CA Datacom® Presspack

User Guide

Version 14.02

CA Technologies Product References

This document references the following CA products:

- CA Datacom®/DB
- CA Datacom® Datadictionary™
- CA Datacom® Presspack
- CA Datacom® VSAM Transparency

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: Overview of CA Datacom Presspack

CA Datacom Presspack is a data compression tool for use at CA Datacom/DB z/OS installations. Its compression efficiency makes it ideal for compressing large-volume databases.

CA Datacom Presspack consists of the following:

- Runtime compression and decompression routines.
- A built-in Data Characteristic Table (DCT).
- The DTCMRVRT batch utility to uncompress backup files.
- The RDTGEN macro to generate an optional Record Descriptor Table (RDT) load module, used for special file processing.
- The DTCMPASS batch utility that allows you to create a custom DCT.
- The DTCMPASS macro to assemble a custom DCT created with the DTCMPASS utility.

By using CA Datacom Presspack, you can do the following:

- **Conserve disk space.**

  Results in significant reductions in storage costs.

- **Compress more data than is typically compressible.**

  Compress CA Datacom/DB tables containing any data type. Most similar products compress mainly character data.

Weak, Strong, and Ziv-Lempel (ZL) Compression

With any compression technique, there is a balance between storage saved, CPU time spent, and simplicity of use. The relative importance of each of these factors to you depends on the priorities of your site. Therefore, CA Datacom Presspack offers you options:

Weak Compression

Offers speed and simplicity. Weak compression is designed for users who are satisfied with marginal compression as long as there are no significant CPU costs.

Strong Compression

Offers the best compression at remarkable speed. CA Datacom Presspack offers two varieties of strong compression, allowing you to further balance simplicity of implementation with results. The two varieties of strong compression are discussed next.

ZL Compression

Uses a custom-built Ziv-Lempel (ZL) dictionary, optimized for one or more specified tables. It uses the IBM compression call instruction CMPSC, if available. If CMPSC is not available, CA Datacom Presspack software emulation is used.

You specify the compression method for each CA Datacom/DB table. Therefore, you can balance storage saved compared to CPU time spent on a table by table basis.

Custom Compression Tables

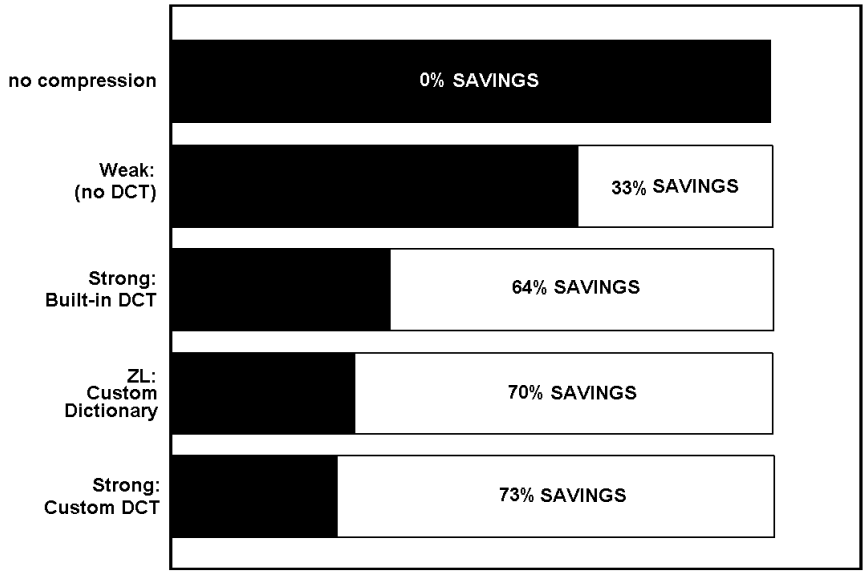
A prebuilt Custom Compression Table (CCT) is required for both the Strong and ZL methods. CCTs are reentrant load modules that can be link-edited to load above or below the 16-MB line. The Strong method requires a CCT known as a Data Characteristic Table (DCT), which you build with the DTCMPASS utility. The ZL method requires a CCT known as a ZL dictionary, which you build with an IBM-supplied REXX utility CSRBDICT.

CA Datacom Presspack loads the CCT upon first reference (or it is link-edited with the CA Datacom Presspack module) and determines whether it is a DCT or a ZL dictionary. The compression method is chosen accordingly.

We recommend that you have a single CCT library which contains all your RDTs, DCTs, and ZL dictionaries. If a disaster recovery situation occurs, you know where your CCTs are. Verify that each CCT has a unique name. Never replace a CCT with a different CCT having the same name, because multiple files can share it. Additionally, include the CCT library in your normal backup list. If you lose a CCT, it could mean the loss of a file.

Amount of Compression Achieved

You can achieve several different types of compression through the use of CA Datacom Presspack. The first type of CA Datacom Presspack compression is weak compression which compresses by removing repeated strings. The second type of CA Datacom Presspack compression is strong compression which uses a data characteristic table (DCT) to encode the data. Strong compression can be achieved using the built-in DCT supplied with CA Datacom Presspack. You can achieve a greater compression ratio by building a custom DCT. The third type, which uses the IBM compression instruction CMPSC (if available, or software emulation if not available), uses a custom-built ZL dictionary in sizes 8 K, 16 K, 32 K, 64 K, and 128 K depending on the degree of compression you want to obtain.



Compression Comparisons for a Typical Record

The graph shows the compression comparison for a typical record. The record consists of about 40 percent repetition strings, 30 percent alpha text, 20 percent packed decimal, 10 percent zoned decimal, and no binary data. Your percentage of savings varies based on the types of data in the record and other factors.

How Strong Compression Works

CA Datacom Presspack uses three data compression steps to achieve strong compression:

1. Elimination of repeating character strings.
2. Use of Huffman Encoding techniques to compress database tables (see the following).
3. Use of Successor Tables to maximize Huffman Encoding (see the following).

Huffman Encoding

Huffman Encoding is an algorithm which assigns bit string codes of different lengths to single characters and character strings. These codes substitute for the characters and character strings in the stored data. The result is that instead of every character using a fixed eight-bit code, some characters have as few as two-bits and others require more than eight-bits. Because the most frequently occurring characters have the shortest bit strings, the result is that the data is stored in less space.

To assign the codes, CA Datacom Presspack uses character and character-string frequencies of occurrence. This method assigns shorter codes to the most frequently occurring characters and character strings. To the characters and character strings that occur less frequently, CA Datacom Presspack assigns longer codes.

Successor Tables

CA Datacom Presspack uses successor tables to allow, in effect, multiple characters to be assigned the same bit string.

In reality, characters that make up your data do not occur in a random fashion. For example, it is likely that an alpha character follows another alpha character rather than by a numeric character or a symbol. Similarly, a numeric character is likely to ollow another numeric character. Within a data characteristic table (DCT), CA Datacom Presspack stores multiple compression tables. The first compression table records not only how to compress a character but also the successor table, that is, the compression table to use to map the next character.

For example, if CA Datacom Presspack encounters the letter M and converts it to a bit string, it uses the successor table information to determine which conversion table to use for the next character. In this case, it would select a table which gives shorter bit strings to the letters which DTCMPASS had found most frequently followed the letter M in the sample data.

Optimizing Strong Compression

To optimize data compression, process CA Datacom/DB data with DTCMPASS, the CA Datacom Presspack compression optimization utility. DTCMPASS samples the data in a table, recording the frequencies of various character strings in the data. This information is used to assign bit string codes for CA Datacom Presspack to use to replace the character strings. For details, see Using the DTCMPASS Utility (see page 19).

Using the Built-in DCT

If you do not use DTCMPASS, you can still achieve good compression results. CA Datacom Presspack uses a built-in DCT when compressing tables that do not have a custom DCT. The information in the built-in DCT consists of statistics that describe typical character data.

Using the ZL Method

The CMPSC hardware instruction and CA Datacom Presspack software emulation both require ZL dictionaries. See the IBM *ESA/390 Data Compression* guide, SA22-7208.

When to Use CA Datacom Presspack

Use CA Datacom Presspack for the following:

- Tables that can be compressed by more than seven-bytes.
  (CA Datacom Presspack requires seven-bytes per table as overhead.)
- Infrequently updated tables.
- Long-lived tables.

CA Datacom Presspack can compress tables that contain data in any data type supported by CA Datacom/DB, including binary or packed decimal. However, to obtain the most efficient compression for a table containing noncharacter data, first process the data area containing the table with DTCMPASS. For more information, see Using the DTCMPASS Utility (see page 19).

Note: A table containing binary data can actually increase up to seven-bytes if you enable CA Datacom Presspack for the table and a customized DCT is not created for it.

