at application program runtime.

Summary of IDMSPASS Execution

The following flow chart summarizes IDMSPASS execution.



IDMSPASS Syntax

IDMSPASS syntax is submitted as input to the system through JCL to control a run of the IDMSPASS utility.

For sample IDMSPASS JCL, see Sample JCL.

Authorization

To run IDMSPASS, you must hold DBAREAD privilege on each area accessed.

Syntax

```
►►─── PROCESS ─────────────────────────────────────────────────────────►

                ┌──────────────────────────────────────────┐
 ►─┬─ SQL ─▼─┬─ TABLE schema-name.table-identifier ─┬─────┴──────────►─
   │         └─ AREA segment-name.area-name ────────┘
   │
   └─ SUBSCHEMA subschema-name ─┬─────────────────────────┬─────────────
                                └─ DBNAME database-name ──┘

─►───────────────────────────────┬─────────────────────────────────────►
                                 │
    ┌──────────────────────────┐ │
 ─▼─┬─ RECORD record-name ─┬───┴─┘
    └─ AREA area-name ─────┘

 ►─┬───────────────────────┬───────────────────────────────────────────►
   └─ DCT dctable-name ────┘

 ►─┬───────────────────────┬───────────────────────────────────────────►
   └─ START start-number ──┘

 ►─┬───────────────────────┬───────────────────────────────────────────►
   └─ SKIP skip-number ────┘

 ►─┬───────────────────────┬───────────────────────────────────────────►
   └─ NRECS sample-number ─┘

 ►─┬─────────────────────────────────────────┬─────────────────────────►
   └─ DEGREE compression-degree-number ──────┘

 ►─┬───────────────────────┬───────────────────────────────────────────►
   └─ USCNTS ─┬─ Y ─┤
              └─ N ─┘

 ►─┬─────────────┬─────────────────────────────────────────────────────►◄
   └─ COMPARE ───┘
```

Parameters

SQL

Indicates that IDMSPASS is to process tables in a CA IDMS/DB SQL-defined database.

TABLE *schema-name.table-identifier*

Specifies a table for IDMSPASS 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 IDMSPASS to process.

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

SUBSCHEMA *subschema-name*

Names the subschema for IDMSPASS 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, see to *CA IDMS Common Facilities Guide*.

RECORD *record-name*

Specifies a record for IDMSPASS to sample.

AREA *area-name*

Specifies an area that contains records for IDMSPASS to process.

IDMSPASS 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 IDMSPASS 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 IDMSPASS.

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

SKIP ***skip-number***

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

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

NRECS ***sample-number***

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

Sample-number must be an integer.

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

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

DEGREE ***compression-degree-number***

Specifies the degree to compress records using the DCT created in the current run of IDMSPASS.

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

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

USCNTS

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

Usage

Data That Can Be Accessed in One Run

IDMSPASS 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 IDMSPASS 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 IDMSPASS, 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 IDMSPASS 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 IDMSPASS:

- 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 IDMSPASS to do:

IDMSPASS 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).

Examples

IDMSPASS Run Against Records

In this example, the input parameters direct IDMSPASS 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 compresses 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
```

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 IDMSPASS Syntax.

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 for compression
  - Use a custom DCT for compression

Contents of an IDMSPASS Report

An IDMSPASS report, part of the IDMSPASS output, contains:

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

Reading the IDMSPASS 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

```
IDMSPASS volser                        Database Prepass and DCT generate utility
mm/dd/yy hhmmss
  PAGE     1

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

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 -------- yy-mm-dd
         COMPILE TIME -------- hh.mm.ss
         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 IDMSPASS, see to *CA IDMS Messages and Codes Guide*.

