IBM Tools Base for z/OS
Version 1  Release 5

IMS Hardware Data Compression
Extended User's Guide

IBM
# Contents

**Figures** ............................................ v

**Tables** ........................................... vii

**About this information** .......................... ix

Service updates and support information .......... ix

Highlighting conventions ............................. ix

How to read syntax diagrams ........................ x

How to send your comments ........................... xi

**Chapter 1. IMS HD Compression Extended overview** ................. 1

Database administration and change management solutions. .............. 1

How IMS HD Compression Extended works ....................... 3

IMS HD Compression Extended features and benefits ............... 3

Data compression and expansion ........................... 5

IMS HD Compression Extended components .................. 6

  Dictionary types .................................... 7

  IMS HD Compression Extended - functions and utilities ............ 9

IMS HD Compression Extended programs .................... 11

  Report types produced ................................ 12

IMS HD Compression Extended scenarios ................... 13

IMS HD Compression Extended documentation and updates. .......... 13

Accessibility features .................................. 14

**Chapter 2. Configuring IMS HD Compression Extended** ............. 15

**Chapter 3. Getting started** ................................ 17

IMS HD Compression Extended process roadmap .................. 17

Starting the IMS HD Compression Extended ISPF interface ........... 20

The IMS HD Compression Extended main menu ............... 20

**Chapter 4. Extracting data** ................................ 23

Image Copy Extract utility ................................ 23

  Extracting segments by using Image Copy Extract .................. 24

  Image Copy Extract panel ................................ 25

  Image Copy Extract JCL requirements ........................ 27

HD Unload Extract utility .................................. 30

  Extracting specific segments by using HD Unload Extract .......... 30

  HD Unload Extract panel ................................ 31

  HD Unload Extract JCL requirements ........................ 32

**Chapter 5. Analyzing segment data** ............................ 35

Analyzing segment compression by using Detail Analyze .......... 35

Detail Analyze JCL requirements ......................... 38

Detail Segment Compression Analysis report (HCO006) ............... 40

Segment Distribution by Compressed Length report (HCO007) .......... 41

**Chapter 6. Analyzing database data** ....................... 43

Analyzing segment or database compression by using Analyze ........ 43

Analyze panel ........................................ 45

Analyze JCL requirements ................................ 46

Compression Projection report (HCO001) ....................... 47

**Chapter 7. Building and relinking a dictionary** .......... 51

Building a custom dictionary by using Build .................... 51

Creating a compression exit routine ...................... 53

Modifying a DBD ...................................... 53

Relinking a compression exit routine ..................... 54

Build panel ........................................ 55

Compression Exit Routine panel ......................... 57

Edit DBD panel ....................................... 58

COMPRTN keyword ...................................... 58

Build JCL requirements ................................ 60

Compression exit routine and DBD JCL requirements ............. 61

Compression Projection report (HCO002) ....................... 62

**Chapter 8. Examining an existing dictionary** ................. 65

Examining a dictionary by using Examine .................... 65

Examine panel ........................................ 66

Examine JCL requirements ................................ 68

Compression Projection report (HCO003) ....................... 69

Compare Old/New Compression Projection report (HCO004) ......... 70

**Chapter 9. Compressing your data** ............................ 73

Compressing your data by using Compress .................... 73

Compress panel ........................................ 74

Compress JCL requirements ................................ 75

**Chapter 10. Messages, abends, and return codes** .............. 77

Messages ............................................. 77

Abend codes ......................................... 87

  Abend U3315 .................................... 88

  Abend U3325 .................................... 88

  Abends for HCOPMAnn ............................. 91

Return codes ......................................... 94

Gathering diagnostic information ....................... 95

**Notices** ............................................ 97

Programming interface information ..................... 98

© Copyright IBM Corp. 1996, 2014
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trademarks</td>
<td>99</td>
</tr>
<tr>
<td>Privacy policy considerations</td>
<td>99</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>101</td>
</tr>
</tbody>
</table>
Figures

1. Compression example ................................................. 6
2. Different dictionary types ......................................... 7
3. IMS HD Compression Extended process flow ................... 18
4. IMS HD Compression Extended main menu panel .............. 21
5. Image Copy Extract panel ........................................... 26
6. HD Unload Extract panel ........................................... 32
7. Detail Analyze panel .................................................. 38
8. Detail Segment Compression Analysis report .................... 40
9. Segment Size Distribution by Compressed Lengths report .... 42
10. Analyze panel .......................................................... 45
11. Example of the START BYTE option for an IMS Unload ....... 46
12. Example of the START BYTE option for a user-built record .. 46
13. Compression Projection report ....................................... 48
14. Build panel .............................................................. 56
15. Compression Exit Routine panel .................................... 57
16. Edit DBD panel ........................................................ 58
17. Compression Projection report ....................................... 63
18. Examine panel .......................................................... 67
19. Compression Projection report ....................................... 69
20. Compare Old/New Compression Projection report ............. 71
21. Compress panel ........................................................ 74
### Tables

1. IMS HD Compression Extended functions and utilities ........................................ 10
2. IMS HD Compression Extended programs .......................................................... 11
3. IMS HD Compression Extended report types ...................................................... 12
4. IMS HD Compression Extended phases ............................................................. 18
5. Functions that can be used to access the Image Copy Extract utility ................. 24
About this information

IBM® IMS™ Hardware Data Compression Extended for z/OS® (also referred to as IMS HD Compression Extended) provides a set of tools designed to help implement and manage compressed IMS databases.

These topics provide instructions for installing, configuring, and using IMS Hardware Data Compression Extended.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform these tasks:
- Plan for the installation of IMS HD Compression Extended
- Install and operate IMS HD Compression Extended
- Customize your IMS HD Compression Extended environment
- Diagnose and recover from IMS HD Compression Extended problems
- Use IMS HD Compression Extended with other IMS products

To use these topics, you should have a working knowledge of:
- The z/OS operating system
- ISPF
- SMP/E

Specific changes since the previous edition of this book are indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

Always check the IMS Tools Library page for the most current version of this publication:


Service updates and support information

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:


Highlighting conventions

This information uses the following highlighting conventions:
- **Boldface** type indicates commands or user interface controls such as names of fields, folders, icons, or menu choices.
- **Monospace** type indicates examples of text that you enter exactly as shown.
- *Italic* type indicates variables that you should replace with a value, to indicate the titles of other publication, and to emphasize significant terms.
How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The `>>>` symbol indicates the beginning of a syntax diagram.
  - The `--->` symbol indicates that the syntax diagram is continued on the next line.
  - The `>---` symbol indicates that a syntax diagram is continued from the previous line.
  - The `---<` symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).
  
  `>>>required_item<---`

- Optional items appear below the main path.
  
  `>>>required_item<--optional_item<---`

If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

  `>>>required_item<--optional_item<---`

- If you can choose from two or more items, they appear vertically, in a stack.
  If you must choose one of the items, one item of the stack appears on the main path.
  
  `>>>required_item<--required_choice1--required_choice2<---`

If choosing one of the items is optional, the entire stack appears below the main path.

  `>>>required_item<--optional_choice1<--optional_choice2<---`

If one of the items is the default, it appears above the main path, and the remaining choices are shown below.

  `>>>required_item<--default_choice<--optional_choice<--optional_choice<---`

- An arrow returning to the left, above the main line, indicates an item that can be repeated.
If the repeat arrow contains a comma, you must separate repeated items with a comma.

A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, column-name). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses, for example (1).

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IMS HD Compression Extended documentation, use either of the following options:

- Use the online reader comment form, which is located at: www.ibm.com/software/data/rcf/
- Send your comments by e-mail to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of IMS HD Compression Extended, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
Chapter 1. IMS HD Compression Extended overview

IMS HD Compression Extended provides functions for compressing IMS data. IMS HD Compression Extended uses the z/OS hardware data compression (HDC) that is available on IBM processors.

IMS provides a basic utility that supports z/OS HDC. IMS HD Compression Extended provides enhanced functions that supports z/OS HDC that are not available in the basic IMS utility. IMS HD Compression Extended can be used to compress HDAM, HIDAM, HISAM, and Fast Path databases.

The advantages of compression include:
- Reduced DASD storage requirements for databases
- Reduced database I/O
- CPU usage is not significantly increased

Compressing data conserves DASD and tape resources. It also reduces high RBA usage, which is especially beneficial to IMS databases that are approaching the 4 GB limit for VSAM or 8 GB limit for OSAM.

Compression can improve database response time and performance because fewer physical I/Os are needed to access data.

Topics:
- “Database administration and change management solutions”
- “How IMS HD Compression Extended works” on page 3
- “IMS HD Compression Extended features and benefits” on page 3
- “Data compression and expansion” on page 5
- “IMS HD Compression Extended components” on page 6
  - “Dictionary types” on page 7
  - “IMS HD Compression Extended - functions and utilities” on page 9
  - “IMS HD Compression Extended programs” on page 11
  - “Report types produced” on page 12
- “IMS HD Compression Extended scenarios” on page 13
- “IMS HD Compression Extended documentation and updates” on page 13
- “Accessibility features” on page 14
- Summary of changes

Database administration and change management solutions

IBM solutions help IT organizations maximize their investment in IMS databases while staying on top of some of today’s toughest IT challenges. Database Administration and Change Management solutions are the core responsibilities of the DBA.
If not managed correctly, they can monopolize data center resources, waste valuable time, and can result in errors. Managing the database administration and change management process requires database administrators to address the following questions:

- How can I ensure that I complete all the necessary steps when making changes?
- What is the best way to propagate changes to other database environments?
- How can I keep IMS software versions current?
- How do I manage a corrupted database?
- What is the most efficient way to convert IMS full-function databases to the new High Availability Large Database (HALDB) format?

Many IMS Tools products can help reduce the negative impact data changes can have on your database.

IMS HD Compression Extended is only one of several IMS Tools products that provide enhancements to the process of managing database operations.

**IMS HD Compression Extended and other IMS Tools**

Many IMS tools provide database management features that are not available in IMS itself or provide enhancements to capabilities built into IMS.

IMS HD Compression Extended is only one of several IMS tools that provide enhancements to the process of database administration and change management.

IMS Tools Base V1.1 for z/OS includes the following tools:

- IMS HD Compression Extended
- IMS Tools Knowledge Base
- IMS Tools Generic Exits
- Policy Services
- Distribute Access Infrastructure

Other IMS Tools products that can assist with database administration and change management include:

- IMS HALDB Toolkit
- IMS Library Integrity Utilities
- IMS Online Reorganization Facility
- IMS Parameter Manager
- IMS Sequential Randomizer Generator
- IMS Sysplex Manager
- IMS Tools Knowledge Base
- IMS High Performance Image Copy for z/OS

IMS HD Compression Extended is a vital part of managing your IMS database system.
How IMS HD Compression Extended works

HD Compression Extended provides a set of tools that are designed to help implement and manage compressed IMS databases.

IMS HD Compression Extended provides:
- An easy-to-use approach for creating and maintaining compressed IMS databases
- Features to evaluate compression impact, build, verify, and archive dictionaries, implement compression, and monitor and maintain production compression routines
- Ability to use generic, enterprise, or custom compression dictionaries
- An easy-to-use ISPF interface

Segment data is expanded and compressed by the compression exit routine whenever a segment is accessed.

The compression routine contains two dictionaries, one for compression and one for expansion (collectively referred to as the dictionary).

An advantage of using IMS HD Compression Extended is that it does not depend on IMS being up, does not require that you schedule around IMS, and does not interfere with IMS. A dictionary can be built with representative data or a subset of the actual data. The data does not have to be current; the backed up data on an IMS image copy or HD unload data set can work as well as segment data that is extracted by one of your own applications.

IMS HD Compression Extended features and benefits

You can save DASD and I/O resources while minimizing CPU usage by taking advantage of IBM Hardware Compression services.

Compression can improve database response time and performance, because fewer physical I/Os are needed to access data.

Because IMS HD Compression Extended uses hardware-assisted compression, accessing compressed data from an application does not significantly increase CPU usage. CPU usage is greatest when compression takes place, for example during reload (writing data). CPU usage is negligible when unloading compressed segments (reading data). Any added CPU usage is based entirely on the amount of compression and expansion required when an application processes the database. A minimal increase in CPU usage when reading is typical, but is offset by the reduced physical I/O.

IMS HD Compression Extended:
- Analyzes what compression savings are possible before creating a dictionary
- Determines what compression savings the generic dictionary supplied with the product could provide
- Determines what compression savings an existing custom dictionary could provide if used on a database other than the one from which it was created
- Evaluates compression savings by segment and provides a distribution of compressed segment lengths (to help you prepare a DBD for reloading a database and minimizing VSAM control interval (CI) splits)
• Creates a dictionary, compression exit routine, and reload DBD in preparation for implementing data compression
• Provides ISPF panels for all functions

IMS HD Compression Extended provides a platform to evaluate, build, verify, archive, implement, and manage the performance of compressed databases. Compressing databases reduces storage and I/O requirements. Hardware compression helps minimize the CPU usage associated with compression. HD Compression Extended is designed around the activities needed to develop compression dictionaries, implement compression, and monitor compression performance. It provides an easy to use frontend to interactively operate these activities. IMS HD Compression Extended:
• Provides an Analyze function that simulates a compression dictionary and reports on its effectiveness. Multiple input data sources are supported for sampling and comparison. Results can be evaluated easily to select the most effective dictionary for production use.
• Provides a Build function to build, verify, and archive a compression dictionary. Linking the dictionary to the base exit routine creates the exit needed for implementation and updates the DBDGEN source.
• Provides a Compress function to implement the compression service. JCL is built to reload the database with the new or revised compression exits.
• Provides an Examine function to evaluate various combinations of dictionaries and databases, and monitors installed compression performance as databases change over time. It also can evaluate the effectiveness of the generic dictionary or an enterprise dictionary when used on a database other than the one from which the source data was obtained.
• Supports compression of:
  – IMS High Availability Databases (HALDB) – PHDAM and PHIDAM
  – Full function databases - HDAM, HIDAM, HISAM
  – Fast Path databases - DEDB
• Uses the z/OS hardware data compression (HDC) available on IBM processors and defaults to software implementation if hardware services are not available.
• Functions are ISPF menu-driven.