When Not to Use CA Datacom Presspack

Do not use CA Datacom Presspack to compress a table if any of the following are true:

- CPU overhead exceeds the space savings that you can obtain for the table.
- You cannot obtain more than 7 bytes in space savings for the table.
  (CA Datacom Presspack requires 7 bytes per table as overhead.)
- The table contains CA Datacom VSAM Transparency variable-length records.

Chapter 2: Implementing CA Datacom Presspack

For details on installing CA Datacom Presspack, refer to the information about the installation process in the *CA Datacom Presspack Installation and Maintenance Guide*.

Compression Decision-Making

After CA Datacom Presspack is installed, make the following decisions:

- Which tables in the information base to compress.
- Which tables to compress using weak compression.
- Which tables to compress using strong compression with the built-in DCT.
- Which tables to compress using the Ziv-Lempel (ZL) method for IBM hardware compression (or CA Datacom Presspack software emulation) with a custom ZL dictionary.
- Which tables to process with DTCMPASS to build a custom DCT.
- How many DCTs to create with DTCMPASS. You can process several tables with DTCMPASS and merge the statistics into one DCT.
- Which tables to process with a RDT to leave fields such as keys uncompressed.

When first implementing CA Datacom Presspack, try weak compression for one or two tables before taking steps to implement strong compression.

Accumulating Statistics

DTCMPASS can accumulate statistics without creating a DCT. At the same time, DTCMPASS can report information about each table processed. This information includes projections of how much the table could be compressed at each of eight degrees of compression. Use this information in making compression decisions before running DTCMPASS to create a DCT for a data area. For more information, see the sample DTCMPASS report in DTCMPASS Report (see page 32).

Using DTCMPASS Report Data

Use the data in a DTCMPASS report to:

- Select a degree of compression for the tables in the data area.
- Decide whether to:
  - Use weak compression
  - Use the built-in DCT
  - Use a custom DCT or RDT/DCT pair
  - Forego compressing selected tables

Implementing Compression Without a Custom DCT

For each table for which you decide to implement weak compression or strong compression without building a custom DCT:

1. Use DBUTLTY to BACKUP the data area containing the table. For details on using the DBUTLTY BACKUP function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

   Note: Specify CMPRS=NO when executing the DBUTLTY BACKUP. Do not specify RECID=YES. The backup must be a reorganizing backup.

2. Specify PRESSPAK as the TABLE entity COMPRESS-EXIT attribute-value. For details about specifying the COMPRESS-EXIT attribute, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

   Note: Specify the COMPRESSION attribute value as N when you specify a value for the COMPRESS-EXIT attribute. Use CA Datacom Presspack in place of, but not in addition to, the standard database compression.

3. To use weak compression for the table, specify WEAK as the TABLE entity ENCRYPTION-KEY attribute-value.

   Note: For more information about specifying the ENCRYPTION-KEY attribute, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

   To implement strong compression for the table using the built-in DCT, do not specify any value in the ENCRYPTION-KEY attribute of the table.

4. CATALOG the modified CA Datacom Datadictionary definition.

   Note: For more information about cataloging CA Datacom Datadictionary definitions, see the *CA Datacom DatadictionaryOnline Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

5. Use DBUTLTY to LOAD the data area containing the table.

   Because the definition of the table now specifies CA Datacom Presspack compression, DBUTLTY calls CA Datacom Presspack to compress the data of the table when loading the data area.

   Note: For more information about using the DBUTLTY LOAD function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

6. Run existing application programs.

   Because the definition of the table now specifies CA Datacom Presspack compression, CA Datacom/DB calls CA Datacom Presspack when an application requests access to the table. If the application retrieves data from the table, CA Datacom Presspack decompresses the data before CA Datacom/DB returns the requested data to the application. If the application updates or adds data to the table, CA Datacom Presspack compresses the data before CA Datacom/DB writes the data.

Implementing Compression With a CCT or RDT

For each table for which you decide to build a custom DCT, ZL dictionary, or an RDT, follow these steps:

1. Use DTCMPASS to generate a DCT. You can optionally specify an RDT to exclude fields from compression by a DCT. For more information, see Assembling and Linking a Record Descriptor Table (RDT) (see page 35). You can generate a DCT specifically for this table or generate one DCT for use with several tables which contain similar data. For details about DTCMPASS, see Using the DTCMPASS Utility (see page 19), or use CSRBDICT, an IBM-supplied REXX EXEC, to build a custom ZL dictionary.

2. Use DBUTLTY to BACKUP the data area containing the table. For details, see the *CA Datacom/DB DBUTLTY Reference Guide*.

   Note: Specify CMPRS=NO when executing DBUTLTY BACKUP. Do not specify RECID=YES. The backup must be a reorganizing backup.

3. Specify PRESSPAK for the COMPRESS-EXIT attribute value for the TABLE entity.

   Note: For details on specifying the COMPRESS-EXIT attribute, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

   Note: Specify the COMPRESSION attribute value as N when you specify a value for the COMPRESS-EXIT attribute. Use CA Datacom Presspack in place of, but not in addition to, the standard database compression.

4. Specify the name of the CCT (that is, the DCT or ZL dictionary) or RDT (assembled in step 1) for the ENCRYPTION-KEY attribute value for the TABLE entity.

   Note: For details about specifying the ENCRYPTION-KEY attribute, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

5. CATALOG the modified CA Datacom Datadictionary definition.

   Note: For details on cataloging CA Datacom Datadictionary definitions, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

6. Use DBUTLTY to LOAD the data area containing the table.

   Because the table definition now specifies CA Datacom Presspack compression, DBUTLTY automatically calls CA Datacom Presspack to compress the data in the table when loading the data area.

   Note: For details about using the DBUTLTY LOAD function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

7. Run existing application programs.

Because the table definition now specifies CA Datacom Presspack compression, CA Datacom/DB calls CA Datacom Presspack when an application requests access to the table. If the application retrieves data from the table, CA Datacom Presspack decompresses the data before CA Datacom/DB returns the requested data to the application. If the application updates or adds data to the table, CA Datacom Presspack compresses the data before CA Datacom/DB writes the data.

Chapter 3: Using the DTCMPASS Utility

DTCMPASS samples the data in the selected table to gather statistics on characteristics of the data. Using these statistics, DTCMPASS builds a custom Data Characteristic Table (DCT) that describes the data in the table. At runtime, CA Datacom Presspack uses the custom DCT to compress or decompress the data when reading or writing to the table. Because the custom DCT contains information specific to the data, CA Datacom Presspack can compress the data efficiently.

Creating Custom DCTs

If you process one table with DTCMPASS, it creates a single DCT specifically for use with that table. If you process more than one table with a single execution of DTCMPASS, DTCMPASS creates a single DCT for all of the tables processed.

The more specific that a DCT is to a table, the more effective the compression of that table. Accordingly, if you create a DCT for a single table, you achieve the most effective compression for that table. Optionally, you can specify a RDT to exclude fields from being processed (see Assembling and Linking a Record Descriptor Table (RDT) (see page 35)).

Consider the characteristics of the data in your information base, cost, and efficiency when deciding how many DCTs to create. For example, create a custom DCT for all of your text type tables (names, addresses, and so on). One custom DCT developed sampling these tables usually produces excellent compression because the data in all of these tables has similar characteristics. Build another custom DCT for tables containing packed data. In general, create one DCT for each group of tables containing similar data. Each file can optionally have a different RDT, but in that case each RDTGEN must specify the same DCT.

In summary, if you have 100 tables, creating a custom DCT for each produces the best compression but forces you to track 100 different DCTs. By grouping tables which have similar data characteristics, you can generate six or seven custom DCTs that result in compression closely approximating what you would achieve with 100 DCTs.

Caution! If a custom DCT is modified, destroyed, or lost, CA Datacom Presspack is unable to expand any data which remains compressed with that custom DCT. Always verify that you have a backup copy of every custom DCT. Never modify an existing DCT once you have used it to compress data.

Creating a Generic DCT

If you do not specify a custom DCT, CA Technologies provides a generic DCT that CA Datacom Presspack uses. However, you can use DTCMPASS to create your own generic DCT.

By processing all of the tables in a database with DTCMPASS, you can create a DCT geared to the special characteristics of the data in your database. You can then use this DCT whenever you would have used the built-in DCT to compress your tables. Because this generic DCT is generated from the data in your tables, it can provide more efficient compression than the built-in DCT.

What DTCMPASS Does

At runtime, DTCMPASS uses the input parameters in SYSIN to read a data set containing the data extracted from a CA Datacom/DB table. DTCMPASS tracks frequency of occurrence character strings and repeating strings. This information is used to print the DTCMPASS report and to generate assembler source code for the DCT which it punches into SYSPCH. To produce a DCT load module, assemble and link the assembler source code generated by DTCMPASS.