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

```
                  Summary of Records Sampled

                        Avg.     Number         Percent    Number of Bytes
Record Name       F/V  Length    Sampled      Compressible   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 IDMSPASS Syntax (see page 29).

'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

                 Average Length After Compression and Percentage of Original Size by
Degree
Record Name       1        2        3        4        5        6        7        8
----------- -------- -------- -------- -------- -------- -------- -------- ---------
OFFICE        72 72%  68 68%  64  64%  64  64%   64 64%   64  64%   64 64%   64 64%_
NON-HOSP-CLAIM136 16% 124 15% 116  14% 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%   76 51%_
HOSPITAL-CLAIM156 50% 140 45% 132  42% 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%   60 75%_
DENTAL-CLAIM   84 10%  76  9%  76   9%  72   8%   76  9%   76   9%   72  8%   72  8%_
COVERAGE  NO COMP.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 volser                       Database Prepass and DCT generate utility
mm/dd/yy hhmmss
  PAGE     1

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

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

UT000018 RUN UNIT HAS BEEN BOUND
         SUBSCHEMA NAME ------ PASSSS01
         COMPILE DATE -------- yy-mm-dd
         COMPILE TIME -------- hh.mm.ss
         SCHEMA VERSION ------ 1100
         SUBSCHEMA VERSION --- 1200


UT002017 Starting Scan - Area SQLDEMO.EMPLAREA

                 Summary of Tables Sampled

                        Avg.      Number        Percent    Number of Bytes
Table Name        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

                              Average Length After Compression and Percentage of
Original Size by Degree
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. NO COMP. NO COMP. NO COMP. NO COMP. NO COMP. NO COMP.NO
COMP._

              You chose Degree 4
```

Creating DCTs

Selecting a DCT for Compression

A DCT is a conversion table built from data that IDMSPASS sampled from database records. 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 the product
- 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 IDMSPASS Syntax.

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.

You can modify the generated assembler source code to set AMODE=31 and RMODE=ANY. This eliminates warnings during the linkedit about mismatched AMODEs.

You must define the DCT to the CV just like a program.

For more information about incorporating DCT load module generation in the JCL stream for IDMSPASS, see Sample JCL.

Appendix A: Creating DCTs

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, 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 compression1
Record control block	Alphanumeric	40 bytes	(Filler— unused)

Parameter	Usage	Length	Description
	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 field1
	Binary	2 bytes	Maximum length of the variable length record (decompressed size) including the RDW1
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 IDMSPRES
  24– Load of IDMSPRES failed
>100– Error during decompression
             (most likely, the wrong DCT was specified);
             PRESSTO return code = return code minus 100
```

Appendix B: Sample JCL

This section contains the following topics:

About This Appendix

In this appendix you will find sample JCL or commands for executing IDMSPASS on the following operating systems:

- z/OS
- z/VSE
- z/VM

z/OS JCL

IDMSPASS (z/OS)

```
//PASS     EXEC PGM=IDMSPASS,REGION=1024K
//STEPLIB  DD   DSN=idms.dba.loadlib,DISP=SHR
//         DD   DSN=idms.custom.loablib,DISP=SHR
//         DD   DSN=idms.cagjload,DISP=SHR
//userdb   DD   DSN=user.userdb,DISP=SHR
//dcmsg    DD   DSN=idms.sysmsg.ddldcmsg,DISP=SHR
//sysjrnl  DD   DUMMY
//SYS002   DD   DSN=idms.statfile,DISP=(NEW,CATLG,DELETE),
//             UNIT=disk,VOL=SER=vvvvvv,SPACE=(TRK,(10,5)),
//             DCB=(RECFM=FB,LRECL=1032,BLKSIZE=bbbbb)
//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=yourHLQ.CAGJMAC,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)
/*
//*
```

assembler	Name of the assembler program
bbbbb	Block size of the statistics file; must be a multiple of 1032
dcmsg	DDname of the system message (DDLDCMSG) area
dctable-name	Name of the DCT
dmcl-name	Name of the DMCL
dictionary-name	Dictionary containing SQL catalog component
disk	Symbolic device name for work files
idms.dba.loadlib	Data set name of the load library containing the DMCL and database name table load modules
idms.cagjload	Data set name of the load library containing the CA IDMS executable modules that do not require customization
idms.custom.loadlib	Data set name of the load library containing the customized CA IDMS executable modules
yourHLQ.CAGJMAC	Data set name of the CA IDMS macro library
yourHLQ.CAGJSRC	Library containing member #CRQID
idms.statfile	File containing statistics gathered by IDMSPASS on sampled records
idms.sysmsg.ddldcmsg	Data set name of the system message (DDLDCMSG) area
sysjrnl	DDname of the dummy journal file
sys1.maclib	Data set name of the operating system macro library

user.loadlib	Data set name of the load library containing the user-defined executable modules
userdb	DDname of the user database file
user.userdb	Data set name of the user database file
vvvvvv	Volume serial identifier
&.&.*dctobj*	Data set name of the temporary file containing the assembler output
&.&.*dctsrc*	Data set name of the temporary file containing the assembler source that becomes the DCT

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:

```
//dccat   DD   DSN=idms.syssql.ddlcat,DISP=SHR
//dccatx  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 z/OS. For instructions on how to assemble and link-edit a module using SMP/E, see to "System Modification Procedures" in *CA IDMS Installation and Maintenance Guide—z/OS*.

z/VSE JCL

IDMSPASS (z/VSE)

```
// EXEC PROC=IDMSLBLS
// EXTENT SYSCLB,nnnnnn,,,ssss,llll
   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,llll
   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
```

dct	Name of the IDMSPASS module; must be the same as DEVICE-MEDIA NAME in source code

dmcl-name	DMCL name
dictionary-name	Dictionary containing SQL catalog component
f	File number of the statistics file
idms.statfile	File ID of file that contains statistics gathered by IDMSPASS; this file can go to either disk or tape
llll	Number of tracks (CKD) or blocks (FBA) of disk extent
nnnnnn	Volume serial number
ssss	Starting track (CKD) or block (FBA) of disk extent
*sys*009	Logical unit assignment of the dummy journal file
*sys*010	Logical unit assignment of the user database file
*sys*020	Logical unit assignment of the punched output
temp.dct	File ID of the temporary data set containing the assembler source that becomes the DCT

Usage

Reassembling CA IDMS Modules

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

z/VM Commands

IDMSPASS (z/VM)

```
*
*    IDMSPASS STEP
*
FILEDEF SYSPCH DISK dct ASSEMBLE a (RECFM FB LRECL bbbb2
FILEDEF SYS002 DISK idmspass 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 idmspass 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
```

bbb	Block size of the user input data file
bbbb1	Must be a multiple of 1032
bbbb2	Must be a multiple of 80
dct	File name of the data characteristic table
dct ASSEMBLE *a*	File ID of the file containing the assembler source that becomes a DCT
idmspass input a	File ID of the file containing the IDMSPASS input parameters
idmspass stats a	File ID of the file containing statistics gathered by IDMSPASS on sampled records

lll	Logical record length of the user input data file
nnn	Number of pages in the user database file
ppp	Page size of the user database file
sysidms parms a	File ID of the file containing SYSIDMS parameters
userdb	DDname of the user database file
user userdb a	File ID of the user database file

Usage

Local Mode

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

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

  ```
  OSRUN IDMSPASS 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 IDMSPASS input parameters:

1. On the z/VM 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 IDMSPASS input parameters
4. Press [Enter] to exit input mode
5. On the XEDIT command line, type **FILE** and press [Enter]

SYSIDMS File

To run IDMSPASS, you should include these SYSIDMS parameters:

- DMCL=*dmcl-name*, to identify the DMCL
- If you are running IDMSPASS 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 z/VM 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, see to *CA IDMS Common Facilities Guide*.

Appendix C: Third Party License Agreements

CA IDMS™ Presspack includes software developed by third-party software providers. This appendix provides information regarding these third-party software providers.

This section contains the following topics:

PRESSTO

CA IDMS™ Presspack utilizes the data compression program PRESSTO™ a proprietary product of Ironwood Software, Inc.

Index