**IMS HD Compression Extended and other IMS Tools**

Many IMS tools provide database management features that are not available in IMS itself or provide enhancements to capabilities built into IMS.

IMS HD Compression Extended is only one of several IMS tools that provide enhancements to the process of database administration and change management.

Other IMS Tools products that can assist with database administration and change management include:
• IMS HALDB Toolkit
• IMS Library Integrity Utilities
• IMS Online Reorganization Facility
• IMS Parameter Manager
• IMS Sequential Randomizer Generator
• IMS Sysplex Manager
• IMS Tools Knowledge Base
IMS HD Compression Extended is a vital part of managing your IMS database system.

**Data compression and expansion**

A general understanding of how data compression and expansion works helps you use IMS HD Compression Extended.

Hardware data compression uses the Ziv-Lempel compression technique, which uses a fixed number of bits to replace a variable number of bytes. This technique requires use of a dictionary. The dictionary does the translation between uncompressed and compressed data. Dictionaries compress and expand data as follows:

- **Compression** takes an input string of data and, using the compression dictionary, produces an output string of compression symbols. The symbols represent a fixed number of bits that replace the variable number of bytes that were input.
- **Expansion** takes an input string of compression symbols and, using the expansion dictionary, produces an output string of data.

A separate dictionary exists for compression symbols and expansion characters.

The dictionary-based compression exit routine created by IMS HD Compression Extended can compress the data of individual IMS segments or entire databases. The compression exit routine can compress segments and databases other than those from which the dictionary was originally created. This allows flexibility in the use of compression exit routines. The amount of space saved by compression depends on how well the dictionary and the data that is being compressed correlate. Due to the finite size of the dictionary (64 KB), it is generally not possible for a single dictionary to hold all the entries necessary to produce maximum compression of all IMS databases. You can use, multiple dictionaries to compress and expand the different types of data in your IMS databases.

An IMS database must be unloaded and reloaded as part of the compression process. The data is expanded when unloaded and compressed by IMS at reload using the compression exit routine.

Compression and expansion are done by using the compression exit routine. The compression exit routine, which is specified in the DBD, is created by linking together a dictionary driver and a dictionary object data set. You can use the Build function to create the compression exit routine, including a dictionary driver.

IMS gives the compression exit routine control whenever a segment that requires compression or expansion is accessed. As shown in the following figure, the phrase "this is data" is compressed from 12 bytes down to 5 bytes.
When you build a custom dictionary, as each input string of data is read, a Ziv-Lempel tree begins to develop arranging parts of the data as dictionary entries. The total size of the dictionary is 64 KB because each entry is 16 bytes (8 for compression and 8 for expansion). If sufficient data is supplied to build a dictionary, 4096 entries are included in the dictionary (2048 for compression and 2048 for expansion). For more information about Ziv-Lempel dictionaries, see Enterprise Systems Architecture/390 Data Compression and Support for z/OS/ESA Compression Services.

IMS HD Compression Extended components

IMS HD Compression Extended consists of several components.

The main components of IMS HD Compression Extended include:

Dictionary
The dictionary tells IMS how to compress and uncompress data as it moves into and out of its location in storage. A good dictionary is an essential part of getting optimal compression on your systems.

An IMS HD Compression Extended dictionary exists as both an:

z/OS data set
The z/OS data set is used by the Examine function to report on the effectiveness of a dictionary.

Object data set
The object data set is used by IMS as part of its compression exit routine.

Functions and utilities
The functions and utilities that come with IMS HD Compression Extended are used to analyze your data and segments, build dictionaries, and compress your data.

Programs
The programs that come with IMS HD Compression Extended support the functions and utilities that are used to analyze your data and segments, build dictionaries, and compress your data.
Reports

The reports that are generated from IMS HD Compression Extended help you analyze what level of storage savings your current dictionary could achieve for you. They also project potential compression savings, and can be used to audit extract information.

Dictionary types

Depending on your data and the amount of compression you need, you can select from four different dictionary types.

As shown in the following figure, IMS HD Compression Extended supports four types of dictionaries to help you find the best possible balance between DASD savings and dictionary management.

- Generic dictionary (provided with IMS HD Compression Extended)
- Enterprise dictionary
- Customized database dictionary
- Customized segment dictionary

The last three dictionaries are custom dictionaries that you create by using the Build function. The three dictionaries are not really different types of dictionaries, but rather different dictionary strategies. They are referred to as custom dictionaries.

Figure 2. Different dictionary types
Generic dictionary

A generic dictionary is provided with IMS HD Compression Extended. The generic dictionary is developed from English-text data that contain frequently used character combinations. Typically the generic dictionary can help you achieve 40 - 60 percent compression.

The generic dictionary provides a single dictionary for compressing all of your data. The generic dictionary resides on the hcohdq.HCOGDIIC1.HDCDDICT data set that you installed.

Use the Examine function to see how much space you could save by using the generic dictionary.

If using the generic dictionary produces less than a 40 percent savings in space or if the database does not contain primarily English text, consider using one of the other dictionary strategies.

Enterprise dictionary

An enterprise dictionary is created by you from your IMS data by using the Build function. Enterprise dictionaries typically define one dictionary per data type.

Selecting sample data for input to the Build function is a part of creating an enterprise dictionary. The sample data is the basis for the entries in the dictionary. The idea of an enterprise dictionary strategy is to create a customized dictionary for data in multiple databases, so you need to identify a representative set of data from the databases across your enterprise. For example, English-text data can be thought of as one group, alphanumeric data (such as names and addresses) as another, and numeric data a third. For each group of data, you can create an enterprise dictionary.

Before you build an enterprise dictionary, choose one database from a group, then use an appropriate program to create a variable-blocked QSAM data set for that group. Repeat this process for each group. You create a dictionary for a group by using one of these QSAM data sets as input to the Build function. Only one dictionary can be created at a time, so run the Build function for each QSAM data set to create a dictionary for each group.

If you want fewer enterprise dictionaries, you can create a single one. However, more effort is needed to create a single enterprise dictionary than multiple ones because a QSAM data set has to be built by using sample data from each database so that the data set holds a cross-section of your database data. The amount of data that is typically needed to build a dictionary is 300-400 KB.

Customized database dictionary

A customized database dictionary can provide better compression than the generic or enterprise dictionary, but might require you to build more dictionaries.

A customized database dictionary is intended to compress only one database. A customized dictionary is created from the data in the database it is to compress. The dictionary entries are, therefore, oriented to data that is unique to the database.
Even though the object of this strategy is to have a separate dictionary for each database, you can still use a customized database dictionary to compress more than one database. If data in one database is like data in another or another group of databases, the customized database dictionary can serve as an enterprise dictionary. The ability to share dictionaries eliminates the need to have a dictionary for each database, yet provides almost customized compression.

**Customized segment dictionary**

A customized segment dictionary compresses one segment, or a group of segments, from a database. A dictionary is built for each segment or group of segments.

The cumulative effect of compressing each segment by using its own dictionary can achieve maximum compression savings. Implementation of a customized segment dictionary strategy generally requires that more dictionaries be built. Because each database can contain up to 255 segments, you can potentially need 255 dictionaries for a single database.

The segments that benefit most from a customized segment dictionary are those segments that use the most space or are read the most often, especially by sequential batch jobs. Compressing these segments can save I/O as well.

To evaluate which segments are good candidates for compression, use the Detail Analyze function. The Detail Analyze function shows the potential savings for each segment in a database without requiring you to actually build a permanent dictionary. You can then determine which segments need their own dictionaries built.

**IMS HD Compression Extended - functions and utilities**

IMS HD Compression Extended includes several functions that help you evaluate, build, verify, archive, implement, and manage the performance of compressed databases.

Effective database compression is achieved through knowledge of the data, the databases, and the effect of your dictionary type. IMS HD Compression Extended provides the functions and reports for analyzing various compression scenarios, extracting the sample data that feeds this process, building the dictionaries, and building the compression user exit routine.

The functions help you answer these questions:

- What dictionary strategy should I employ?
- Will compression using the IBM-supplied generic dictionary be satisfactory?
- Should I build an enterprise-wide dictionary based on my own database data?
- Should I compress each database by using its own customized database dictionary?
- Will it be worth the effort to build a customized segment dictionary for each segment of each database?

The following table lists and provides a brief description for each of the IMS HD Compression Extended functions and utilities.
Table 1. IMS HD Compression Extended functions and utilities

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>To obtain the data that is used in the other IMS HD Compression Extended functions, use the Extract function. The Extract function consists of two utilities:</td>
</tr>
<tr>
<td></td>
<td><strong>Image Copy Extract utility</strong></td>
</tr>
<tr>
<td></td>
<td>The Image Copy Extract utility creates a data set of segment records. This data set might be a selection of one or more segments or it might be all segments from a database. This image copy extract data set can be operated on by all the IMS HD Compression Extended functions:</td>
</tr>
<tr>
<td></td>
<td>• in its entirety</td>
</tr>
<tr>
<td></td>
<td>• one segment at a time</td>
</tr>
<tr>
<td></td>
<td><strong>HD Unload Extract utility</strong></td>
</tr>
<tr>
<td></td>
<td>This utility is used only by the Detail Analyze function. Any data set to be operated on by the Detail Analyze function must be in HD unload format. If you do not have an HD unload data set, this utility can convert an image copy extract data set into HD unload format so that it can be operated on by the Detail Analyze function. The HD Unload Extract utility splits the HD unload data set into multiple data sets. Each data set contains one segment. These data sets are processed all in one pass of the Detail Analyze function. Alternatively, each data set could be operated on one at a time by the Analyze, Build, and Examine functions.</td>
</tr>
<tr>
<td>Examine</td>
<td>The Examine function reports the estimated percent of savings for sample data by using an existing dictionary. The Examine function can help you:</td>
</tr>
<tr>
<td></td>
<td>• Determine whether the generic dictionary provides adequate compression savings. The Examine function compresses segments or databases using the generic dictionary.</td>
</tr>
<tr>
<td></td>
<td>• Determine the ongoing effectiveness of a custom dictionary, or determine if a custom dictionary that was built for one segment or database can be effectively used to compress another segment or database. The old estimates from when the dictionary was built and the new estimates from compressing a different set of data are reported.</td>
</tr>
<tr>
<td>Analyze</td>
<td>The Analyze function builds a temporary dictionary from your sample data and then rereads your data to compress it. The Analyze function produces a single report projecting your compression savings for the sample data based on a custom dictionary. The Analyze function is limited to treating one entire data set or one selected segment at a time.</td>
</tr>
</tbody>
</table>
Table 1. IMS HD Compression Extended functions and utilities (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail Analyze</td>
<td>The Detail Analyze function builds a temporary dictionary from your sample data, rereads your data to compress it, and then produces two reports. The Detail Analyze function can analyze multiple segments in a single pass. It provides statistics for both generic and custom dictionaries. Reports from the Detail Analyze function show: • Compression savings for each segment • Distributing lengths of the data portion of the segments after compression Use the Detail Analyze function to determine maximum segment length before building the DBD and when minimizing VSAM control interval (CI) splits.</td>
</tr>
<tr>
<td>Build</td>
<td>The Build function creates a single dictionary for whichever segment or database is selected and reports the estimated savings on the compressed part of the segment data. It also creates the load module that becomes the compression exit routine for that DBD (and any other DBDs your strategy calls for). After you determine the segments or databases for which you want to use customized compression, you use the Build function. From IMS’s perspective, the compression exit routine is an IMS Segment Edit/Compression exit routine.</td>
</tr>
<tr>
<td>Compress</td>
<td>The Compress function creates reload JCL and guides you through the reload process. After you create a dictionary and an IMS compression exit routine by using the Build function, you are ready to compress your data. Compression is done when you reload your database.</td>
</tr>
</tbody>
</table>

z/OS and object data sets can also be used as input to the Analyze, Build, and Examine functions and can be processed as a whole.

**IMS HD Compression Extended programs**

IMS HD Compression Extended consists of several programs and two dictionary driver modules.

The following table lists the program names and brief descriptions of the functions, utilities, and drivers included in IMS HD Compression Extended.

Table 2. IMS HD Compression Extended programs

<table>
<thead>
<tr>
<th>Program name</th>
<th>Function or utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCOPMA&lt;version&gt;</td>
<td>Image Copy Extract utility</td>
<td>The IMS release-dependent Image Copy Extract utility programs. The last two characters indicate what IMS release they support. Use only the program that is appropriate for your IMS release. These programs create a data set that contains the segments that you select for extraction from an IMS image copy data set.</td>
</tr>
</tbody>
</table>
Table 2. IMS HD Compression Extended programs (continued)

<table>
<thead>
<tr>
<th>Program name</th>
<th>Function or utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCOZLDH0</td>
<td>HD Unload</td>
<td>Extract utility, creates a separate data set for each segment you select for extraction from an IMS HD unload data set or IMS image copy data set.</td>
</tr>
<tr>
<td>HCOZLDE0</td>
<td>Detail Analyze</td>
<td>Function produces two reports.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- &quot;Detail Segment Compression Analysis report (HCO006)&quot; on page 40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- &quot;Segment Distribution by Compressed Length report (HCO007)&quot; on page 41</td>
</tr>
<tr>
<td>HCOZLDU0</td>
<td>Analyze, Build, and Examine functions</td>
<td>Creates a dictionary z/OS format data set and reports.</td>
</tr>
<tr>
<td>HCOZLDX0 and HCOZLDXV</td>
<td>dictionary drivers</td>
<td>HCOZLDX0 and HCOZLDXV must be linked to the dictionary object data set to create the IMS compression exit routine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Use HCOZLDX0 for full-function databases.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Use HCOZLDXV for variable-length Fast Path DEDB segments when padding is needed to force segments to the minimum segment length after compression is done.</td>
</tr>
</tbody>
</table>

The HCOPMAxx programs run under the IMS batch region controller. HCOZLDH0, HCOZLDE0, and HCOZLDU0 run as standard z/OS batch jobs. The dictionary driver runs when an IMS application requests access to a segment that references the compression exit routine.

**Report types produced**

IMS HD Compression Extended produces three types of reports.

The following table lists and describes the three types of reports produced by IMS HD Compression Extended. The Segment Compression Analysis and Compression Projection reports can help you determine the best methods and settings for compression of your data. The auditing reports can be used to track where data has come from.