Procedure for Building a Custom DCT for a Single Table

Perform the following steps to build a custom DCT for a single table:

1. Execute the DBUTLTY EXTRACT function to produce a data set containing the table data. For more information about the EXTRACT function, see the *CA Datacom/DB DBUTLTY Reference Guide*
2. Execute DTCMPASS against the data set containing the output from the DBUTLTY EXTRACT function. (See the sample JCL in Sample JCL for Processing a Single Table (see page 27).) DTCMPASS produces the following:
   - A report (see the sample DTCMPASS report in DTCMPASS Report (see page 32)).
   - A custom DCT for the table.
   - A statistics file (SYSUT2 in the sample JCL in Sample JCL for Processing a Single Table (see page 27).).

3. Review the DTCMPASS report which provides information about how much compression is achieved by using each of the eight degrees of compression. Increasing the degree of compression increases the size of the DCT (3360 bytes/degree). Decide if you are satisfied with the degree of compression you specified to DTCMPASS. If so, you can delete the statistics file.
4. To produce a DCT with a different degree of compression than you specified in the previous step, reexecute DTCMPASS, using the sample JCL in Sample JCL Changing Degree of Compression (see page 31). After this execution of DTCMPASS completes, you can delete the statistics file.

Procedure for Building a Custom DCT Shared by Several Tables

Perform the following steps to build a custom DCT shared by three tables:

1. Execute the DBUTLTY EXTRACT function to produce a data set for each table containing the table data. For more information about the EXTRACT function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

   Note: The sample JCL for steps 2, 3, and 4 is in Sample JCL for Processing Three Tables (see page 29).
2. Execute DTCMPASS against one of the data sets containing output from the DBUTLTY EXTRACT function, specifying DEGREE=0, to gather statistics about the first table. For more information, see How to Use DTCMPASS (see page 22). This execution produces:

   - A report (see the sample DTCMPASS report in DTCMPASS Report (see page 32)).
   - A statistics file (SYSUT2 in the sample JCL in Sample JCL for Processing Three Tables (see page 29)).

   The statistics file is used as input to the next step. Save the report for use in step 5.
3. Execute DTCMPASS against another of the data sets containing output from the DBUTLTY EXTRACT function, specifying DEGREE=0,USCNTS=Y, to gather statistics about the second table (see How to Use DTCMPASS (see page 22)). This execution of DTCMPASS produces:

   - A report (see the sample DTCMPASS report in DTCMPASS Report (see page 32)).
   - A statistics file (SYSUT2 in the sample JCL in Sample JCL for Processing Three Tables (see page 29)).

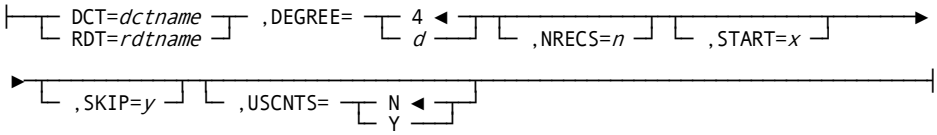
   The statistics file is used as input to the next step. Save the report for use in step 5.

4. Execute DTCMPASS against the last data set containing output from the DBUTLTY EXTRACT function to gather statistics about the last table and to build a DCT. Specify a value other than zero for DEGREE= or allow it to default and specify USCNTS=Y (see How to Use DTCMPASS (see page 22)). This execution of DTCMPASS produces:
   - A report (see the sample DTCMPASS report in DTCMPASS Report (see page 32)).
   - A custom DCT for the three tables.
   - A statistics file (SYSUT2 in the sample JCL in Sample JCL for Processing Three Tables (see page 29)).
5. Review the DTCMPASS reports (from steps 2Ä4) which provide information about how much compression is achieved by using each of the eight degrees of compression. Increasing the degree of compression increases the size of the DCT (3360 bytes/degree). Decide if you are satisfied with the degree of compression you specified to DTCMPASS. If so, you can delete statistics file (SYSUT2 in the sample JCL in Sample JCL for Processing Three Tables (see page 29)).

   To produce a DCT with a different degree of compression than you specified in the previous step, reexecute DTCMPASS, specifying NRECS=0,USCNTS=Y and supplying the statistics file created by DTCMPASS in the previous step as SYSUT2 for this execution. After this execution of DTCMPASS completes, you can delete the statistics file (SYSUT2 in the sample JCL in Sample JCL for Processing Three Tables (see page 29)).

How to Use DTCMPASS

To execute DTCMPASS, use the JCL provided in Sample JCL for Processing a Single Table (see page 27), supplying the following keywords:



Note: Place either a comma or a space between keywords. You cannot split a keyword between lines.

Commands

DCT=

Specifies the name of a custom data characteristic table to build with this run of DTCMPASS. You can specify only one DCT name. If you specify a value for DCT=, you cannot also specify a value for RDT=.

Note: When building a new DCT for use in compressing a CA Datacom/DB table which has been compressed using an existing DCT, do not specify the existing DCT name for the new DCT. Specifying an existing DCT name for the new DCT causes the new DCT to overwrite the existing DCT. This makes it impossible to read any data compressed using the existing DCT.

Valid Entries:

1&emdash.8 alphanumeric characters

Default Value:

(No default)

RDT=

Specifies the name of the previously generated RDT load module for the input table. An RDT is required to leave fields uncompressed in a table. The DCT name is taken from the DCT=dctname parameter of the RDTGEN (dctname must match the linkage editor NAME operand). If you specify a value for RDT=, you cannot also specify a value for DCT=.

Valid Entries:

1&emdash.8 alphanumeric characters

The RDTGEN must not specify Weak or ZL compression.

Default Value:

(No default)

Required Keyword

DEGREE=

Specifies the degree to which CA Datacom Presspack compresses tables using this DCT. The degree of compression relates to the number of successor tables which DTCMPASS builds into the custom DCT. The higher the degree, the more successor tables are built and the more tailored the DCT is to the sampled data.

DEGREE=0 specifies that DTCMPASS accumulate statistics but not create a DCT. Use this option, together with USCNTS=, to build up a statistical base.

The size (in bytes) of the DCT increases when you specify a higher degree of compression (by 3360 bytes per degree). The higher the degree of compression, the greater the compression efficiency, but on a diminishing scale. Compression efficiency increases the most from degree 1 through 2, less from 2 to 3, . . . and least from 7 to 8.

An additional consideration is how static your data is. A degree eight DCT is finely tuned to the data sampled by DTCMPASS. Verify that the data sampled by DTCMPASS is representative of the data contained in the table compressed with that DCT. If your data changes over time in such a way that the original sample is no longer representative, CA Datacom Presspack begins to provide less compression for the table than it provided when you first compressed the table.

Valid Entries:

An integer in the range 0&emdash.8

Default Value:

4 (DEGREE=4 yields medium compression efficiency and a 13K DCT.)

Optional Keywords

NRECS=

Specifies the maximum number of records from the DBUTLTY EXTRACT data set that DTCMPASS samples in the current run.

DTCMPASS samples data until it reaches:

1. The end of the data in the table.
2. 4 MB of data in its internal table. (DTCMPASS excludes repeated characters when building this table. For example, a string of XXXXX counts as one character.)
3. The number of records specified for the NRECS= parameter.

You can use the SKIP= parameter with the NRECS= parameter to produce a statistically reasonable sample.

NRECS=0 specifies that DTCMPASS not process any data on this execution. **NRECS=0 is only valid when USCNTS=Y is also specified.**

Valid Entries:

An integer from 0 through 2 billion, inclusive

Default Value:

2 billion (or the entire table)

SKIP=

Specifies the number of records that DTCMPASS skips to obtain statistics. For example, specifying SKIP=1 causes DTCMPASS to sample every other record.

Valid Entries:

An integer

Default Value:

0 (process every record)

START=

Specifies the number of the record of the input data set where DTCMPASS is to begin sampling.

Valid Entries:

an integer

Default Value:

1

USCNTS=

Specifies whether statistics calculated on a previous DTCMPASS run are used. DTCMPASS merges old statistics with the statistics accumulated on this run.

If you specify USCNTS=Y, specify a value greater than zero for either NRECS= or DEGREE=. If you specify a number greater than zero for NRECS= and DEGREE=0, DTCMPASS merges the old statistics with the new statistics and generates a report. If you specify a number greater than zero for both NRECS= and DEGREE=, DTCMPASS merges the old statistics with the new statistics and generates both a report and a custom DCT. If you specify a number greater than zero for DEGREE= and NRECS=0, DTCMPASS uses only the old statistics to generate both a report and a custom DCT.

If you specify USCNTS=N, DTCMPASS uses only statistics calculated in the current run.

Valid Entries:

Y or N

Default Value:

N (do not use previously calculated statistics)

DTCMPASS Keyword Summary

To Achieve This Output	Code These Input Statements
Custom DCT for use with one table	**DCT=dctname,DEGREE=d** Omit the DEGREE=d keyword to accept the default (DEGREE=8).

To Achieve This Output	Code These Input Statements
Custom DCT for use with one table, using previously gathered data. Use this option if you want to experiment with different degrees of compression on the same table, without scanning the DBUTLTY EXTRACT file again. When you specify this option, DTCMPASS uses the statistics in SYSUT2 from a previous run of DTCMPASS.	**DCT=dctname,DEGREE=d,NRECS=0,USCNTS=Y** Omit the DEGREE=d keyword to accept the default (DEGREE=8).
One custom DCT for use with multiple tables	**For the first table:** DCT=dctname,DEGREE=0 **For all other tables, except the last table:** DCT=dctname,DEGREE=0,USCNTS=Y **For the last table:** DCT=dctname,DEGREE=d,USCNTS=Y Omit the DEGREE=d keyword to accept the default (DEGREE=8).
One custom DCT for use with multiple tables where RDTs are used to exclude some fields in some tables from compression.	**For the first table:** RDT=rdtname,DEGREE=0 **For all other tables, except the last table:** RDT=rdtname,DEGREE=0,USCNTS=Y **For the last table:** RDT=rdtname,DEGREE=d,USCNTS=Y Omit the DEGREE=d keyword to accept the default (DEGREE=8). **Note:** The RDTs for these tables have been generated with appropriate KEY=parameters or no KEY= parameter with each DCT= parameter the same.
In these statements, you can use the SKIP=, START=, and NRECS= keywords to limit and select the data scanned by DTCMPASS to build the custom DCT.	

Sample JCL for Processing a Single Table

```
//PASS     EXEC PGM=DTCMPASS,REGION=1M
//STEPLIB  DD DSN=cai.presspak.cuslib,DISP=SHR
//         DD DSN=cai.thlq.cabdload,DISP=SHR
//SYSUT1   DD DSN=extract.data.set,DISP=SHR
//SYSUT2   DD  DSN=datacom.statfile,DISP=(NEW,PASS),
//             DCB=(RECFM=VB,LRECL=1036,BLKSIZE=bbbb1),
//             UNIT=disk,SPACE=(bbbb1,(257))
//SYSUT3   DD  DSN=&.Data Characteristic Table (DCT),DISP=(NEW,PASS),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=bbbb2),
//             UNIT=disk,SPACE=(bbbb2,(257))
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD  *
        DCT=dctname
//ASM      EXEC PGM=assemblr,PARM='DECK,RENT',REGION=1M
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//         DD DSN=cai.thlq.cabdmac,DISP=SHR
//SYSUT1   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSUT2   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSUT3   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSPUNCH DD  DSN=&asmblr.,UNIT=disk,DISP=(NEW,PASS), }
//             SPACE=(80,(400,40)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSIN  DD DSN=&.Data Characteristic Table (DCT),DISP=(OLD,DELETE)           omit
if no DCT
//LINK     EXEC PGM=IEWL,PARM='SREF,LIST,LET,RENT,',           is to be created
//             REGION=1024K
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DDNAME=SYSIN
//INPUT    DD  DSN=&asmblr.,DISP=(OLD,DELETE)
//SYSUT1   DD  UNIT=disk,SPACE=(TRK,(20,5))
//SYSLMOD  DD  DSN=cai.presspak.cuslib,DISP=OLD               .

//SYSIN    DD  *
       INCLUDE INPUT
       NAME dctname(R)
/*
```

JCL Explanation

STEPLIB and SYSLMOD

Specify the data set name of the CA Datacom Presspack load library. Optionally, concatenate STEPLIB or specify SYSMOD of RDT/CCT library if different from CA Datacom Presspack load library. If you are using an RDT, the STEPLIB concatenation must contain RDT load module.

SYSUT1

Specify the name of the data set which contains the output from the DBUTLTY EXTRACT of the table to process.

SYSUT2

Specify the file to contain statistics gathered by DTCMPASS on sampled tables. The block size (*bbbb*1) must be 4 plus a multiple of 1036.

SYSUT3

Specify the data set name of the temporary file containing the assembler source that for the DCT. The block size (*bbbb*2) must be a multiple of 80.

SYSIN

Specify the name for the load module containing the DCT link edit output.

ASM

Specify the name of the assembler program.

SYSLIB

Specify the data set name of the CA Datacom Presspack macro library.

SYSPUNCH and INPUT

Specify the temporary file containing the assembler output.

Sample JCL for Processing Three Tables

The following is a sample JCL for processing three tables.

```
//STEP1    EXEC PGM=DTCMPASS,REGION=1M
//STEPLIB  DD DSN=cai.presspak.cuslib,DISP=SHR
//         DD DSN=cai.thlq.cabdload,DISP=SHR
//SYSUT1   DD DSN=extract.data.set1,DISP=SHR
//SYSUT2   DD  DSN=datacom.statfile,DISP=(NEW,PASS),
//             DCB=(RECFM=VB,LRECL=1036,BLKSIZE=bbbb1),
//             UNIT=disk,SPACE=(bbbb1,(257))
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD  *
        DCT=dctname,DEGREE=0
//STEP2    EXEC PGM=DTCMPASS,REGION=1M
//STEPLIB  DD DSN=cai.presspak.cuslib,DISP=SHR
//         DD DSN=cai.thlq.cabdload,DISP=SHR
//SYSUT1   DD DSN=extract.data.set2,DISP=SHR
//SYSUT2   DD  DSN=datacom.statfile,DISP=(OLD,PASS)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD  *
        DCT=dctname,DEGREE=0,USCNTS=Y
//STEP3    EXEC PGM=DTCMPASS,REGION=1M
//STEPLIB  DD DSN=cai.presspak.cuslib,DISP=SHR
//         DD DSN=cai.thlq.cabdload,DISP=SHR
//SYSUT1   DD DSN=extract.data.set3,DISP=SHR
//SYSUT2   DD  DSN=datacom.statfile,DISP=(OLD,PASS)
//SYSUT3   DD  DSN=&.Data Characteristic Table (DCT),DISP=(NEW,PASS),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=bbbb2),
//             UNIT=disk,SPACE=(bbbb2,(257))
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD  *
        DCT=dctname,DEGREE=d,USCNTS=Y
//ASM      EXEC PGM=assemblr,PARM='DECK,RENT',REGION=1M
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//         DD  DSN=cai.thlq.cabdmac,DISP=OLD
//SYSUT1   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSUT2   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSUT3   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSPUNCH DD  DSN=&asmblr.,UNIT=disk,DISP=(NEW,PASS),
//             SPACE=(80,(400,40)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
```

```
//SYSIN    DD  DSN=&.Data Characteristic Table (DCT),DISP=(OLD,DELETE)
//LINK     EXEC PGM=IEWL,PARM='LIST,LET,RENT',
//              REGION=1M
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DDNAME=SYSIN
//INPUT    DD  DSN=&asmblr.,DISP=(OLD,DELETE)
//SYSUT1   DD  UNIT=disk,SPACE=(TRK,(20,5))
//SYSLMOD  DD  DSN=cai.presspak.cuslib,DISP=OLD
//SYSIN    DD  *
       INCLUDE INPUT
       NAME dctname(R)
/*
```

Note: In the example, if RDTs were used, DCT=dctname would be replaced with RDT=rdtname. Also, RDTs would have previously been generated. An example of the definitions to produce them:

```
RDTGEN RDT=RDT1,DCT=dctname,KEY=(0,,8)
RDTGEN RDT=RDT2,DCT=dctname
RDTGEN RDT=RDT3,DCT=dctname,KEY=((5,,8),(100,,10))
```

For instructions about creating an RDT, see Assembling and Linking a Record Descriptor Table (RDT) (see page 35).

JCL Explanation

In the JCL in Sample JCL for Processing Three Tables (see page 29):

STEPLIB

Specify the data set name of the CA Datacom Presspack load library. Optionally, concatenate STEPLIB or specify SYSMOD of RDT/CCT library if different from CA Datacom Presspack load library. If you are using an RDT, the STEPLIB concatenation must contain RDT load module.

SYSUT1

Specify the name of the data set which contains the output from the DBUTLTY EXTRACT of the first, second, and third tables to process.

SYSUT2

Specify the file to contain statistics gathered by DTCMPASS on sampled tables. The block size (bbbb1) you specify must be 4 plus a multiple of 1036.

SYSIN

Specify the name for the load module containing the DCT link edit output.

SYSUT3

Specify the data set name of the temporary file containing the assembler source that becomes the DCT. The block size (bbbb2) you specify must be a multiple of 80.

ASM

Specify the name of the assembler program.

SYSLIB

Specify the data set name of the CA Datacom Presspack macro library.

SYSPUNCH

Specify the temporary file containing the assembler output.