Table 3. IMS HD Compression Extended report types

<table>
<thead>
<tr>
<th>Report type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Compression Analysis</td>
<td>These reports show, on a per segment basis, kilobytes used before and after compression. Compression savings are shown in kilobytes and as a percent of the original space before compression. A distribution of segment lengths after compression is also provided.</td>
</tr>
<tr>
<td>Report type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Compression Projection</td>
<td>These reports show minimum, maximum, and average segment size before and after compression and the average amount of space saved per segment. The Build, Analyze, and Examine functions each produce a version of this report.</td>
</tr>
<tr>
<td>Auditing</td>
<td>Created by the Extract functions.</td>
</tr>
</tbody>
</table>

**IMS HD Compression Extended scenarios**

You can use IMS HD Compression Extended to address many of your day-to-day business problems.

The following scenario illustrates how you can use IMS HD Compression Extended to address typical business problems.

**Saving storage space**

Your business has grown rapidly, and your data volume has become difficult to manage. To maximize your company’s data storage strategies and to avoid having to purchase more computers, you decide to use IMS HD Compression Extended.

By compressing data, you can free up over 50% of your current systems storage capacity.

**Reaching the IMS storage limits**

If your IMS databases are approaching the 4 GB VSAM or 8 GB limit for OSAM and you do not want to convert to HALDB, use IMS HD Compression Extended to achieve enough data storage savings to continue using your existing systems. By compressing your data instead of purchasing more hardware, you save money and keep maintenance requirements lower. Fewer databases to monitor and maintain decreases the burden on database administrators and IT staff.

**IMS HD Compression Extended documentation and updates**

This topic explains where to find DB2® and IMS Tools information on the Web, and explains how to receive information updates automatically.

**IMS HD Compression Extended information on the Web**

The IMS Tools Product publications web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following web page:


You can also access documentation for many IMS Tools from IBM Knowledge Center:

Search for a specific IMS Tool product or browse the Information Management > IMS family.

IBM Redbooks® publications that cover DB2 and IMS Tools are available from the following Web page:


The Data Management Tools Solutions Web site shows how IBM solutions can help IT organizations maximize their investment in DB2 and IMS databases while staying ahead of today's top data management challenges:


Receiving documentation updates automatically

To automatically receive a weekly e-mail that notifies you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Support service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Support service:
1. Go to http://www.ibm.com/support/mysupport
2. Enter your IBM ID and password, or create one by clicking register now.
3. When the My Support page is displayed, click add products to select those products that you want to receive information updates about. The DB2 and IMS Tools categories are located under Software > Data and Information Management > Database Tools & Utilities.
4. Click Subscribe to e-mail to specify the types of updates that you would like to receive.
5. Click Update to save your profile.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in IMS HD Compression Extended enable users to:
- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
  - z/OS ISPF User's Guide, Volume 1
  - z/OS TSO/E Primer
  - z/OS TSO/E User's Guide

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.
Chapter 2. Configuring IMS HD Compression Extended

Information about configuring IMS HD Compression Extended and other Tools Base components for IMS is provided in IBM Tools Base for z/OS Configuration for IMS.

You can also download a PDF version of this information from the IMS Tools Product Documentation page.

For information about migrating an existing installation of IMS HD Compression Extended after upgrading to a new release of IBM Tools Base for z/OS, see “Relinking a compression exit routine” on page 54.
Chapter 3. Getting started

To start using IMS HD Compression Extended, you need to learn how to start the product, how to use its ISPF interface, and how to complete the processes that are associated with the five phases of compressing data by using IMS HD Compression Extended.

Topics:
- “IMS HD Compression Extended process roadmap”
- “Starting the IMS HD Compression Extended ISPF interface” on page 20
- “The IMS HD Compression Extended main menu” on page 20

IMS HD Compression Extended process roadmap

You need to progress through five main phases when using IMS HD Compression Extended. The information in this topic provides a quick reference to those phases, the processes you need to run, and links to the topics that explain how to use the function or utility to complete the task.

The following diagram illustrates the online reorganization process flow for an IMS HD Compression Extended environment.
Table 4. IMS HD Compression Extended phases

<table>
<thead>
<tr>
<th>Phase</th>
<th>Instructions</th>
<th>Link to topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decide on a dictionary type</td>
<td>Decide on the type of dictionary that you want to use. If you are not sure, you can use the generic dictionary to test the effectiveness of data compression.</td>
<td>“Dictionary types” on page 7</td>
</tr>
</tbody>
</table>
Table 4. IMS HD Compression Extended phases (continued)

<table>
<thead>
<tr>
<th>Phase</th>
<th>Instructions</th>
<th>Link to topic</th>
</tr>
</thead>
</table>
| 1. Analyze your data         | Analyze your data to determine which segments to compress. If most of your data uses plain English text and 50% compression is acceptable, use the generic dictionary. Use the Examine function to access either:  
  - “Image Copy Extract utility” on page 23  
  - “HD Unload Extract utility” on page 30  
  If you want a high percentage of compression or have non-English text in the database, use a custom dictionary. Either:  
  - “Analyzing segment compression by using Detail Analyze” on page 35  
  - “Analyzing segment or database compression by using Analyze” on page 43 |                                                                                                                                                                               |
| 2. Build your dictionary     | If you are using a custom dictionary strategy, then create your custom dictionary using the Build function.  
  Create the compression exit routine.  
  Create new DBD.  
  This compression exit routine is invoked just like any other IMS Segment Edit/Compression exit routine. For example, review the invocation of DFSCMPFX0, which is described in IMS Customization Guide: Database.  
  The exit routine name is specified in the DBD SEGM statement COMPRTN parameter.  
  “Building a custom dictionary by using Build” on page 51  
  “Creating a compression exit routine” on page 53  
  “Modifying a DBD” on page 53 |                                                                                                                                                                               |
| 3. Unload the database       | Unload the database using an appropriate IMS tool and the old DBD.  
  An IMS database must be unloaded and reloaded as part of the compression process. The data is expanded when it is unloaded and compressed by IMS at reload using the compression exit routine. Use IMS tools provided by IBM or other IMS tools provider. |                                                                                                                                                                               |
| 4. Run DBDGEN and ACBGEN for new DBD | Run DBDGEN and ACBGEN for new DBD. Use IMS tools provided by IBM or other IMS tools provider. |                                                                                                                                                                               |
### Starting the IMS HD Compression Extended ISPF interface

You start the IMS HD Compression Extended ISPF interface by issuing a TSO command. The command that you enter depends on whether the HCOC00 REXX procedure is installed in your executable CLIST library.

**About this task**

Use the appropriate command to start IMS HD Compression Extended.

**Procedure**

1. Log on to TSO and specify a minimum REGION of (6144).
2. Select ISPF option 6, and issue the following command on the TSO command line.
   - If the REXX procedure is installed in your executable CLIST library, enter:
     ```
     HCOC00 'hlq'
     ```
   - If REXX procedure is not installed in your executable CLIST library, enter:
     ```
     EX 'hlq,SHCOCXE(HCOC00)' 'hlq'
     ```
   Where hlq is the data set high-level qualifier to install IMS HD Compression Extended.
   When IMS HD Compression Extended starts, the main menu opens after the copyright panel.

### The IMS HD Compression Extended main menu

You can access all basic functions of the product from the IMS HD Compression Extended main menu panel except Extract.

The following figure shows the IMS HD Compression Extended main menu panel. The Extract function is a subfunction and is accessed from the Analyze, Build, Detail Analyze, and Examine functions.
The options on this panel are:

**Detail Analyze**
Select this function to generate compression statistics at the segment level for each segment that you specify. This process involves:
- Extracting the segments to analyze
- Generating JCL for the Detail Analyze function
- Editing the completed JCL and then submitting a job

**Analyze**
Select this function to generate compression projections at the segment database level. This process involves:
- Extracting the segments to analyze
- Generating JCL for the Analyze function
- Editing the completed JCL and then submitting the job

**Build**
Select this function to build a dictionary, an IMS compression exit routine, and the appropriate DBD. This process involves:
- Extracting the segments to build a dictionary
- Generating JCL for the Build function
- Generating dictionary link-edit JCL
- Editing the DBD source
- Generating DBDGEN JCL
- Editing the completed JCL and then submitting the job

**Examine**
Select this function to examine an existing or generic dictionary’s effectiveness with current data or data from a different database. This process involves:
- Extracting the segments to examine
- Generating JCL for the Examine function
- Editing the completed JCL and then submitting the job

**Compress**
Select this function to reload a database using the new DBD (data is compressed during the reload). This process involves:
- Generating database reload JCL
- Editing the completed JCL and then submitting the job
Chapter 4. Extracting data

To generate compression statistics or build a dictionary, you must extract segment data into a variable-blocked QSAM data set. The Extract function is accessed from the Detail Analyze, Analyze, Build, and Examine functions, and it is the first step in the compression process.

IMS HD Compression Extended provides two utilities to help you extract data:

**Image Copy Extract utility**
Extracts specified segments from an IMS image copy data set.

**HD Unload Extract utility**
Extracts specified segments from an IMS HD unload data set or, in some cases, from the Image Copy Extract utility output.

To separate the segments that you have extracted into individual data sets, use the HD Unload Extract utility.

The output from these utilities is one or more QSAM data sets that contain extracted segments that can be used for creating a dictionary or analyzing compression.

**Topics:**
- “Image Copy Extract utility”
  - “Extracting segments by using Image Copy Extract” on page 24
  - “Image Copy Extract panel” on page 25
  - “Image Copy Extract JCL requirements” on page 27
- “HD Unload Extract utility” on page 30
  - “Extracting specific segments by using HD Unload Extract” on page 30
  - “HD Unload Extract panel” on page 31
  - “HD Unload Extract JCL requirements” on page 32

**Image Copy Extract utility**

The Image Copy Extract utility extracts specified segments from an IMS image copy data set to produce QSAM data sets and to customize the QSAM output.

The Image Copy Extract utility does two things:
- It produces a variable-blocked QSAM data set from an IMS image copy data set. Only QSAM data sets can be used as input to IMS HD Compression Extended functions.
- It customizes the output QSAM data set so that it contains only the segments that you want to extract.

The Image Copy Extract utility extracts segment data. To access the Image Copy Extract utility, you can use several of the IMS HD Compression Extended functions. The following table lists the functions that can access the Image Copy Extract utility.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail Analyze</td>
<td>Use the Detail Analyze function to extract all segments from the IMS image copy data set.</td>
</tr>
<tr>
<td>Analyze, Build, or Examine</td>
<td>Use these functions to extract all the segments that you want.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Important:** The Image Copy Extract utility does not work for HALDB Image Copies. You must use the IMS Unload Image Copy utility.

**Extracting segments by using Image Copy Extract**

The Image Copy Extract function extracts specified segments from an IMS image copy data set.

**Before you begin**

**Restrictions:**

- An IMS image copy data set cannot be used directly as input to IMS HD Compression Extended functions. It can be used only as input to the Image Copy Extract utility.
- Only one IMS image copy data set can be processed in a single execution of the utility. Each data set that belongs to an IMS database image copy must be run separately (except for HISAM overflow data sets).
- DEEB image copy data sets are not supported by the utility.
- Compression exits that are invoked from the utility must be in 31-bit mode.
- Only one data set of extracted segments is produced in a single execution of the utility.

You can use either of the following methods to run the Image Copy Extract utility:

- From the ISPF panel-driven functions Detail Analyze, Analyze, and Examine
- Directly from JCL that is run under the IMS batch region controller

**About this task**

The following procedure describes how to run the utility from the ISPF panels.

**Procedure**

1. Generate a PSB (if necessary) for the utility to access the IMS image copy data set.
2. Determine the names of the segments that you want to extract (unless you are extracting all segments).
3. From the ISPF panel of the Detail Analyze, Analyze, or Examine function, enter Y (yes) in the IMAGE COPY EXTRACT field.
4. Select segments for extraction. For specific segments, specify a segment name; for all segments, leave the segment name blank.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise or customized database dictionary</td>
<td>Select all segments from the image copy. Selecting all segments is the default setting.</td>
</tr>
<tr>
<td>Customized segment dictionary and only one segment</td>
<td>Specify the segment. Because the Image Copy Extract utility produces only one extract data set, that specifies multiple segments produces a single data set of combined segments. A combined segment data set is useful if you plan to compress multiple segments. A single dictionary can then be used to compress several similar segments.</td>
</tr>
<tr>
<td>Separate data sets for each segment</td>
<td>Begin with the Detail Analyze function and indicate which segments you want the Image Copy Extract utility to extract. If you want all segments extracted (the default), do not specify a segment name. The Detail Analyze function invokes the HD Unload Extract utility to separate segments into separate data sets from the Image Copy Extract utility output.</td>
</tr>
</tbody>
</table>
The Image Copy Extract panel contains the following fields:

**DBD NAME**
Specify the name of the database DBD. This information is added to the PROCCTL DD control statement. See "Image Copy Extract JCL requirements" on page 27.

**DATABASE DD NAME**
Specify the database DD name, as specified in the DBD. If an overflow data set exists, you must add the overflow data set name to the JCL and the DD name to the PROCCTL control statement parameters. The generated JCL is returned in ISPF edit mode for final changes after this panel is processed. For more information about these parameters, see "Image Copy Extract JCL requirements" on page 27.

**SEGMENT NAME**
Specify the name of the segment that you want to extract from the IMS image copy data set. Specify this field only if you want segments extracted. The segment name is added to the SEGIN DD control statements that are generated in the JCL. If you want to extract multiple segments, add them to the SEGIN DD entries when the JCL is returned for editing.

If you leave this field blank, all segments are extracted from the IMS image copy data set (the default).

**IMS RELEASE**
Specify the 2-digit IMS release ID (for example, 61 for IMS 6.1 or 71 for IMS 7.1).

**EXTRACT UTILITY PSB**
Specify the name of the PSB that this utility uses to access the IMS image copy data set. The value that you specify can be any PSB that has read access to all segments in the database.

**EXTRACT UTILITY PSBLIB**
Specify the name of the library that contains the PSB that is used by this utility.

**IMS RESLIB**
Specify the name of the IMS RESLIB in which this utility resides.
DBDLIB
Specify the data set name of the library that contains the DBD that is used by this utility.

DFSVSAMP DATASET NAME
Specify the name of the IMS VSAM buffer control data set that is used by this utility.