Sample JCL Changing Degree of Compression

```
//PASS     EXEC PGM=DTCMPASS,REGION=1M
//STEPLIB  DD DSN=cai.presspak.cuslib,DISP=SHR
//         DD DSN=cai.thlq.cabdload,DISP=SHR
//SYSUT1   DD  DSN=extract.data.set,DISP=SHR
//SYSUT2   DD  DSN=datacom.statfile,DISP=(OLD,PASS),
//             DCB=(RECFM=VB,LRECL=1036,BLKSIZE=bbbb1),
//             UNIT=disk,SPACE=(bbbb1,(257))
//SYSUT3   DD  DSN=&.Data Characteristic Table (DCT),DISP=(NEW,PASS),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=bbbb2),
//             UNIT=disk,SPACE=(bbbb2,(257))
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
        DCT=dctname,DEGREE=d,USCNTS=Y
//ASM      EXEC PGM=assemblr,PARM='DECK,RENT',REGION=1M
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//             DSN=cai.thlq.cabdmac,DISP=SHR
//SYSUT1   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSUT2   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSUT3   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSPUNCH DD  DSN=&asmblr.,UNIT=disk,DISP=(NEW,PASS),
//             SPACE=(80,(400,40)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSIN    DD  DSN=&.Data Characteristic Table (DCT),DISP=(OLD,DELETE)
//LINK     EXEC PGM=IEWL,PARM='LIST,LET,RENT,',
//              REGION=1024K
//SYSPRINT DD  SYSOUT=*
```

```
//SYSLIN   DD  DDNAME=SYSIN
//INPUT    DD  DSN=&asmblr.,DISP=(OLD,DELETE)
//SYSUT1   DD  UNIT=disk,SPACE=(TRK,(20,5))
//SYSLMOD  DD  DSN=cai.presspak.cuslib,DISP=OLD
//SYSIN    DD  *
       INCLUDE INPUT
       NAME dctname(R)
/*
```

DTCMPASS Report

Each execution of DTCMPASS produces a report. Use the data in a DTCMPASS report to:

- Select a degree of compression for the tables in the data area.
- Decide whether to:
  - Use weak compression.
  - Use the built-in DCT.
  - Use a custom DCT or use an RDT/CCT pair.
  - Forego compressing selected tables.

```
P R E S S T O -- DTCMPASS UTILITY -- CA, INC.



DCT=DCTPMF,DEGREE=8,NRECS=200,START=1

 PHYS SEQ FILE ORGANIZATION.   FIXED LENGTH RCDS.   RCD LENGTH =   80

    200 RECORDS SAMPLED

REPETITION STRING (WEAK) COMPRESSION ALONE GIVES A COMPRESSION RATIO OF .28

THE EXPECTED COMPRESSION RATIOS FOR THE 8 DEGREE OPTIONS ARE:

DEGREE    1   2   3   4   5   6   7   8

RATIO   .46 .50 .52 .54 .55 .56 .56 .58

 NORMAL END - DTCMPASS
```

DCT...

The DTCMPASS input parameters that you coded for the run.

PHYS SEQ FILE ORGANIZATION...

Details about the data that DTCMPASS accessed during the run.

RECORDS SAMPLED

Number of records sampled in the run.

COMPRESSION RATION OF

Projected percentage of table compression for weak compression.

RATIO

Projected percentage of table compression for each of eight degrees of compression.

Note: DTCMPASS bases its compression projections on the DBUTLTY EXTRACT data which it samples. If the data sampled by DTCMPASS is not typical of the table, actual compression results can vary greatly from the reported projections. Use the NRECS= and SKIP= parameters to select a typical sample.

Chapter 4: Assembling and Linking a Record Descriptor Table (RDT)

This chapter discusses using the RDTGEN macro to assemble and link an RDT.

RDTGEN Macro

For special purpose record processing you can use the optional RDTGEN macro to generate an RDT load module. Functions for which you can use the RDTGEN macro include:

- Making fields within a record exempt from compression
- Specification of the method of compression
- Association of a CCT with the record that is compressed

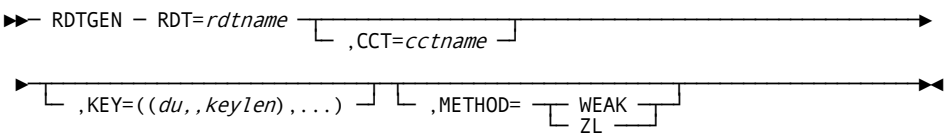
When you use an RDT to compress a file, use that same RDT when you re-expand the records of that file. If you do not use the same RDT, an error message is issued.

When it is appropriate to do so, an RDT can be shared between files. For simplicity, it is recommended that if you are going to associate a CCT with an RDT, you put them both in the same library. With regard to the size of the RDT, although its size depends on the number of keys, it cannot exceed 512 bytes.

With one exception, RDTs are reentrant load modules that you can link edit to load above or below the 16M line. Their reentrancy ensures that RDTs are loaded into protected storage. For the DTCMPASS utility, the module must be linked to be loaded below the 16M line.

The following example shows the JCL and control statements used to create an RDT. Note the following:

- RDTGEN must be preceded and followed by at least one blank.
- Commas separate the operands.
- Although you cannot have operands extending beyond column 71, you can continue lines after a comma by placing an X in column 72 and beginning the next line in column 16.



RDT=rdtname	The unique load module name for this RDT. The link editor name (on the last line of the example, see NAME rdtname) must match the RDT operand name in RDT=rdtname.
CCT=cctname	Specifies the load module name of the CCT that is associated with this RDT. The Strong compression method (using the built-in DCT) is used if you do not use CCT= to specify a CCT and METHOD= to specify a method. You are not allowed to specify CCT= and METHOD=WEAK together, but if METHOD=ZL is specified, CCT= is required. Although to use a CCT you must specify the CCT= parameter, you can eliminate the DTCMPASS or CSRBDICT step and the chance of table mix-up by not specifying CCT=. However, not specifying CCT= does sacrifice some compression. Fields that contain a significant amount of lowercase alphabetic data or binary data (such as the PICTURE COMP fields in COBOL) can grow instead of decrease in size if you do not specify a CCT for files in which those fields reside. An error message is issued if, when you re-expand a record, you do not use the same RDT/CCT pair that was used to compress that record.
KEY=((du,,keylen),...)	Can be used to exempt fields from compression. You can use up to 16 keys. Key displacements must all be below 32K. **du** Is the displacement of key field in the uncompressed record. For the first position of a record, du is 0 (zero). **keylen** Specifies the length of this key field (1 to 255). Omit the outer set of parentheses in the KEY= specification if you are specifying a single key. For example, KEY=(0,,8), not KEY=((0,,8)). **Note:** The double commas are required even though the empty field between the commas is not used at this time.
METHOD=WEAK or **METHOD=ZL**	Use METHOD= to specify either the Weak or Ziv-Lempel (ZL) compression method. If METHOD=ZL is specified, CCT= is required, but you are not allowed to specify CCT= together with METHOD=WEAK.

Sample RDTGEN JCL to Compress a File

```
//ASM      EXEC PGM=assemblr,PARM='DECK,RENT',REGION=1M
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//            DSN=cai.thlq.cabdmac,DISP=SHR
//SYSUT1   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSUT2   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSUT3   DD  UNIT=disk,SPACE=(CYL,(2,2))
//SYSPUNCH DD  DSN=&asmblr.,UNIT=disk,DISP=(NEW,PASS),
//            SPACE=(80,(400,40)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSIN    DD  *
                                                        column 72
         RDTGEN RDT=rdtname&lbrk.,CCT=cctname                      X
                ,KEY=((du,,keylen),...)                            X
                ,METHOD=WEAK                                       X
//LINK     EXEC PGM=IEWL,PARM='LIST,LET,RENT',REGION=1M
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DDNAME=SYSIN
//INPUT    DD  DSN=&asmblr.,DISP=(OLD,DELETE)
//SYSUT1   DD  UNIT=disk,SPACE=(TRK,(20,5))
//SYSLMOD  DD  DSN=cai.presspak.cuslib,DISP=OLD
//SYSIN    DD  *
  INCLUDE INPUT
  NAME rdtname
```

ASM

Specify the name of the assembler program.

SYSLIB

Specify the data set name of the CA Datacom Presspack macro library.

SYSPUNCH and INPUT

Specify the temporary file containing the assembler output.

SYSLMOD

Specify the data set name of the CA Datacom Presspack load library. Optionally, concatenate STEPLIB or specify SYSMOD of RDT/CCT library if different from CA Datacom Presspack load library. If an RDT is used, the STEPLIB concatenation must contain RDT load module.

Sample RDTGEN JCL to Uncompress a File

```
//ASMRDT EXEC PGM=ASMA90,REGION=2048K,PARM=(TERM,'LINECOUNT(66)',
// XREF(SHORT),'USING(WARN(11))',DECK,NOOBJECT,RENT,'SYSPARM(I)')
//SYSLIB   DD DSN=SYS1.MACLIB,DISP=SHR
//         DD  DISP=SHR,
//         DSN=cai.thlq.cabdmac
//SYSUT1    DD UNIT=SYSDA,SPACE=(1700,(600,100))
//SYSUT2    DD UNIT=SYSDA,SPACE=(1700,(300,50))
//SYSUT3    DD UNIT=SYSDA,SPACE=(1700,(300,50))
//SYSTERM   DD SYSOUT=*
//SYSPUNCH DD  DSN=&.&ASMOBJ.,UNIT=SYSDA,
//         SPACE=(3120,(400,100),RLSE),
//         DISP=(,PASS),
//         DCB=BLKSIZE=3120
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
//     RDTGEN RDT=DL1014,CCT=DCT014,KEY=(4,,9)
/*
//LNKRDT  EXEC PGM=IEWL,COND=(5,LT,ASMRDT),
//             PARM='LIST,LET,XREF,MAP,NCAL'
//SYSLIN   DD  DSN=&.&ASMOBJ.,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSLMOD  DD  DSN=cai.presspak.cuslib
//         DISP=SHR
//SYSUT1   DD  DSN=&.&SYSUT1.,SPACE=(1024,(120,120),,,ROUND),
//         UNIT=VIO,DCB=BUFNO=1
//SYSPRINT DD  SYSOUT=*
//SYSIN DD *
 NAME DL1014(R)
/*
```

Chapter 5: Converting CA Datacom/DB Tables

Once you have updated a table definition to specify COMPRESS-EXIT=PRESSPAK or to change the ENCRYPTION-KEY value, convert the table data before accessing it in any way. You can convert:

- A CA Datacom/DB table to use CA Datacom Presspack compression.

  These CA Datacom/DB tables may currently be either uncompressed or compressed with a CA Datacom compression (TABLE attribute COMPRESS=Y) or a user compression exit.

  For this type of conversion, use the procedures in Converting Tables to Use CA Datacom Presspack Compression (see page 40).

- A CA Datacom/DB table from using one form of CA Datacom Presspack compression to using a different form.

  This includes converting:

  - A CA Datacom/DB table which uses Weak compression to use Strong compression (with either the ZL dictionary, the built-in DCT, or a CCT that is a custom DCT).
  - A CA Datacom/DB table which uses compression with the built-in DCT to use a custom CCT or an RDT.
  - A CA Datacom/DB table which uses compression with one custom CCT or RDT to use a different CCT or RDT.
  - A CA Datacom/DB table which uses Strong compression (with either the built-in DCT or a CCT or an RDT) to use Weak compression.

  For any of these four types of conversion, use the procedures in Changing a Table's Compression Mode (see page 41).

- A CA Datacom/DB table which uses CA Datacom Presspack compression to no longer use it.

  For this type of conversion, use the procedures in Removing Tables from CA Datacom Presspack Compression (see page 43).

Converting Tables to Use CA Datacom Presspack Compression

Select the tables that use CA Datacom Presspack compression. For each of the selected tables:

1. Use DBUTLTY to BACKUP the data area containing the table. For details on using the DBUTLTY BACKUP function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

   Note: Specify CMPRS=NO when executing the DBUTLTY BACKUP. Do not specify RECID=YES. The backup must be a reorganizing backup.

2. Specify PRESSPAK for the COMPRESS-EXIT attribute value for the TABLE entity. For details on specifying the COMPRESS-EXIT attribute, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

   Note: Specify the COMPRESSION attribute value as N when you specify a value for the COMPRESS-EXIT attribute. You can use CA Datacom Presspack in place of, but not in addition to, the standard database compression.

3. If the table is to use Weak compression, specify WEAK for the ENCRYPTION-KEY attribute value for the TABLE entity.

   If the table is to use Strong compression with a custom DCT or ZL dictionary, specify the name of the DCT or ZL dictionary module for the ENCRYPTION-KEY attribute value.

   If the table is to use an RDT, specify the name of the RDT module for the ENCRYPTION-KEY attribute value.

   If the table is to use Strong compression with the built-in DCT, leave the ENCRYPTION-KEY attribute blank.

   For details about building a custom DCT, see Using the DTCMPASS Utility (see page 19). For details about specifying the ENCRYPTION-KEY attribute value, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*. For details on creating a ZL dictionary, see Appendix A of the IBM *ESA/390 Data Compression* manual, SA22-7208. For details about creating an RDT, see Assembling and Linking a Record Descriptor Table (RDT) (see page 35).

4. CATALOG the modified CA Datacom Datadictionary definition.

   Note: For details about cataloging CA Datacom Datadictionary definitions, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

5. Use DBUTLTY to LOAD the data area containing the table.

   Since the table definition now specifies CA Datacom Presspack compression, DBUTLTY automatically calls CA Datacom Presspack to compress the table's data when loading the data area.

   Note: For details about using the DBUTLTY LOAD function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

Once the DBUTLTY LOAD has completed, the compressed table is available for CA Datacom/DB access.

Changing a Table's Compression Mode

You must convert a table data to change:

- A CA Datacom/DB table that uses Weak compression to use Strong compression (with either the built-in DCT or a custom DCT or a ZL dictionary) or to use an RDT.
- A CA Datacom/DB table that uses compression with the built-in DCT to use a CCT (a custom DCT or ZL dictionary) or an RDT.
- A CA Datacom/DB table that uses compression with one CCT or RDT DCT to use a different CCT or RDT.
- A CA Datacom/DB table that uses Strong compression (with either the built-in DCT or a custom DCT or a ZL dictionary) or an RDT to use Weak compression.

To convert the table data:

1. Use DBUTLTY to BACKUP the data area containing the table. For details on using the DBUTLTY BACKUP function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

   Note: Specify CMPRS=NO when executing the DBUTLTY BACKUP. Do not specify RECID=YES. The backup must be a reorganizing backup.

2. If the table is to use Weak compression, specify WEAK for the ENCRYPTION-KEY attribute value for TABLE entity.

   If the table is to use Strong compression with a custom DCT, specify the name of the DCT module for the ENCRYPTION-KEY attribute value.

   If the table is to use Strong compression with the built-in DCT, leave the ENCRYPTION-KEY attribute blank.

   If the table is to use Strong compression with a ZL dictionary, specify the name of the ZL dictionary module for the ENCRYPTION-KEY attribute value.

   If the table is to use an RDT, specify the name of the RDT module for the ENCRYPTION-KEY attribute value.

   For details on building a custom DCT, see Using the DTCMPASS Utility (see page 19). For details on specifying the ENCRYPTION-KEY, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*. Appendix A of the IBM *ESA/390 Data Compression* guide, SA22-7208. For details on creating an RDT, see Assembling and Linking a Record Descriptor Table (RDT) (see page 35).

3. CATALOG the modified CA Datacom Datadictionary definition.

   Note: For details about cataloging CA Datacom Datadictionary definitions, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

4. Use DBUTLTY to LOAD the data area containing the table.

   Because the table definition now specifies CA Datacom Presspack compression with the new CCT or RDT, when DBUTLTY calls CA Datacom Presspack to compress the table data, CA Datacom Presspack uses the new CCT or RDT.

   Note: For details about using the DBUTLTY LOAD function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

After the DBUTLTY LOAD has completed, the compressed table is available for CA Datacom/DB access.

Removing Tables from CA Datacom Presspack Compression

After CA Datacom/DB tables have been converted to use CA Datacom Presspack compression, you may occasionally want to remove selected tables from CA Datacom Presspack use.

To disable CA Datacom Presspack for a table:

1. Use DBUTLTY to BACKUP the data area containing the table. For details about using the DBUTLTY BACKUP function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

   Note: Specify CMPRS=NO when executing the DBUTLTY BACKUP. Do not specify RECID=YES. The backup must be a reorganizing backup.

2. Modify the table definition in CA Datacom Datadictionary, removing PRESSPAK from the COMPRESS-EXIT attribute. If the table uses Weak compression or Strong compression with a CCT or RDT, remove the value specified for the ENCRYPTION-KEY attribute.

   Note: For details about modifying the COMPRESS-EXIT attribute see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

3. CATALOG the modified CA Datacom Datadictionary definition.

   Note: For details about cataloging CA Datacom Datadictionary definitions, see the *CA Datacom Datadictionary Online Reference Guide* or *CA Datacom Datadictionary Batch Reference Guide*.

4. Use DBUTLTY to LOAD the data area containing the table.

   Because the table definition no longer specifies CA Datacom Presspack compression, DBUTLTY loads the data uncompressed.

   Note: For details about using the DBUTLTY LOAD function, see the *CA Datacom/DB DBUTLTY Reference Guide*.

After the DBUTLTY LOAD has completed, the uncompressed table is available for CA Datacom/DB access.