DFSVSAMP MEMBER NAME
Specify the name of the IMS VSAM buffer control data set member.

USER LOAD LIB
Specify the data set name of the load library that contains any preexisting user exit, randomizing exit, or compression exit routines that is needed by this utility.

EXTRACTED OUTPUT FILE PARAMETERS
Specify the primary and secondary space allocations, UNIT, VOLSER, and BLKSIZE parameters for the data set that is created by this utility. The VOLSER and UNIT fields can be left blank for DFSMS management. The BLKSIZE field can be the same as the one that is used for the IMS image copy data set. BLKSIZE must be large enough to contain the largest segment, plus an additional 50 bytes.

Image Copy Extract JCL requirements
The JCL for the Image Copy Extract utility can be created from your ISPF panel input. You can also create your own JCL.

The following example shows the JCL for the Image Copy Extract utility. You can use this sample to help you develop your own JCL for the Image Copy Extract utility.

```jcl
!HCOPROC PROC HCOPVERS=, HCOPMA71(IMS 7.1) HCOPMA81(IMS 8.1)
! PSB=, PSB NAME
! HCohlq=, IMSHLQ=
! IMShlq=, DATABASE NAME
! Databas=, DATABASE NAME
! Prim=, EXTRACTED OUTPUT PRIMARY SPACE ALLOC
! Sec=, EXTRACTED OUTPUT SECONDARY SPACE ALLOC
! Blk= BLOCK SIZE OF THE IMAGE COPY INPUT DATA SET
!*

!*******************************************************************
!* IMAGE COPY EXTRACT *
!*******************************************************************
!ICPYEXTR EXEC PGM=DFSRRC00,
! PARM='DLI,&HCOPVERS,&PSB,,,,,N,N',
! REGION=8192K
! STEPLIB DD DSN=&HCOHLQ..SHCOLMOD,DISP=SHR
! DD DSN=&IMSHLQ..RESLIB,DISP=SHR
!="/------------------------------------------------------------
!="/ FOR IMS DATA SETS
!="/*****************************************************************/
!="/ IMS DD DSN=&IMSHLQ..PSLIB,DISP=SHR
!="/ DD DSN=&IMSHLQ..DBDLIB,DISP=SHR
!="/ IMS2 DD DSN=&IMSHLQ..RESLIB,DISP=SHR
!="/ DD DSN=&HCohlq..SHCOLMOD,DISP=SHR
!="/ DFSRESLB DD DSN=&IMSHLQ..RESLIB,DISP=SHR
!="/ DFVSAMP DD DSN=&IMSHLQ..DFSVSAMP,DISP=SHR
!="/ IEFRDER DD DUMMY
!="/ SYSPRINT DD SYSDUT=*
!="/*******************************************************************/
!="/ REPORTS
```

Chapter 4. Extracting data 27
The Image Copy Extract JCL references the following data sets:

**STEPLIB**
The input PDS that contains the HCOPMAIN load module. The load library for your existing compression routines must be concatenated into this STEPLIB if compression routines are specified in the IMS image copy DBD.

**IMS**
The input PDS that contain your PSB and DBD load libraries.

**IMS2**
The input PDS that contain your IMS RESLIB and the IMS HD Compression Extended load library.

**DFSRESLB**
The input PDS that contain IMS load modules.

**DFSVSAM**
The input data set that contain buffer information for the DL/I buffer handler.

**IEFRDER**
The primary system log data set (code as DD DUMMY).

**SYSPRINT**
The output data set that contains IMS messages.
PRIMAPRT
A report data set that is used to print the control statements that are used and any statement errors. The report ID is HCO008.

STATIPRT
A report data set that is used to print DBD name, segment names, and number of occurrences of each segment read. The number of output segments written and any unidentified data read is also printed. Unidentified data is data that did not represent a valid segment or free space element, possibly due to slack bytes or errors in the data. The report ID is HCO009.

HCOQSAM
The output data set that contains each segment that was selected from the IMS image copy data set. The length, segment code, and segment name precede the data portion of the segment. If the image copy DBD indicates that a compression exit routine exists, the exit is invoked to uncompress the segment before it is extracted. The library that contains the exit must be concatenated into the STEPLIB. The exit is invoked in AMODE 31. The recommended block size for this data set is the block size of the input IMS image copy data set.

SEGIN
The input data set that contains the names of segments to be written to the HCOQSAM output data set. If DD DUMMY is coded, all segments are written to HCOQSAM. Each segment name must be on a separate line. Each segment name to be selected from the IMS image copy data set must be coded in column 1 of an 80-byte record. A maximum of 255 names can be specified.

//SEGIN DD *

ORDER
ORDART

PROCCTL
A control data set that specifies the database that contains segments to be written to the HCOQSAM output data set. The following code sample shows the syntax of the PROCCTL statement.

//PROCCTL DD *
PROC TYPE=SCAN
DATABASE DB=dbdname, DD=ddname, OVERFLOW=ddname, DATASET=IMAGECOPY
END

PROC TYPE=SCAN must be specified as the first control statement in the data set.

A DATABASE control statement must be added with the following parameters:
• DB=dbdname
  Name of the DBD. The specified database must be HDAM, HIDAM, or HISAM. HIDAM index or secondary index databases are not supported.
• DD=ddname
  Name of the DD that is described by the DBD for a data set group in the database. The specified value can be the same DD name that is used in the JCL DD statement for the IMS image copy data set that is being extracted. Only one data set from this data set group can be specified for the database in each execution of the Image Copy Extract utility.
• OVERFLOW=ddname
Name of the DD that is described by the DBD for the overflow part of a HISAM data set group. The specified value can be the same DD name that is used in the JCL DD statement for the IMS image copy overflow data set that is being extracted.

For HISAM databases, both the DD and OVERFLOW parameters are required. If you do not want to process HISAM overflow data sets, code DD DUMMY for the overflow DSN.

Only one DATABASE statement can be specified in PROCCTL. PROCCTL control statements can begin in column 2 or any column to the right of column 2.

**DDNAME**

The input image copy data set that was created during a previously run IMS image copy.

---

**HD Unload Extract utility**

The HD Unload Extract utility prepares segment data for a customized segment dictionary.

The HD Unload Extract utility produces a separate QSAM data set for each segment you extract. You can extract segments from either of the following sources:

- An IMS HD unload data set
- The output data set from the Image Copy Extract, which is useful if you want to create a customized segment dictionary but do not have an IMS HD unload data set

You enter the names of the segments to be extracted as control statements in a SYSIN DD data set. You can specify up to 255 different segment names for each database unload.

**Extracting specific segments by using HD Unload Extract**

The HD Unload Extract utility extracts specified segments from an IMS HD unload data set (or, in some cases, from the Image Copy Extract utility output). You run the HD Unload Extract utility by using the HCOZLDH0 program. HCOZLDH0 runs as a standard z/OS batch job.

**Before you begin**

DEDB unload data sets are not supported by this utility.

You can use either of the following methods to run the HD Unload Extract utility:

- From the ISPF panel-driven functions Detail Analyze and Examine
- Directly from JCL as a standard z/OS batch job

**About this task**

The following procedure describes how to run the utility from the ISPF panels.

**Procedure**

1. Allocate a PDS for the SYSIN data set to hold the extract control statements for each segment.
2. Determine the names of the segments that you want to extract.

3. From the ISPF panel of the Examine function, enter 2 in the INPUT SOURCE field to select the HD Unload Extract utility. If you are using the Detail Analyze function, the HD Unload Extract utility is selected automatically.

4. Optional: Evaluate potential compression savings by using the following functions:
   
   **Detail Analyze function**
   
   Build a separate data set for each segment that you want to evaluate.
   
   The Detail Analyze function shows you the compression savings each segment received from its corresponding customized segment dictionary.

   **Examine function**
   
   Build a separate data set for a segment to evaluate the compression savings that another existing dictionary can offer. The existing dictionary is compared to the customized segment dictionary for the segment.

5. Optional: If you want compression statistics for a complete database, process all the volumes of an IMS HD unload. If you do not need statistics for a complete database, use only one volume (or subset of one volume) to build a dictionary or to analyze the degree of compression that each segment receives.

6. Generate the JCL by exiting the ISPF panel.
   
   IMS HD Compression Extended can create all JCL that is required to run dictionary functions and the HD Unload Extract utility. You can use the ISPF dialogs for the Detail Analyze function and the Examine function to create JCL to process the SGOUTnnn DD extract data sets.

7. Edit the JCL. Create or change the SYSIN DD entries for the segments to be extracted. If you are using the ISPF panels, the SYSIN DD is opened in edit mode. Otherwise, you must add the necessary SYSIN DD entries to the JCL.

8. Interpret the output report to verify that the process completed successfully.

**Results**

When you run the HD Unload Extract utility from ISPF panels, each segment that is selected for extraction from the HD unload data set has its own output SGOUTnnn DD statement. The nnn is the sequence number of the segment name parameter in the SYSIN DD control statements.

The ISPF-built JCL sets up temporary data sets for these segments. You might want to create permanent data sets to save reprocessing time when you build the dictionary.

The report from the utility is sent to the HCO005 data set. The report ID is also HCO005.

**HD Unload Extract panel**

The HD Unload Extract panel opens when you use the Detail Analyze function. It also opens after you specify INPUT SOURCE=2 on the Examine panel.

The following figure shows a sample of the HD Unload Extract panel.
The HD Unload Extract panel contains the following fields:

**DATASET NAME OF PDS FILE FOR SYSIN CONTROL CARDS**
Specify the name of the PDS that was allocated for a SYSIN data set.

**MEMBER NAME**
Specify the name of the member in the SYSIN data set that contains the control statements for the segments to be extracted. The member can be new or existing. After the member name is entered, the member is opened in ISPF edit so that you can create or change the control statements.

For an explanation of the control statements and format for the SYSIN data set, see "HD Unload Extract JCL requirements."

**HD Unload Extract JCL requirements**
The JCL for the HD Unload Extract utility is created for you from ISPF panel input. You can also create your own JCL.

The following example shows the JCL for the HD Unload Extract utility. You can use this sample to help you develop your own JCL for the HD Unload Extract utility.

---

```plaintext
//HD UNLOAD EXTRACT

//*HD UNLOAD EXTRACT
EXEC PGM=HCOZLDH0,PARM='dbdname',REGION=20M
STEPLIB DD DSN=hcohlq.SHCOLMOD,DISP=SHR,PARM='ADBDNAME'
HCO005 DD SYSOUT=*,DCB=RECFM=FBA
HDCDHIN DD DSN=hlq.dbname.UNLOAD,DISP=OLD

//* THIS EXAMPLE SHOWS segname1 AND segname2 BEING EXTRACTED FROM THE HD UNLOAD FILE & WRITTEN OUT TO DDS SGOUT001 AND SGOUT002 RESPECTIVELY.
SGOUT001 DD DSN=&&SEG001,UNIT=SYSDA,DISP=(NEW,PASS),DCB=hlq.dbname.UNLOAD.DSN,SPACE=(CYL,(xxx),RLSE)
SGOUT002 DD DSN=&&SEG002,UNIT=SYSDA,DISP=(NEW,PASS),DCB=hlq.dbname.UNLOAD.DSN,SPACE=(CYL,(yyy),RLSE)
```
---

The following additional DD statements are also required. ISPF file tailoring (or manual edits) might be used to build appropriate SYSIN control stmts, SGOUTnnn DD cards and space allocation JCL params.
The temporary data sets are passed and concatenated as input to the HCOZLDE0 reporting function.

The HD Unload Extract utility JCL references the following data sets:

**EXEC**  This statement must adhere to the following format:

```
// EXEC PGM=HCOZLDH0,PARM='dbname',REGION=20M
```

For the PARM parameter, specify the name of the DBD for the database that is being extracted. This name is used for reporting purposes only. No IMS interface is invoked.

**STEPLIB**  The input PDS that contains the HCOZLDH0 load module.

**HCO005**  The output data set that contains the HD Unload Extract utility report.

**HDCDHIN**  The HD unload (or the equivalent) data sets that are input to the HD Unload Extract utility. Concatenate all data sets if you want segment statistics to be totaled for the database.

**SGOUTnnn**  The temporary output data set for the extracted data of a single segment. Each segment that is specified in the SYSIN DD statement has its own output SGOUTnnn DD statement. Each output data set and segment name are matched using nnn, which is the sequence number where the segment name is specified in the SYSIN DD statement.

The number of cylinders to be allocated (xxx and yyy in the example JCL) must also be specified in each SGOUTnnn DD statement.

You must change the DISP= parameter if you do not want the data set to be temporary.

**SYSIN**  The control data set that is used to specify control statements for each segment to be extracted. You must add a unique parameter that names each segment to be extracted.

```
//SYSIN DD *
SEGNAME1
SEGNAME2
```

Use columns 1 - 8 to specify the segment name from the DBD. This parameter is required.
Chapter 5. Analyzing segment data

The Detail Analyze function is a reporting tool that lets you evaluate hardware data compression without creating a compression exit routine or reloading the database.

The Detail Analyze function can analyze multiple segments in a single pass. The Detail Analyze function skips all HD unload segment prefix data when it analyzes a segment. It includes a KEYEND SYSIN option, which lets you skip the key field or other fields at the beginning of a segment. You can also skip the segment length (LL) field for variable-length segments. These options let you do a selective compression.

Acceptable input to the Detail Analyze function is an IMS image copy or HD unload (or the equivalent) data set. Regardless of the source of input, you must run the HD Unload Extract utility before you run the Detail Analyze function. The HD Unload Extract utility creates separate data sets for the segments that you have selected.

The Detail Analyze function builds a temporary dictionary from your sample data, rereads your data to compress it, and then produces the following two reports:

- Segment Compression Analysis Report
- Segment Distribution by Compressed Length Report

These reports provide information about generic and customized segment dictionary space savings for each segment that you select for analysis. These reports can help you prepare the reload DBD and specify optimal VSAM CI splits.

You run the Detail Analyze function by using the HCOZLDE0 program. The entire analysis runs as a single-step, standard z/OS batch job.