Chapter 6: Messages and Codes

DTCMPASS Utility Messages

This section contains a description of messages issued by the DTCMPASS utility. All DTCMPASS utility messages are preceded by a number and are documented in numeric order. I/O error messages are accompanied by operating system messages.

3943

(no message)

Reason:

An operating system message accompanies this message indicating that SYSUT1 cannot be opened.

Action:

Verify that the JCL specification for SYSUT1 points to a valid data set containing output from DBUTLTY EXTRACT. If it does not, correct and rerun. If it does, verify that your job is authorized to access the data set and that there is not a hardware problem.

3944

(no message)

Reason:

An operating system message accompanies this message indicating that SYSUT2 cannot be opened.

Action:

If you have specified USCNTS=Y, verify that SYSUT2 specifies a data set containing statistics from an earlier run of DTCMPASS. If you have specified USCNTS=N, verify that the JCL specification for SYSUT2 is correct. SYSUT2 specifies a data set in which DTCMPASS writes statistics. If the specification for SYSUT2 is incorrect, correct and rerun.

If the specification for SYSUT2 is correct, verify that your job is authorized to access the data set and that there is not a hardware problem.

3946

(no message)

Reason:

An operating system message accompanies this message indicating that either SYSUT1 or SYSUT2 cannot be opened.

Action:

If the problem is with SYSUT1:

Verify that the JCL specification for SYSUT1 points to a valid data set containing output from DBUTLTY EXTRACT. If it does not, correct and rerun. If it does, verify that your job is authorized to access the data set and that there is not a hardware problem.

If the problem is with SYSUT2:

If you have specified USCNTS=Y, verify that SYSUT2 specifies a data set containing statistics from an earlier run of DTCMPASS. If you have specified USCNTS=N, verify that the JCL specification for SYSUT2 is correct. SYSUT2 specifies a data set in which DTCMPASS writes statistics. If the specification for SYSUT2 is incorrect, correct and rerun.

If the specification for SYSUT2 is correct, verify that your job is authorized to access the data set and that there is not a hardware problem.

3948

PRESSTO: I/O ERROR

Reason:

An operating system message accompanies this message indicating the nature of the I/O problem.

Action:

Correct the I/O problem and rerun DTCMPASS.

3949

PRESSTO: I/O ERROR

Reason:

An operating system message accompanies this message indicating the nature of the I/O problem.

Action:

Correct the I/O problem and rerun DTCMPASS.

3950

PRESSTO: I/O ERROR

Reason:

An operating system message accompanies this message indicating the nature of the I/O problem.

Action:

Correct the I/O problem and rerun DTCMPASS.

3951

PRESSTO: PDS HAS NO MEMBER NAME SPECIFIED

Reason:

The data set specified for SYSUT1 or SYSUT2 is a partitioned data set and no member name is indicated.

Action:

Correct and rerun.

3963

STAT-FILE INVALID. SEE 'USCNTS=Y' DESCRIPTION

Reason:

The specification for SYSUT2 in the DTCMPASS JCL is incorrect. When specifying USCNTS=Y, specify SYSUT2 with DISP=(OLD,PASS) and a data set containing statistics from a previous run of DTCMPASS.

Action:

Correct the JCL and rerun.

3965

//SYSUT3 REQUIRED UNLESS 'DEGREE=0' SPECIFIED

Reason:

SYSUT3 is not specified in the DTCMPASS JCL. SYSUT3 indicates the data set in which DTCMPASS writes the code which is assembled as the DCT. SYSUT3 is required by DTCMPASS unless DEGREE=0 is specified, indicating that no DCT is generated.

Action:

Correct and rerun.

3967

CONTROL CARD MUST BE ON //SYSIN

Reason:

DTCMPASS expects to find its control statement as SYSIN in the JCL. Either it did not find a DTCMPASS control statement or it found an invalid control statement.

Action:

Correct and rerun. For correct control statement syntax, see How to Use DTCMPASS (see page 22).

3968

(text varies)

Reason:

This message indicates a syntax problem in the DTCMPASS input stream. The text of the message indicates the nature of the syntax problem.

Action:

Correct and rerun.

3970

(No message)

Reason:

DTCMPASS could not open SYSPRINT. An operating system message accompanies this message indicating the nature of the problem with SYSPRINT.

Action:

Correct and rerun.

3972

NO STAT FILE. REQUIRED IF USCNTS=Y. JOB TERMINATED.

Reason:

You specified USCNTS=Y, but did not supply a statistics file (SYSUT2) containing data from a previous DTCMPASS execution.

Action:

Either supply a valid statistics file or specify USCNTS=N.

3973

UNEXPECTED EOF ON STAT-FILE. JOB TERMINATED.

Reason:

When reading the statistics file (SYSUT2), DTCMPASS did not find complete data.

Action:

Supply a valid statistics file. If one does not exist, rerun DTCMPASS to generate one.

3974

CANNOT OPEN STAT-FILE ON //SYSUT2. JOB TERMINATED.

Reason:

DTCMPASS could not open the statistics file.

Action:

Verify that the JCL specification for SYSUT2 is correct.

PRESSTO Subroutine Return Codes and Messages

This section lists return codes and messages for the following:

CRQC Subroutine (Compression)

4

PRS004: INVALID RDT

Reason:

Module name is invalid.

Action:

Correct encryption-key and rerun.

8

PRS008: INVALID CCT

Reason:

Module name is invalid.

Action:

Correct encryption-key and rerun.

12

PRS012: CANNOT LOAD SPECIFIED RDT/DCT/DICT

Reason:

The RDT/DCT/DICT cannot be loaded.

Action:

Verify the spelling and that the module is on //STEPLIB, //JOBLIB, or in the linklist.

16

PRS016: RECORD LENGTH DISCREPANCY

Reason:

Possible reasons are the following:

- For fixed-length records, the *lur* parameter is not equal to LRECL in the RDT.
- The *lur* parameter is greater than LRECL in the RDT.
- Variable-length uncompressed record ends before final key.
- Record length is greater than 32K-1 for Weak compression.

Action:

Determine the cause of the discrepancy and correct the error.

20

PRS020: INVALID PARAMETER LIST

Reason:

If an Assembly language call is involved, the VL bit is either absent or incorrect.

Action:

Correct the error.

24

PRS024: VERIFICATION FAILURE

Reason:

Possible reasons are the following:

- The verify bit is on in the SCB, but the reexpanded record does not match the original.
- The verify bit is off. PRESSTO automatically verifies the first record in this case, but the reexpanded record does not match the original.

Action:

Determine the reason for the failure and correct the error.

32

PRS032: LOADED RDT/DCT/DICT IS NON-EXECUTABLE

Reason:

The RDT or CCT did not link-edit properly.

Action:

Verify the link edit, correct, relink, and rerun.

36

PRS036: LOADED MODULE IS NEITHER RDT, DCT, NOR DICT

Reason:

An error as indicated in the message text has occurred.

Action:

Verify the spelling and verify that there is not another module with the same name.

40

PRS040: LOADED MODULE IS AN RDT BUT RDT ALREADY PRESENT

Reason:

Possible causes for this error are the following:

- The SCB names an RDT, but the RDT address in the SCB is non-zero.
- The CCT name in the RDT is actually another RDT name.

Action:

Correct the error.

44

PRS044: LOADED MODULE IS A CCT BUT CCT IS ALEADY PRESENT

Reason:

The SCB names a CCT, but the CCT address in the SCB is non-zero.

Action:

Correct the error.

48

PRS048: ZL DICTIONARY NOT ON A PAGE BOUNDARY

Reason:

ZL dictionaries must be linkedited with the PAGE statement to load on a page boundary.

Action:

Correct the error.

52

PRS052: RDT LRECL MUST BE SET FOR ZL COMPRESSION

Reason:

The loaded CCT is a ZL dictionary, but LRECL was not specified in the RDTGEN.

Action:

Correct the error.

56

PRS056: PRESSTO system error -- no message.

Reason:

A PRESSTO system error has occurred.

Action:

Contact CA Support.

60

PRS060: MODULE LOADED ABOVE 16M BUT PROGRAM IN AMODE 24

Reason:

Subroutines must be in AMODE 31 if RDT or CCT is link edited to load above the 16M line.

Action:

Correct the error.

64

PRS064: TARGET EXHAUSTED BEFORE SOURCE

Reason:

The *lur* parameter is incorrect.

Action:

Verify that the *lur* is correct.

68

PRS068: SYMBOL EXCEEDS 260 CHARACTERS

Reason:

The ZL dictionary is invalid.

Action:

Correct the error indicated in the message text.

72

PRS072: MORE THAN 260 CHILDREN

Reason:

The ZL dictionary is invalid.

Action:

Correct the error indicated in the message text.

76

PRS076: CHILD COUNT IS NOT 0-6

Reason:

The ZL dictionary is invalid.