Topics:
- “Analyzing segment compression by using Detail Analyze”
- “Detail Analyze panel” on page 37
- “Detail Analyze JCL requirements” on page 38
- “Detail Segment Compression Analysis report (HCO006)” on page 40
- “Segment Distribution by Compressed Length report (HCO007)” on page 41

Analyzing segment compression by using Detail Analyze

You can use the Detail Analyze function to analyze the segment compression that you can achieve by using the generic and customized dictionaries.

Before you begin

To use the Detail Analyze function, you must first run the HD Unload Extract utility create a separate data set for each segment to be analyzed.

Restrictions:
- DEDB unload data sets are not supported by this function.
• A maximum of 255 segments can be analyzed per database.
• A segment smaller than 4 bytes is not analyzed for compression. Any such data is reported without change in both the before and after statistics.

About this task

You can use either of the following methods to run the Detail Analyze function:
• From ISPF as a panel-driven function
• Directly from JCL as a standard z/OS batch job

When you analyze many segments, the ISPF panel-driven function can save you time by building the required HD unload extract JCL, creating control statements that can be shared between the Detail Analyze function and the HD Unload Extract utility, and building the JCL for Detail Analyze.

The following procedure describes how to run the Detail Analyze function from the ISPF interface.

Procedure

1. Select the Detail Analyze function on the IMS HD Compression Extended main menu panel.
2. Make sure that the HD Unload Extract utility (HCOZLDH0) has been run.
   You can enter N for IMAGE COPY EXTRACT, or you can run the HD Unload Extract JCL.
3. Provide an acceptable source of input data.
   Acceptable input to the Detail Analyze function is either an IMS image copy or an HD unload (or the equivalent) data set.
   If you do not have an input source already available, you can access both the Image Copy Extract and HD Unload Extract utilities.
4. Optional: Optional: If you are using an image copy data set:
   a. Enter Y (yes) in the IMAGE COPY EXTRACT field on the Detail Analyze panel. Alternately, you can make the JCL for the Image Copy Extract utility the first step in the batch job that you create, so that the JCL precedes the HD Unload Extract utility step.
   b. Specify the names of the segments that you want to extract.
   You can specify these values on the Image Copy Extract utility panel. The Image Copy Extract utility panel opens after you finish entering all the required fields on the Detail Analyze panel.
5. Optional: Optional: If you are using an HD unload data set, complete the following steps:
   a. Allocate a PDS for the SYSIN data set if one does not exist. This value is validated by the panel, so it must exist before you can generate the JCL.
   b. Create or change the SYSIN DD entries for the segments to be extracted.
   You can specify these values on the HD Unload Extract utility panel. The HD Unload Extract utility panel opens after you finish entering all the required fields on the Detail Analyze panel.
6. Specify a data set name or accept the default data set name.
   If you want the Detail Analyze function to generate compression statistics for an entire database, specify a control statement in the SYSIN data set for every segment in the database.
Any segments that are not included in the SYSIN data set are not analyzed. When you run the Detail Analyze function from an ISPF panel, the function shares control statements with the HD Unload Extract utility.

7. Generate the JCL by exiting the ISPF panel.
   When you run the Detail Analyze function from an ISPF panel, the data sets that are produced by the HD Unload Extract utility are concatenated into the HDCDEIN DD statement of the generated JCL. The SYSIN PDS member that you select is opened in ISPF edit for creation, review, or update.

8. Optional: Edit the JCL if required.

9. Submit the generated JCL.

10. Interpret the output report to verify that the process completed successfully.

Results

The reports that are created by the Detail Analyze function are sent to the HCO006 and HCO007 data sets. The report IDs are the same name as the data sets.

The Detail Analyze function produces the following two reports:

**Segment Compression Analysis Report**

Use this report to evaluate the segments that you selected to see how effectively each was compressed and how much space was saved as a result. Space savings are shown for each segment based on a generic and customized segment dictionary. This report indicates which segments save the most space when they are compressed.

You can substitute an enterprise or customized database dictionary in place of the generic dictionary. This change can be made when the JCL that was created by using the ISPF Detail Analyze panel is edited. The amount of compression that is produced by one of these other dictionaries typically is better than the compression that is produced by the generic dictionary, but not as good as the compression that is produced by customized segment dictionaries.

**Segment Distribution by Compressed Length Report**

For each segment that you select, a distribution of lengths for the segment data after compression is provided. Use this distribution to help you specify the minimum segment size in the new DBD that will be used to reload the database. Consider a size from the upper range of lengths if the segment is updated frequently and is subject to segment splits.

---

**Detail Analyze panel**

The Detail Analyze panel opens when you select the Detail Analyze function on the IMS HD Compression Extended main menu panel.

The following figure shows a sample of the Detail Analyze panel.
The Detail Analyze panel contains the following fields:

**IMAGE COPY EXTRACT**
Type Y if your input data set (specified in INPUT DSNAME) is an IMS image copy data set. Otherwise, specify N.

**INPUT DSNAME**
Specify the fully qualified name of the input data set. This data set must be an IMS image copy or HD unload (or the equivalent) data set. If you specify Y in the IMAGE COPY EXTRACT field, this data set must be an IMS image copy data set.

**USER DSN HLQ**
Specify the high-level qualifier for the output data sets.

**DATABASE IDENTIFIER**
Specify a name to identify the database that contains the segments that are being analyzed. For example, you can use the name of the DBD.

Based on the values that you provide, IMS HD Compression Extended returns a panel for extracting data, creating the SYSIN control data set, and creating JCL.

### Detail Analyze JCL requirements

The JCL for the Detail Analyze function can be created from your ISPF panel input. You can also create JCL for the Detail Analyze function manually.

If SYSIN DD DUMMY is used, all input segments are analyzed and are assumed to be fixed length with a KEYEND value of 00000.

If the SYSIN statements are built by using the HDCE ISPF panels, the value of the CYL parameter remains in columns 18-21. The value is ignored by the Detail Analyze process.

The following example shows the JCL for the Detail Analyze function. You can use this sample to help you develop your own JCL for the Detail Analyze function.

```plaintext
//**********************************************************************
// DETAIL ANALYZE FUNCTION
//**********************************************************************
//** EXEC PGM=HCOZLDE0,PARM='dbname',REGION=8M
//** STEPLIB DD DSN=hcoh1q.SHCOLMOD,DISP=SHR
```
The Detail Analyze JCL references the following data sets:

**EXEC**  This statement must adhere to the following format:

```
// EXEC PGM=HCOZLDE0,PARM='dbname',REGION=8M
```

For the PARM parameter, specify the name of the DBD for the segments that is being analyzed. This name is used for reporting purposes only. No IMS interface is invoked.

**STEPLIB**  The input PDS that contains load module HCOZLDE0.

**HCO006**  The output data set that contains the compression report for a generic or customized segment dictionary. If you substitute a different dictionary for the generic dictionary, the generic dictionary that is specified in the output is actually the substituted dictionary. Each input segment is on a unique line.

**HCO007**  The output data set that contains the distribution report of compressed segment lengths. Each input segment has a unique page.

**HDCDICT**  The fully qualified name of the generic dictionary (hlq.HCOGDIC1.HDCDDICT) or the dictionary you substitute. The dictionary must have the z/OS data set format of 16 records of 4 KB each. Do not use the dictionary object data set.

**HDCDWK01**  The temporary output data set that is used for accumulating statistics.

**HDCDEIN**  The input DD of concatenated segment data sets from the HD Unload Extract utility.

**SYSIN**  The control data set that is used to specify control statements for each segment to be extracted. For each segment, you must specify unique parameters that identify the following attributes:

- The name of the segment to be extracted
- The end position of the segment key
- The segment type (fixed or variable)

```
columns 1 - 8
```

The segment name from the DBD. This parameter is required.
The KEYEND parameter, which specifies the byte number in the segment where the key data ends. Five zeros (00000) means that key compression is wanted.

The SEGTYPE parameter, which specifies the segment type. You can specify either F (fixed) or V (variable). The 2-byte LL field for variable-length segments is skipped if V is specified.

The following code shows an example of how you specify SYSIN data set input.

```
//SYSIN DD *
SEGNAME1 00000 F
SEGNAME2 00000 V
```

**Detail Segment Compression Analysis report (HCO006)**

The Detail Segment Compression Analysis report (HCO006) report shows the compression analysis report that is produced by the Detail Analyze function. IMS prefix length and access method usage are excluded from these statistics.

The following figure show a typical Detail Segment Compression Analysis report:

```
<table>
<thead>
<tr>
<th>SEGMENT NAME</th>
<th>KEY END</th>
<th>SEG TYPE</th>
<th>DICT TYPE</th>
<th>BEFORE COMPRESS</th>
<th>AFTER COMPRESS</th>
<th>SAVINGS</th>
<th>INPUT</th>
<th>BUILD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>KBYTES USED AVG LGTH</td>
<td>KBYTES USED AVG LGTH</td>
<td>KBYTES PCT</td>
<td>RECORDS</td>
<td>RECORDS</td>
</tr>
<tr>
<td>SEGMENTA</td>
<td>2</td>
<td>V</td>
<td>Custom</td>
<td>3,291 247</td>
<td>162 12</td>
<td>3,129 95</td>
<td>13,625</td>
<td>11,602</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Generic</td>
<td>524 39</td>
<td>2,767 84</td>
<td>13,625</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEGMENTC</td>
<td>0</td>
<td>V</td>
<td>Custom</td>
<td>10,282 761</td>
<td>149 11</td>
<td>10,133 98</td>
<td>13,832</td>
<td>13,104</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Generic</td>
<td>571 42</td>
<td>9,711 94</td>
<td>13,832</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS (KB)</td>
<td></td>
<td></td>
<td>Custom</td>
<td>13,573 311</td>
<td>1,095 89</td>
<td>13,262 97</td>
<td>13,478</td>
<td>91</td>
</tr>
</tbody>
</table>
```

**Figure 8. Detail Segment Compression Analysis report**

The Detail Segment Compression Analysis report contains the following information:

**SEGMENT NAME**

The name of the segment that was selected for analysis.

**KEYEND**

Byte number in the segment where the key data ends. Compression starts following the specified value. If you specify 00000, key data is included in the analysis.

**SEGTYPE**

The type of segment, which can be either variable (V) or fixed (F).

**DICTTYPE**

The type of dictionary that was used to compress the data for analysis.
DATA LENGTH AFTER KEYEND
Provides the following information:

KBYTES USED
Kilobytes of data that were read by the Detail Analyze function before and after the segment was compressed.

AVG LGTH
The average length of the segment data that was read by the Detail Analyze function before and after compression.

SAVINGS KBYTES
The number of kilobytes saved, which is determined as follows:
BEF COMPRESS KB USED - AFT COMPRESS KB USED

SAVINGS PCT
The percent of storage saved, which is determined as follows:
100 - ((AFT COMP KB USED / BEF COMP KB USED)*100)

INPUT RECORDS
The number of records that the Detail Analyze function analyzed for each segment.

BUILD RECORDS
The number of input records that the Detail Analyze function used to build the dictionary.

---

Segment Distribution by Compressed Length report (HCO007)

The Segment Distribution by Compressed Length report (HCO007) shows the compression analysis reports that are produced for each segment by the Detail Analyze function. A distribution report is created for each segment that is specified to the Detail Analyze function.

The report provides a count of the number of compressed segment lengths that fall into each segment range. This information can help you determine the maximum and minimum segment size in the new DBD. Consider a size from the upper range of lengths if the segment is updated frequently and is, therefore, subject to segment splits. For instructions on how to modify or create a DBD, see “Modifying a DBD” on page 53.

The following figure shows the Segment Distribution by Compressed Length report that is produced by the Detail Analyze function. IMS prefix length and access-method usage are excluded from these statistics.
The Segment Size Distribution by Compressed Lengths report contains the following information:

**LENGTH OF COMPRESSED SEGMENT DATA AFTER KEYEND**
The range of lengths across which compressed segment data is distributed.

**CUSTOM COMPRESSION**
The number of segments with lengths that were compressed to a size within that range. A percentage of the total number of segments is given for each range. The statistics are for a customized segment dictionary strategy.

**GENERIC COMPRESSION**
The number of segments with lengths that were compressed to a size within that range. A percentage of the total number of segments is given for each range. The statistics are for a generic dictionary strategy. If you substitute a different dictionary for the generic dictionary, the generic dictionary that is specified in the output is actually the substituted dictionary.

**INPUT RECORDS**
The number of records that the Detail Analyze function analyzed for the segment.

---

<table>
<thead>
<tr>
<th>LENGTH OF COMPRESSED SEGMENT DATA AFTER KEYEND (BYTE RANGES)</th>
<th>NUMBER AND PERCENT OF INPUT RECORDS WITHIN EACH LENGTH RANGE</th>
<th>INPUT RECORDS: 13,832</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CUSTOM COMPRESSION</td>
<td>GENERIC COMPRESSION</td>
</tr>
<tr>
<td>1 - 15</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>16 - 23</td>
<td>11,077 80%</td>
<td>0 0%</td>
</tr>
<tr>
<td>24 - 31</td>
<td>602 4%</td>
<td>504 3%</td>
</tr>
<tr>
<td>32 - 39</td>
<td>0 0%</td>
<td>1,106 7%</td>
</tr>
<tr>
<td>40 - 47</td>
<td>400 2%</td>
<td>1,672 12%</td>
</tr>
<tr>
<td>48 - 55</td>
<td>249 1%</td>
<td>2,556 18%</td>
</tr>
<tr>
<td>56 - 63</td>
<td>84 0%</td>
<td>188 1%</td>
</tr>
<tr>
<td>64 - 71</td>
<td>810  5%</td>
<td>282 2%</td>
</tr>
<tr>
<td>72 - 79</td>
<td>294 2%</td>
<td>73 0%</td>
</tr>
<tr>
<td>80 - 87</td>
<td>38 0%</td>
<td>4 0%</td>
</tr>
<tr>
<td>88 - 95</td>
<td>76 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>96 - 103</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>104 - 127</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>128 - 135</td>
<td>0 0%</td>
<td>298 2%</td>
</tr>
<tr>
<td>136 - 143</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>144 - 151</td>
<td>0 0%</td>
<td>199 1%</td>
</tr>
<tr>
<td>152 - 207</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>208 - 215</td>
<td>0 0%</td>
<td>399 2%</td>
</tr>
<tr>
<td>216 - 271</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>272 - 279</td>
<td>0 0%</td>
<td>199 1%</td>
</tr>
<tr>
<td>280 - 327</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
<tr>
<td>328 - 335</td>
<td>0 0%</td>
<td>199 1%</td>
</tr>
<tr>
<td>336 - 32,767</td>
<td>0 0%</td>
<td>0 0%</td>
</tr>
</tbody>
</table>

---

Figure 9. Segment Size Distribution by Compressed Lengths report
Chapter 6. Analyzing database data

The Analyze function is a reporting tool that lets you evaluate hardware data compression without creating a compression exit routine or reloading the database.

The Analyze function analyzes one entire data set or one selected segment at a time. The Analyze function builds a temporary dictionary from your sample data, rereads your data to compress it, and then produces a Compression Projection report.

The Compression Projection report details projected segment data compression. You can use this report to evaluate the database or segment space that would be saved if you created a dictionary to compress the data.

The Analyze function can evaluate enterprise or customized database dictionary solutions.

You run the Analyze function by using the HCOZLDU0 program. The entire analysis runs as a single-step, standard z/OS batch job.

Topics:

- “Analyzing segment or database compression by using Analyze”
- “Analyze panel” on page 45
- “Analyze JCL requirements” on page 46
- “Compression Projection report (HCO001)” on page 47

Analyzing segment or database compression by using Analyze

Use the Analyze function to produce a report to evaluate data space savings that would result if you created a dictionary to compress the input data. The ISPF panel functions help you create and submit JCL.

Before you begin

The Analyze function has a START BYTE option, which lets you adjust where compression begins in the input records. You can skip key data, unload-record prefix data, or specific fields in a segment. These choices let you do a selective compression.

START BYTE is different from the KEYEND SYSIN option in the Detail Analyze function. The Detail Analyze function input is always in HD unload format, whereas the Analyze function input can come from any variable-blocked QSAM input source. As a result, by using the Analyze function you can specify where compression starts in the input records.

The Analyze function supports DEDB unload data sets.

If your data is not an image copy data set, you can run the Analyze function without having to run the Extract function.
About this task

You can use either of the following methods to run the Analyze function:
- From ISPF as a panel-driven function
- Directly from JCL as a standard z/OS batch job

The following procedure describes how to run the utility from the ISPF panels.

Procedure

1. Specify the dictionary name.
2. Provide an acceptable source of input data. Acceptable input includes any of the following data set types:
   - DEDB unload data sets
   - Image copy extract data sets
   - HD unload extract data sets (or equivalent)
   If you do not have an input source already available, the function provides access to both the Image Copy Extract and HD Unload Extract utilities.
3. Run the Analyze function from ISPF as a panel-driven function

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you are using an image copy data set</td>
<td>From the Analyze function ISPF panel, enter Y (yes) in the IMAGE COPY EXTRACT field.</td>
</tr>
<tr>
<td>If you are using an HD unload data set</td>
<td>From the Analyze function ISPF panel, enter N (no) in the IMAGE COPY EXTRACT field.</td>
</tr>
</tbody>
</table>

4. Complete the other required fields on the ISPF panel.
5. Specify a data set name in the INPUT HDCDIT DSNAME field on the ISPF panel to generate statistics.
   This data set can be the same one that was specified in the INPUT DSNAME field.
   If a name is not specified, the INPUT DSNAME data set name is used.
6. Optional: If necessary, specify job card data on the Job Card Data panel.
7. Generate the JCL by exiting the ISPF panel.
8. Optional: Edit the JCL if required.
9. Submit the generated JCL.
10. Interpret the output report to verify that the process completed successfully.

Results

The report that is created by the Analyze function is sent to the SYSPRINT data set. The report ID is HCO001.

Output from the Analyze function is a report that shows projected segment data compression. The Analyze function produces a Compression Projection report. Use this report to evaluate (based on the input to the Analyze function) database or segment data space savings if a dictionary was created to compress the input data.
The Analyze panel opens when you select the Analyze function on the IMS HD Compression Extended main menu panel.

The fields on this panel are used to open the analyze source for editing and to generate the analyze JCL. The following figure shows a sample of the Analyze panel.

--- IMS HD Compression Extended (Function = ANALYZE) ---

<table>
<thead>
<tr>
<th>Command</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DICTIONARY NAME</td>
<td>=&gt;</td>
<td></td>
</tr>
<tr>
<td>IMAGE COPY EXTRACT</td>
<td>=&gt;</td>
<td>Y OR N</td>
</tr>
<tr>
<td>INPUT DSNAMES</td>
<td>=&gt;</td>
<td></td>
</tr>
<tr>
<td>INPUT HCDIT DSNAMES</td>
<td>=&gt;</td>
<td></td>
</tr>
<tr>
<td>USER DSN HLQ</td>
<td>=&gt;</td>
<td></td>
</tr>
<tr>
<td>START BYTE</td>
<td>=&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Press END to exit.

Figure 10. Analyze panel

The Analyze panel contains the following fields:

**DICTIONARY NAME**
Specify the dictionary name. You can specify up to eight characters for the name. This name is used for reporting purposes only; no permanent dictionary is created.

**IMAGE COPY EXTRACT**
Type Y if your input data set (specified in INPUT DSNAMES) is an IMS image copy data set. Otherwise, specify N.

**INPUT DSNAMES**
Specify the fully qualified name of the input data set. This data set must be an IMS image copy or HD unload (or the equivalent) data set. If you specify Y in the IMAGE COPY EXTRACT field, this data set must be an IMS image copy data set.

**INPUT HCDIT DSNAMES**
Specify the name of the data set that was used to generate compression statistics. If this data set is different from the one that was specified in INPUT DSNAMES, statistics are generated for this (INPUT HCDIT DSNAMES) data set's space savings for the dictionary that was built from the INPUT DSNAMES data set. For example, you might want to create a dictionary from one volume of an HD unload data set but use a different volume of the same data set to generate the statistics.

Specification of this field is optional. It defaults to the data set name that is specified in INPUT DSNAMES.

**USER DSN HLQ**
Specify the high-level qualifier for the output data set.
**START BYTE**

Specify the relative byte number in the input source record at which compression analysis begins. Its point of reference is the beginning of the record, not the end of an unload utility record prefix (as is the case with the KEYEND parameter of the Detail Analyze function). The START BYTE option lets you adjust where compression analysis begins in the input records. The following figures show examples of the START BYTE option used for the HCDIN and HCDIT input data sets.

---

**Figure 11. Example of the START BYTE option for an IMS Unload**

You can specify the offset where the compression analysis begins for an IMS Unload or an Image Copy Extract record. As shown in Figure 11, for example, if START BYTE = 35, the data that starts at position 36 is used for dictionary building and statistics generation. The IMS prefix data is ignored. If START BYTE = 48, the data that starts at position 49 is used for dictionary building and statistics generations. Both the IMS prefix data and the segment key are ignored.

---

**Figure 12. Example of the START BYTE option for a user-built record**

You can specify the offset where the compression analysis for a user-built record begins. As shown in Figure 12, for example, if you specify START BYTE = 0, the data that starts at position 1 is used for dictionary building and statistics generation. Unwanted user prefix data is ignored. If START BYTE = 29, the data that starts at position 30 is used for dictionary building and statistics generations. Unwanted user prefix data is ignored.

---

**Analyze JCL requirements**

You can create the JCL for the Analyze function by using ISPF panel input. You can also create your own JCL.

The following example shows the JCL for the Analyze function. You can use this sample to help you develop your own JCL for the Analyze function.
The Analyze JCL references the following data sets:

**EXEC**
This statement must have the following format:

```c
// EXEC PGM=HCOZLDU0,PARM='dictnameAnnnn'
```

where:

- **dictname**
  Specifies the 1- to 8-character name of the dictionary from long.

- **A**
  The letter A, as the ninth character in the parameter list, indicates that you are running the Analyze function. This specification can be omitted. It defaults to A.

- **nnnnn**
  Specifies the value of the START BYTE option. This 1- to 5-character specification begins at character 10 in the parameter list. This specification can be omitted. It defaults to 0. Figure 11 on page 46 shows examples of use of this parameter.

**STEPLIB**
The input PDS that contains load module HCOZLDU0.

**HDCDIN**
The input data set that is used to build the temporary dictionary.

**HDCDIT**
The name of the data set that is used to generate compression statistics. If this data set is different from the one that is specified in HDCDIN, statistics are generated for this (HDCDIT) data set's space savings for the dictionary that was built from the HDCDIN data set. For example, you might want to create a dictionary from one volume of an HD unload data set but use a different volume of the same data set to generate the statistics.

Specifying a data set name in this field is optional. It defaults to the data set name that is specified in INPUT DSNAME.

**HDCDOUT**
Must be specified as DD DUMMY. A permanent dictionary is not created by the Analyze function.

---

**Compression Projection report (HCO001)**

The Compression Projection report (HCO001) shows the compression projection report that is produced by the Analyze function. You can use this report to evaluate the compression you could achieve using a custom dictionary, without having to create a compression exit routine or reloading the database.
Before you run the Compress function, lengths might not match the DBD-specified lengths. If a segment with an odd length value is defined in the DBD, a hexadecimal zero pad byte is added at the end of each segment by the IMS Unload utility. If such records are used as input to HDCE, the lengths differ from the DBD-specified lengths.

The following figure shows the compression projection report produced by the Analyze function for a custom dictionary. IMS prefix length and access method usage can be excluded from these statistics.

---

**Figure 13. Compression Projection report**

The Compression Projection report that is generated by the Analyze function contains the following information:

**START BYTE**
The value that is specified in the START BYTE parameter.

**DICT TYPE**
The type of dictionary, which is always custom.

**DATA LENGTH FROM START BYTE**
Shows the following columns:

- **BEFORE COMPRESS MIN**
  Smallest input segment size before compression.

- **BEFORE COMPRESS AVG**
  Average input segment size before compression.

- **BEFORE COMPRESS MAX**
  Largest input segment size before compression.

- **AFTER COMPRESS MIN**
  Smallest input segment size after compression.

- **AFTER COMPRESS AVG**
  Average input segment size after compression.

- **AFTER COMPRESS MAX**
  Largest input segment size after compression.

- **SAVINGS AVG**
  Average savings of bytes per segment.

Bytes that are skipped as a result of the START BYTE specification are excluded.
SAVINGS PCT
The percent of storage that is saved as determined by the following formula:

\[ \text{PCT SAVINGS} = \frac{\text{AVG SAVINGS} \times 100}{\text{AVG BEFORE}} \]

Bytes that are skipped as a result of the START BYTE specification are excluded.

INPUT RECORDS
The number of records that the Analyze function read from the HDCDIT data set to generate these statistics.

BUILD RECORDS
The number of records that the Analyze function read to build the temporary dictionary (from the HDCDIN data set).
Chapter 7. Building and relinking a dictionary

Use the Build function to help you build a dictionary.

After you have decided what segments or databases that you want to compress, you use the Build function to build a dictionary and an IMS compression exit routine. The dictionary that the Build function creates can be an enterprise, customized segment, or customized database dictionary.

The Build function performs the following tasks:
• Creates a dictionary and the compression exit routine that uses it when HDC is invoked
• Evaluates the space savings that are reported by the dictionary build process so that you can check that expected compression conforms to your earlier expectations (based on earlier analysis)
• Generates DBDGEN JCL after you edit the old DBD or create a DBD

The Build function has a START BYTE option, which lets you adjust where compression begins in the input records. You can also do selective compression by skipping key data, unload-record prefix data, or specific fields in a segment.

The Build function produces a Compression Projection report. You can compare the space savings that are projected in this report to the savings that are projected from an earlier analysis.

You run the Build function by using the HCOZLDU0 program. The Build function runs as a single-step, standard z/OS batch job.

Topics:
• “Building a custom dictionary by using Build” on page 53
• “Creating a compression exit routine” on page 53
• “Modifying a DBD” on page 53
• “Relinking a compression exit routine” on page 54
• “Build panel” on page 55
• “Compression Exit Routine panel” on page 57
• “Edit DBD panel” on page 58
• “COMPRTN keyword” on page 58
• “Build JCL requirements” on page 60
• “Compression exit routine and DBD JCL requirements” on page 61
• “Compression Projection report (HCO002)” on page 62

Building a custom dictionary by using Build

Use the Build function to build your custom dictionary. Creating a custom dictionary is optional, but typically custom dictionaries provide the most optimal data compression.
Before you begin

The Build function can use any of the acceptable input sources, including DEEDB unload data sets.

Restrictions:

- Only one dictionary can be created with each execution of the Build function.
- When you use a HIDAM or HISAM database HD Unload file to build a dictionary, the beginning records are not a representative sample of the database. You must use an IMS HD Compression Extended extract utility or your own application in these cases.

You can use either of the following methods to run the Build function:

- From ISPF as a panel-driven function
- Directly from JCL that is run as a standard z/OS batch job

The ISPF panel functions help you create and submit JCL and modify the DBD.

About this task

The following procedure describes how to run the utility from the ISPF panels.

Procedure

1. Provide a variable-blocked QSAM data set that contains uncompressed database segments. You can create the QSAM data set by using an appropriate program or one of the following utilities:
   - Image Copy Extract utility
   - HD Unload Extract utility
2. Run the Build function from ISPF as a panel-driven function. Either complete the required fields on the Build panels, or create the JCL yourself.
3. Specify a data set name in the INPUT HDCDIT DSNAME field on the ISPF panel to generate statistics.
   - This data set must be the same one that was specified in the INPUT DSNAME field.
   - If a name is not specified, the INPUT DSNAME data set name is used.
4. Optional: For selective compression, adjust where compression begins in the input records by using the START BYTE option.
   - You can skip key data, unload-record prefix data, or specific fields in a segment.
5. Optional: Edit the DBD if required.
   - If you specify Y for the EDIT DBD SOURCE field, the Job Card panel, the Compression Exit panel, and the DBD panel are displayed.
6. Generate the JCL by exiting the ISPF panel.
7. Optional: Edit the JCL if required.
8. Submit the generated JCL.
9. Interpret the output report to verify that the process completed successfully.
Results
The report that is created by the Build function is sent to the SYSPRINT data set. The report ID is HCO001.

The Build function produces a Compression Projection report. Compare the space savings that are projected in this report with the space savings that were projected from an earlier analysis.

Creating a compression exit routine
An essential part of using IMS HD Compression Extended is the creation of the exit routine that is used to compress your data.