Action:

Correct the error indicated in the message text.

80

PRS080: EXTENSION COUNT IS 5/6/7

Reason:

The ZL dictionary is invalid.

Action:

Correct the error indicated in the message text.

84

PRS084: SIBLING COUNT IS 0

Reason:

The ZL dictionary is invalid.

Action:

Correct the error indicated in the message text.

92

PRS092: COMPR RECORD OR ZL EXPANSION DICTIONARY INVALID

Reason:

The verify function failed. The first record compressed is always verified, even when the verify option is off.

Action:

Determine why the verify function failed and correct the problem.

CRQR Subroutine (Expansion)

104

PRS004: INVALID RDT

Reason:

Module name is invalid.

Action:

Correct encryption-key and rerun.

108

PRS008: INVALID CCT

Reason:

Module name is invalid.

Action:

Correct encryption-key and rerun.

112

PRS012: CANNOT LOAD SPECIFIED RDT/DCT/DICT

Reason:

The RDT/DCT/DICT cannot be loaded.

Action:

Verify the spelling and that the module is on //STEPLIB, //JOBLIB, or in the linklist.

116

PRS016: RECORD LENGTH DISCREPANCY

Reason:

Possible reasons are the following:

- The expanded length differs from the fixed LRECL in the RDT.
- The expanded length is greater than the LRECL in the RDT.
- The expanded record ends before the final key.
- There is a discrepancy between the *lcr* parameter and the length field generated when LL=YES is specified in the RDTGEN.
- The compressed record is not valid.

Action:

Determine the cause of the discrepancy and correct the error.

120

PRS020: INVALID PARAMETER LIST

Reason:

If an Assembly language call is involved, the VL bit is either absent or incorrect.

Action:

Correct the error.

124

PRS024: EXPANDED RECORD DIFFERS FROM ORIGINAL RECORD

Reason:

There is a checkbyte mismatch because of one of the following:

- The *cr* parameter does not point to the start of a compressed record (past any RDW).
- The compressed record has been improperly modified.
- The expansion CCT or DCT is missing or invalid.

Action:

Determine the cause of the problem and correct the error.

128

PRS028: RECORD WAS NOT COMPRESSED WITH SPECIFIED RDT/CCT PAIR

Reason:

Reasons for this error include:

- The RDT or CCT specified in the SCB are not those used to compress the record.
- The *cr* parameter does not point to the start of a compressed record (past any RDW).

Action:

Determine the cause of the problem and correct the error.

132

PRS032: LOADED RDT/DCT/DICT IS NON-EXECUTABLE

Reason:

The RDT or CCT did not link-edit properly.

Action:

Verify the link edit, correct, relink, and rerun.

136

PRS036: LOADED MODULE IS NEITHER RDT, DCT, NOR DICT

Reason:

An error as indicated in the message text has occurred.

Action:

Verify the spelling and that there is not another module with the same name.

140

PRS040: LOADED MODULE IS AN RDT BUT RDT ALREADY PRESENT

Reason:

Possible causes for this error are the following:

- The SCB names an RDT, but the RDT address in the SCB is non-zero.
- The CCT name in the RDT is actually another RDT name.

Action:

Correct the error.

144

PRS044: LOADED MODULE IS A CCT BUT CCT IS ALEADY PRESENT

Reason:

The SCB names a CCT, but the CCT address in the SCB is non-zero.

Action:

Correct the error.

148

PRS048: ZL DICTIONARY NOT ON A PAGE BOUNDARY

Reason:

ZL dictionaries must be linkedited with the PAGE statement to load on a page boundary.

Action:

Correct the error.

152

PRS052: RDT LRECL MUST BE SET FOR ZL COMPRESSION

Reason:

The loaded CCT is a ZL dictionary, but LRECL was not specified in the RDTGEN.

Action:

Correct the error.

156

PRS056: PRESSTO system error -- no message.

Reason:

A PRESSTO system error has occurred.

Action:

Contact CA Support.

160

PRS060: MODULE LOADED ABOVE 16M BUT PROGRAM IN AMODE 24

Reason:

Subroutines must be in AMODE 31 if RDT or CCT is link edited to load above the 16M line.

Action:

Correct the error.

164

PRS064: TARGET EXHAUSTED BEFORE SOURCE

Reason:

The cause of this error is one of the following:

- The *lcr* or *lur* parameter is invalid.
- The *cr* parameter cannot point to the beginning of a valid compressed record.
- The LRECL in the RDT is invalid.

Action:

Determine the cause of the error and correct the problem.

188

PRS088: EXPANSION OF A SYMBOL USED MORE THAN 127 DICTIONARY ENTRIES

Reason:

There was an invalid expansion dictionary.

Action:

Correct the error.

192

PRS092: EXPANSION DICTIONARY INVALID

Reason:

The expansion dictionary is invalid.

Action:

Fix the dictionary. If you believe that this message was obtained in error, contact CA Support.

Chapter 7: Utility I/O Messages

Note: Usually IBM system messages accompany I/O error messages.

3939

RECORD LENGTH CANNOT BE GREATER THAN 32767

Reason:

The record length is too long on //SYSUT1.

Action:

Correct the record length error on //SYSUT1.

3940

RECORD LENGTH CANNOT BE GREATER THAN 32767

Reason:

The record length is too long on //SYSUT2.

Action:

Correct the record length error on //SYSUT2.

3943

(no message)

Reason:

IBM issues a message indicating that //SYSUT1 cannot be opened.

Action:

Correct JCL and rerun.

3944

(no message)

Reason:

IBM issues a message indicating that //SYSUT2 cannot be opened.

Action:

Correct JCL and rerun.

3946

(no message)

Reason:

IBM issues a message indicating that //SYSUT1 or //SYSUT2 cannot be opened.

Action:

Correct JCL and rerun.

3948

PRESSTO: I/O ERROR

Reason:

See accompanying IBM message.

Action:

Correct JCL and rerun.

3949

PRESSTO: I/O ERROR

Reason:

See accompanying IBM message.

Action:

Correct JCL and rerun.

3950

PRESSTO: I/O ERROR

Reason:

See accompanying IBM message.

Action:

Correct JCL and rerun.

3951

PRESSTO: PDS HAS NO MEMBER NAME SPECIFIED

Reason:

A partitioned data set can be input to PRESSTO only if it has a member name.

Action:

Correct the error.

PREPASS Utility

none

WARNING: KEYED FILE HAS KEYLEN 0

none

WARNING: KEYED FILE BUT NO KEY IN RDT

none

WARNING: KEY WILL BE SCREAMBLED ON COMPRESSED FILE

none

WARNING: INPUT KEY NOT WITHIN RDT KEY DEFS

3960

LOAD FAILED FOR RDT

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.

3961

INVALID RDT LOADED. MAY BE MODULE WITH SAME NAME.

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.

3962

LENGTH IN RDT MISMATCHES LENGTH ON INPUT FILE

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.

3963

STAT-FILE INVALID. SEE 'USCNTS=Y' DESCRIPTION

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.

3965

//SYSUT3 REQUIRED UNLESS 'DEGREE=0' SPECIFIED

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.

3967

CONTROL CARD MUST BE ON //SYSIN

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.

3968

INVALID CHARACTER OR DELIMITER (flagged with an *)

3968

INVALID PARAMETER

3968

MULTIPLE RDT PARAMETERS

3968

NO RDTNAME

3968

RDTNAME LONGER THAN 8 CHARACTERS

3968

RDTNAME NOT A VALID SYMBOL

3968

MULTIPLE START PARAMETERS

3968

NUMERIC PARM MUST BE LESS THAN 16 DIGITS

3968

NUMERIC PARM TOO BIG

3968

MULTIPLE DEGREE PARAMETERS

3968

MULTIPLE NRECS PARAMETERS

3968

MULTIPLE SKIP PARAMETERS

3968

MULTIPLE USCNTS PARAMETERS

3968

DEGREE MUST BE 0 THRU 8

3968

NRECS + DEGREE CANNOT BE 0

3968

IF NRECS=0 THEN MUST SPECIFY USCNTS=Y

3968

NO RDT OR CCT PARAMETER

3968

RDT=...,CCT=... CANNOT BOTH BE SPECIFIED

Reason:

An error occurred as indicated in the 3968 message text.

Action:

Correct the error.

3970

(no message)

Reason:

IBM issues a message indicating that //SYSPRINT cannot be opened.

Action:

Correct JCL and rerun.

3972

NO STAT-FILE. REQUIRED IF USCNTS=Y. JOB TERMINATED.

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.

3973

UNEXPECTED EOF ON STAT-FILE. JOB TERMINATED.

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.

3974

CANNOT OPEN STAT-FILE ON //SYSUT2. JOB TERMINATED.

Reason:

An error occurred as indicated in the message text.

Action:

Correct the error.