About this task
You can use either of the following methods to create a compression exit routine:
- From ISPF as a panel-driven function
- By editing JCL

You create an IMS compression exit routine by linking together the following items:
- IMS HD Compression Extended dictionary
- HCOZLDX0 or HCOZLDXV (compression drivers)

Procedure
1. Run the Build function from ISPF as a panel-driven function, and specify Y for the EDIT DBD SOURCE field.
2. On the Compression Exit Routine panel, specify the following values:
   - Compression dictionary name
   - Compression driver name
   - Compression exit name
     The dictionary must be the first CSECT in the compression exit routine, and it must be on a page boundary.
3. Exit ISPF or continue on to the DBD panel.

Modifying a DBD
The database definition must point to the correct compression exit routine to use.

About this task
IMS runs compression or expansion by calling the compression exit routine that is specified in the COMPRTN keyword of the DBDGEN SEGM statement. You must create or modify the existing DBD to specify the compression exit routine.

Procedure
1. Add or change the COMPRTN keyword of the SEGM statement for each segment that contains compressed data. The COMPRTN keyword must specify either KEY or DATA compression, and must specify INIT. For more information, see "COMPRTN keyword" on page 58. The following example shows how to enter the COMPRTN keyword of the SEGM statement for fixed-length segments:
The following example shows how to enter the COMPRTN keyword of the SEGM statement for variable-length segments:

```
SEGM NAME=SEGMENTA, PARENT=0, BYTES=(256, 16), PTR=TB,
     COMPRTN=(DRVRSCNM, DATA, INIT)
```

2. To allow for control information, add 10 bytes to the maximum length definition for database segments.

   If the format or contents of the data in a segment precludes compression, the HDCE exit routine adds control information that indicates that the segment cannot be compressed.

   The control information might lengthen the segment beyond the maximum length definition. For example, an application on a test IMS system uses a production compression exit. If the test application tries to insert an X'00' string segment and no X'00' data was present when the production dictionary was built, then that input string cannot be compressed.

   Failure to increase the MAXBYTES field by 10 might result in an ABENDU0853.

   a. Modify the MAXBYTES field of the BYTES parameter on the SEGM definition in your database definition (DBD) for fixed- and variable-length segments in full function (HIDAM, HISAM, and HDAM) and Fast Path (DEDB) databases.

   b. Optional: If your database is a DEDB, make sure that the new MAXBYTES field is at least 120 bytes less than the VSAM control interval (CI) size. If it is not, you must redefine your VSAM CI size.

   c. If a segment cannot be compressed due to data inconsistency, IMS manages the increased length internally with no affect to the application or DBD. For more information, see the IMS Version 9 Customization Guide (SC18-7817) and the IMS Version 10 Exit Routine Reference (SC18-9708).

3. Make sure the maximum and minimum size of the segment are adjusted to allow for padding as needed. To determine the maximum and minimum segment size in the new DBD, see the Segment Distribution by Compressed Length report (HCO007) on page 41.

4. Run an ACBGEN for online databases. Running ACBGEN is required whenever you modify a DBD.

### Relinking a compression exit routine

You must relink all IMS compression exit routines if you upgrade IBM Tools Base for z/OS to a new release or apply maintenance to the IMS compression driver program (HCOZLDX0 or HCOZLDXV).

#### About this task

You can use sample job HCORLNK in the sample library to link the updated HCOZLDX0 program into your existing IMS compression exit routines.

#### Procedure

1. Put a copy of the old exit in a backup library (to be used if the new link edit fails).
2. Create a copy of the old exit on the test system.
3. Create a copy of the database on the test system.
4. Link the new exit into test user.RESLIB. Use the old exit (see DDNAME DD1 in the sample JCL below) and the new exit driver (HCOZLDX0 or HCOZLDXV, see DDNAME DD2 in the sample JCL below) as SYSLIN input.

5. Do one of the following steps:
   • Unload and reload the test database.
   • Create a DFSDDLT0 job that does GET NEXTs and updates through the test database to check expansion and compression.

6. After testing is successful:
   a. Create production database backups by using the image copy or unload utilities.
   b. Install the updated exits into the production IMS exit library using IMS exit update procedures.

Example

The following sample JCL shows how sample job HCORLNK can be used to link HCOZLDX0 or HCOZLDXV to the IMS compression exit routines.

```jcl
//*-----------------------------------------------------------------*
//* HCT V2R2 - LINK EDIT - HCO RELINK DICTIONARY LOAD MODULES *
//*-----------------------------------------------------------------*
//LINK2 EXEC PGM=IEWL,
//   PARM='SIZE=(880K,64K),RENT,REFR,NCAL,LET,XREF,LIST=ALL'
//SYSPRINT DD SYSOUT=*
//DD1 DD DSN=hlq.user.RESLIB,DISP=SHR
//SYSLMOD DD DSN=hlq.user.RESLIB,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSLIN DD *
INCLUDE DD2(HCOZLDX0) /* new HCOZLDX0 */
INCLUDE DD1(compexit) /* name of compression exit routine */
ORDER HCOZHDCD,HCOZLDX0 /* dictionary must be first */
PAGE HCOZHDCD
ENTRY HCOZLDX0
NAME compexit(R) /* name of compression exit routine */
```

Build panel

The Build panel opens when you select the Build function on the IMS HD Compression Extended main menu panel.

The fields on this panel are used to generate the Build JCL. The following figure shows a sample of the Build panel.
The Build panel contains the following fields:

**DICTIONARY NAME**
Specify the dictionary name. You can specify up to eight characters for the dictionary name. This name is used as the name for these output data sets:
- HDCDDICT
- HCDSTAT
- HDCDOUT

**IMAGE COPY EXTRACT**
Type Y if your input data set (specified in INPUT DSNAME) is an IMS image copy data set. Otherwise, specify N.

**INPUT DSNAME**
Specify the fully qualified name of the input data set. This data must already be processed by an extract utility or be in variable-blocked QSAM format for input source data.

**INPUT HDCDIT DSNAME**
Specify the name of the data set that is used to generate compression statistics. If this data set is different from the one that is specified in INPUT DSNAME, statistics are generated for this (INPUT HDCDIT DSNAME) data set's space savings for the dictionary that was built from the INPUT DSNAME data set. For example, you might want to create a dictionary from one volume of an HD unload data set but use a different volume of the same data set to generate the statistics.

Specification of this field is optional. It defaults to the data set name that is specified in INPUT DSNAME.

**USER DSN HLQ**
Specify the high-level qualifier for the output data set.

**USER DSN VOLSER**
Specify the volume ID, if needed, to which the HDCDDICT, HCDSTAT, and HDCDOUT data sets are output.

**UNIT RECORD DEVICE**
Specify the device that is assigned to the output data sets.
EDIT DBD SOURCE
Specify whether you want to edit the DBD source to change the compression-sensitive SEGM statements. If you select Y, JCL for DBDGEN is built. If you select N, no action is taken.

START BYTE
Specify the relative byte number in the input source record at which compression analysis begins. Its point of reference is the beginning of the record, not the end of an unload utility record prefix, as is the case with the KEYEND parameter of the Detail Analyze function. START BYTE lets you adjust where compression analysis begins in the input records. Figures 11 on page 46 and Figure 12 on page 46 show examples of this parameter.

Compression Exit Routine panel
The Compression Exit Routine ISPF panel opens to prompt you for information to build the dictionary link-edit JCL and the compression exit routine.

The Compression Exit Routine panel is displayed as part of running the Build function through the ISPF interface. The following figure shows a sample of the Compression Exit Routine panel.

```
------- IBM IMS HD Compression -------- Function = BUILD ------------------

COMMAND ===>

COMP NAME ==> HDCDICT1 COMPRESS DICTIONARY NAME.
DRIVER NAME ==> HCOZLDXV COMPRESS DRIVER NAME. (HCOZLDX0/HCOZLDXV)
EXIT NAME ==> COMPRTN1 COMPRESS EXIT NAME.
USER RESLIB WHERE COMPRESSION EXIT MUST BE PLACED
   ==> IMSTOOLS.HSMCE.PGMLIB

Press END to exit.
```

Figure 15. Compression Exit Routine panel

The exit routine panel contains the following fields:

COMP NAME
Specify the name of the dictionary that is to be linked to the dictionary driver. Typically, this name is the same as the DICTIONARY NAME in Figure 14 on page 56.

DRIVER NAME
Specify the name of the compression dictionary driver.

EXIT NAME
Specify the name of the IMS compression exit routine.

USER RESLIB WHERE COMPRESSION EXIT MUST BE PLACED
Specify the name of the library that will contain the IMS compression exit routine.
Edit DBD panel

The Edit DBD ISPF panel opens when you select EDIT DBD SOURCE = Y on the Build panel. The fields on this panel are used to open the DBD source for editing and to generate the DBDGEN JCL.

The Edit DBD ISPF panel is displayed as part of specifying the information needed for the Build function. The following figure shows a sample of the Edit DBD panel.

![Edit DBD panel](image)

Press END to exit.

Figure 16. Edit DBD panel

The edit DBD panel contains the following fields:

- **DBD MACRO LIBRARY DATASET NAME**
  Specify the data set that contains the library where DBD macro files are located.

- **DBD SOURCE LIBRARY**
  Specify the library where database definition DBD files are located.

- **DBD SOURCE LIBRARY MEMBER NAME**
  Specify the member name of the library where DBD files are located.

- **NEW DBD LOAD LIBRARY**
  Specify the name that you want to give to the DBD load library.

- **NEW DBD LOAD LIBRARY MEMBER NAME**
  Specify the name that you want to give to the DBD load library.

**COMPRTN keyword**

The COMPRTN keyword of the SEGM statement for each segment that contains data must be modified to allow for data compression.
Purpose

The COMPRTN keyword statement specifies the compression exit routine.

```
COMPRTN=(EXITNAME, DATA, INIT, MAX, PAD)
```

Parameters

The following parameters apply to both variable-length and fixed-length segments:

**EXITNAME**
- Specifies the name of the user-supplied compression exit routine. This name must be a 1- to 8-character alphanumeric value and must not be the same as any other name in IMS.RESLIB.

**DATA**
- Specifies that the indicated exit routine compresses or modifies data fields only. Sequence fields must not be modified, nor data fields that change the position of the sequence field in respect to the start of the segment. DATA is the default value if a compression routine is named but no operand is selected. When you use the DATA option, only data after the key field is compressed. If your key field is not first in your records, using this option might not result in the best possible compression.

**KEY**
- Specifies that the exit routine can compress or modify any fields within the named segment. This parameter is invalid for the root segment of a HISAM database. Use this option to achieve maximum data compression.

**INIT**
- Indicates that initialization and termination processing control are required by the segment exit routine. When this parameter is specified, the edit or compression routine gains control after a database opens and after a database closes.

The following parameters apply to fixed-length segments:

**MAX**
- Specifies the maximum number of bytes by which fixed-length segments can increase during compression exits. You can specify from 1 bytes to 32,767 bytes. This keyword is optional.

**PAD**
- Use the PAD keyword to indicate that the numeric value that is supplied in the previous parameter of MAX is used for padding and not for MAX. The numeric range from 1 to 32767 still applies; however, it is used in a different way. The value specifies a size to which an inserted segment is padded when the compression of that segment results in a length less than the PAD value. This keyword is optional.

The following example shows how to enter the COMPRTN keyword of the SEGM statement for fixed-length segments:

```
SEGM NAME=SEGMENTA, PARENT=0, BYTES=50, PTR=TB,
      COMPRTN=(DRVRSCNM, DATA, INIT, MAX, PAD)
```

The following example shows how to enter the COMPRTN keyword of the SEGM statement for variable-length segments:
Build JCL requirements

The JCL for the Build function can be created from your ISPF panel input. You can also create your own JCL.

The following example shows a portion of the Build JCL. You can use this sample to help you develop your own JCL for the Build function.

```plaintext
//******************************************************************
//* BUILD FUNCTION
//******************************************************************
//STATS EXEC PGM=HCOZLDU0,PARM='dictnameBnnnn'
//STEPLIB DD DISP=SHR,
// DSN=whq.SHCLMOD
//SYSPRINT DD SYSOUT=*,DCB=RECFM=FBA
//SYSUDUMP DD SYSOUT=* *
//HDCDIN DD DISP=SHR,
// DSN=hlq.input.data
//HDCDIT DD DISP=SHR,
// DSN=hlq.input.data
//HDCDOUT DD DISP=(,CATLG,DELETE),
// DSN=hlq.dictname.OBJECT,
// SPACE=(TRK,(8,4)),
// UNIT=uuuu,VOL=SER=ssssss,
// DCB=(LRECL=80,BLKSIZE=3200,RECFM=FB)
//HDCDDICT DD DISP=(,CATLG,DELETE),
// DSN=hlq.dictname.HDCDDICT,
// SPACE=(TRK,(8,4)),
// UNIT=uuuu,VOL=SER=ssssss,
// DCB=(LRECL=4096,BLKSIZ=4096,RECFM=FB)
//HDCDSTAT DD DISP=(,CATLG,DELETE),
// DSN=hlq.dictname.HDCDSTAT,
// SPACE=(TRK,(1)),
// UNIT=uuuu,VOL=SER=ssssss,
// DCB=(LRECL=80,BLKSIZ=80,RECFM=FB)
```

The Build JCL references the following data sets:

**EXEC**  This statement must have the following format:

```plaintext
// EXEC PGM=HCOZLDU0,PARM='dictnameBnnnn'
```

where:

- **dictname**  Specifies the 1- to 8-character name of the dictionary.

- **B**  The letter B, as the ninth character in the parameter list, indicates that you are running the Build function.

- **nnnnn**  Specifies the value of the START BYTE option. This 5-character specification begins at character 10 in the parameter list. This specification is optional. It defaults to 0. Figure 11 on page 46 shows examples of use of this parameter.

**STEPLIB**  The input PDS that contains load module HCOZLDU0.

**HDCDIN**  The input data set that is used to build a dictionary.
**HDCDIT**

The name of the data set that is used to generate compression statistics. If this data set is different from the one that is specified in HDCDIN, statistics are generated for the HDCDIT data set’s space savings for the dictionary that was built from the HDCDIN data set. For example, you might want to create a dictionary from one volume of an HD unload data set but use a different volume of the same data set to generate the statistics.

**HDCDOUT**

The output data set that contains the generated dictionary in object format. This dictionary is linked with the dictionary driver when the compression exit routine is created.

**HDCDDICT**

The output data set that contains the generated dictionary in the z/OS format of 16 records of 4 KB each. This data set can be used later as input to the Examine function.

**HDCDSTAT**

The output data set that contains the compression statistics. This data set can be used later as input to the Examine function.

---

**Compression exit routine and DBD JCL requirements**

If you specified EDIT DBD SOURCE = Y on the Build panel, the Compression Exit Routine and DBD panels are displayed.

When you have made your changes, JCL for creating the compression exit routine and a new DBD are created for you.

The following example shows a portion the JCL that is used to specify the DBD for the Build function. You can use this sample to help you develop your own JCL for the Build function.

```plaintext
//********************************************************************
//* LINK DICTIONARY INTO LOAD LIBRARY
//********************************************************************
//** LINK DICTIONARY INTO LOAD LIBRARY
//********************************************************************
//LINK1 EXEC PGM=IEWL,
// PARM='SIZE=(180K,28K),RENT,REFR,NCAL,LET,XREF,LIST=ALL'
//SYSLIN DD DISP=SHR,
// UNIT=uuuu,VOL=SER=ssssss,
// DSN=hlq.dictname.OBJECT
//SYSLMD DD DISP=SHR,
// DSN=hlq.user.RESLIB(dictname)
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(1,1),RLSE)
//SYSPRINT DD SYSOUT=* 
//**
//** LINK DICTIONARY WITH HCOZLDX0 GENERIC DICTIONARY DRIVER
//********************************************************************
//** LINK DICTIONARY WITH HCOZLDX0 GENERIC DICTIONARY DRIVER
//********************************************************************
//LINK2 EXEC PGM=IEWL,COND=(4,LT,LINK1),
// PARM='SIZE=(880K,64K),RENT,REFR,NCAL,LET,XREF,LIST=ALL'
//SYSPRINT DD SYSOUT=* 
//DD1 DD DISP=SHR,
// DSN=hlq.user.RESLIB
//DD2 DD DISP=SHR,
// DSN=hco.hlq.SHCOLMOD
//SYSLMD DD DISP=SHR,
// DSN=hlq.user.RESLIB
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(10,1),RLSE)
//SYSLIN DD *
CHANGE dictname(HCOZHDCD) /* dictionary name must be first */
```
Compression Projection report (HCO002)

The Compression Projection report (HCO002) shows the compression analysis reports that project your data savings by using the compression dictionary that is specified for the Build function.

IMS prefix length and access method usage can be excluded from these statistics.

Before you run the Compress function, lengths might not match the DBD-specified lengths. If a segment with an odd length value is defined in the DBD, a hexadecimal zero pad byte is added at the end of each segment by the IMS Unload utility. If such records are used as input to HDCE, the lengths differ from the DBD-specified lengths.

The following figure shows the compression projection report produced by the Build function. IMS prefix length and access method usage can be excluded from these statistics.
The Compression Projection report that is generated by the Build function contains the following information:

**START BYTE**
- The value that is specified in the START BYTE parameter.

**DICT TYPE**
- The type of dictionary, which is always custom.

**DATA LENGTH FROM START BYTE**
- Shows the following columns:
  - **BEFORE COMPRESS MIN**
  - Smallest input segment size before compression.
  - **BEFORE COMPRESS AVG**
  - Average input segment size before compression.
  - **BEFORE COMPRESS MAX**
  - Largest input segment size before compression.
  - **AFTER COMPRESS MIN**
  - Smallest input segment size after compression.
  - **AFTER COMPRESS AVG**
  - Average input segment size after compression.
  - **AFTER COMPRESS MAX**
  - Largest input segment size after compression.
  - **SAVINGS AVG**
  - Average savings of bytes as determined by the following formula:
    \[(\text{BEFORE COMPRESS AVG} - \text{AFTER COMPRESS AVG})\]
    - Bytes that are skipped as a result of the START BYTE specification are excluded.
  - **SAVINGS PCT**
  - The percent of storage that is saved as determined by the following formula:
    \[(\text{PCT SAVINGS} = \frac{\text{AVG SAVINGS} \times 100}{\text{AVG BEFORE}})\]

The projected savings do not include IMS or access method usage.

**Example Table**

<table>
<thead>
<tr>
<th>START BYTE</th>
<th>DICT TYPE</th>
<th>BEFORE COMPRESS</th>
<th>AFTER COMPRESS</th>
<th>SAVINGS</th>
<th>INPUT RECORDS</th>
<th>BUILD RECORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MIN AVG MAX</td>
<td>MIN AVG MAX</td>
<td>AVG PCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Custom</td>
<td>38 52 76</td>
<td>9 18 42</td>
<td>34 65%</td>
<td>169</td>
<td>4,906</td>
</tr>
</tbody>
</table>

Figure 17. Compression Projection report

The Compression Projection report that is generated by the Build function contains the following information:

**START BYTE**
- The value that is specified in the START BYTE parameter.

**DICT TYPE**
- The type of dictionary, which is always custom.

**DATA LENGTH FROM START BYTE**
- Shows the following columns:
  - **BEFORE COMPRESS MIN**
  - Smallest input segment size before compression.
  - **BEFORE COMPRESS AVG**
  - Average input segment size before compression.
  - **BEFORE COMPRESS MAX**
  - Largest input segment size before compression.
  - **AFTER COMPRESS MIN**
  - Smallest input segment size after compression.
  - **AFTER COMPRESS AVG**
  - Average input segment size after compression.
  - **AFTER COMPRESS MAX**
  - Largest input segment size after compression.
  - **SAVINGS AVG**
  - Average savings of bytes as determined by the following formula:
    \[(\text{BEFORE COMPRESS AVG} - \text{AFTER COMPRESS AVG})\]
    - Bytes that are skipped as a result of the START BYTE specification are excluded.
  - **SAVINGS PCT**
  - The percent of storage that is saved as determined by the following formula:
    \[(\text{PCT SAVINGS} = \frac{\text{AVG SAVINGS} \times 100}{\text{AVG BEFORE}})\]

The projected savings do not include IMS or access method usage.
Bytes that are skipped as a result of the START BYTE specification are excluded.

**INPUT RECORDS**

The number of records that the Build function read from the HDCDIT data set to generate these statistics.

**BUILD RECORDS**

The number of input records that the Build function read from the HDCDIN data set to build the dictionary.
Chapter 8. Examining an existing dictionary

The Examine function is a reporting tool that lets you compare hardware data compression for alternate dictionaries.

The Examine function reports the compression savings that an existing IMS HD Compression Extended dictionary can provide when it is used to compress data that is different from the data that was used to create the dictionary.

The Examine function has the following two subfunctions:

**Examine Generic**
- This subfunction compresses segments or databases by using the generic dictionary. This subfunction produces the HCO003 report, which can help you evaluate whether the generic dictionary provides adequate compression savings.

**Examine Compare**
- This subfunction compresses segments or databases by using an existing custom dictionary. This subfunction produces the HCO004 report, which can help you evaluate the effectiveness of compression in the following situations:
  - Your enterprise or custom dictionary in compressing data that is different from the data that was originally used to create the dictionary.
  - Your dictionary in compressing data after a major update of your database. The Examine function can also be used to periodically check that compression of your data is optimal.

To allow selective compression, the Examine function has a START BYTE option, which lets you adjust where compression begins in the input records. You can also choose data selectively by skipping key data, unload-record prefix data, or specific fields in a segment. Figure 11 on page 46 show examples of the START BYTE option.

You run the Examine function by using the HCOZLDU0 program. It runs as a single-step, standard z/OS batch job.

**Topics:**
- “Examining a dictionary by using Examine”
- “Examine panel” on page 66
- “Examine JCL requirements” on page 68
- “Compression Projection report (HCO003)” on page 69
- “Compare Old/New Compression Projection report (HCO004)” on page 70

### Examining a dictionary by using Examine

Use the Examine function to produce reports about potential compression savings when you use an existing dictionary.
Before you begin

The Examine function supports all the acceptable data input sources.

Requirement: The output data sets HDCDDICT and HDCDSTAT must exist before a custom dictionary can be examined.

You can use either of the following methods to run the Examine function:
- From ISPF as a panel-driven function
- Directly from JCL as a standard z/OS batch job

About this task

To determine the ongoing effectiveness of the dictionary, use the old dictionary to compress a current data set of the same segment or database.

To determine if the dictionary could be used on another segment or database, use the dictionary to compress that data.

The following procedure describes how to run the utility from the ISPF panels.

Procedure

1. Provide an acceptable source of input data, or use the Examine function to access the Image Copy utility or the HD Unload utility directly. Acceptable input data includes the following sources:
   - Image Copy File
   - HD Unload File
   - Other - enter INPUT DSNAME (QSAM data set)
2. Run the Examine function from ISPF as a panel-driven function. Either complete the required fields on the Examine panel.
3. Specify the data set name in the INPUT DSNAME field to generate statistics.
4. Optional: For selective compression, adjust where compression begins in the input records by using the START BYTE option.
   You can skip key data, unload-record prefix data, or specific fields in a segment.
5. Generate the JCL by exiting the ISPF panel.
6. Optional: Edit the JCL if required.
7. Submit the generated JCL.
8. Interpret the output report to verify that the process completed successfully.

Results

The report that is created by the Examine function is sent to the SYSPRINT data set. The report that is produced depends on the type of database that is examined. The Compression Projection (HCO003) report is produced when the generic dictionary is examined. The Compare Old/New Compression Projection (HCO004) report is produced when a custom dictionary is examined.

Examine panel

You can use the Examine panel to specify the parameters that are required to examine an existing compression dictionary.
The Examine ISPF panel opens when you select the Examine function on the IMS HD Compression Extended main menu panel.

The Examine panel contains the following fields:

**DICTIONARY NAME**
Specify the name of the existing dictionary to be examined. Enter HCOGDIC1 if the generic dictionary is being examined. The name that you specify is used to find the existing dictionary and statistics data sets (HDCDDICT and HDCDSTAT) that are created by the Build function.

IMS HD Compression Extended uses *hlq.dictname*.HDCDDICT and *hlq.dictname*.HDCDSTAT as the naming convention for these data sets. Specify the high-level qualifier in the **USER DSN HLQ** field. If the dictionary and statistics data sets do not conform to this convention, you must change the JCL.

**INPUT SOURCE**
Specify the input to the Examine function.

**INPUT DSNAME**
Specify the fully qualified name of the input data set. If you specified the INPUT SOURCE as 1, you must enter an image copy file data set name. If you specified the INPUT SOURCE as 2, you must enter an HD unload file data set name. If you specified the INPUT SOURCE as 3, you must enter a variable-blocked QSAM data set.

**USER DSN HLQ**
Specify the high-level qualifier for the input dictionary and statistics data sets.

**START BYTE**
Specify the relative byte number in the input source record at which compression analysis begins. Its point of reference is the beginning of the record, not the end of an unload utility record prefix as is the case with the KEYEND parameter of the Detail Analyze function. Specifying a value in the...
START BYTE field lets you adjust where compression analysis begins in the input records. Figure 11 on page 46 and Figure 12 on page 46 show examples of this parameter.

DATABASE IDENTIFIER
Specify a valid DBD name to identify the database whose segments are being examined.

Examine JCL requirements

The JCL for the Examine function can be created from your ISPF panel input. You can also create your own JCL.

The following example shows the JCL for the Examine function. You can use this sample to help you develop your own JCL for the Examine function.

```plaintext
//**********************************************************************
//* EXAMINE FUNCTION
//**********************************************************************
//STATS EXEC PGM=HCOZLDU0,PARM='dictnameMnnnn'
//STEPLIB DD DISP=SHR,
// DSN=hcohlq.SHCOLMOD
//SYSPRINT DD SYSOUT=*,DCB=RECFM=FBA
//SYSUDUMP DD SYSOUT=* 
//HDCDIN DD DISP=SHR,
// DSN=hlq.input.data
//HDCDDICT DD DISP=SHR,
// DSN=hlq.dictname.HDCDDICT
//HDCDSTAT DD DISP=SHR,
// DSN=hlq.dictname.HDCDSTAT

The Examine JCL references the following data sets:

EXEC This statement must have the following format:
  // EXEC PGM=HCOZLDU0,PARM='dictnameMnnnn'

where:

dictname
  Specifies the 1- to 8-character name of the dictionary to be examined.

M
  The letter M, as the ninth character in the parameter list, indicates that you are running the Examine function.

nnnnn
  Specifies the value of the START BYTE option. This 5-character specification begins at character 10 in the parameter list. This specification can be omitted. It defaults to 0. Figure 11 on page 46 shows examples of this parameter.

STEPLIB
  The input PDS that contains load module HCOZLDU0.

HDCDIN
  The input data set.

HDCDDICT
  The name of the generic dictionary (hlq.HCOGDIC1.HDCDDICT) or your custom dictionary (hlq.dictname.HDCDDICT). Use this parameter to specify the dictionary that is to be examined. The dictionary must have the z/OS data set format of 16 records of 4 KB each. The Build function created this data set for the dictionary that you defined. IMS HD Compression
Extended provided this data set for the generic dictionary. Do not use the dictionary object data set form of either dictionary.

**HDCDSTAT**

The name of the custom dictionary statistics data set (hlq.dictname,HDCDSTAT). This data set was created by the Build function when the dictionary was created. Specify DD DUMMY when you examine the generic dictionary.

**Compression Projection report (HCO003)**

The Compression Projection report (HCO003) shows the compression projection report that is produced by the Examine function when the generic dictionary is being examined.

Before you run the Compress function, lengths might not match the DBD-specified lengths. If a segment with an odd length value is defined in the DBD, a hexadecimal zero pad byte is added at the end of each segment by the IMS Unload utility. If such records are used as input to HDCE, the lengths differ from the DBD-specified lengths.

The following figure shows the compression projection report produced by the Examine function when the generic dictionary is being examined. IMS prefix length and access method usage can be excluded from these statistics.

<table>
<thead>
<tr>
<th>START BYTE</th>
<th>DICT TYPE</th>
<th>BEFORE COMPRESS MIN</th>
<th>BEFORE COMPRESS AVG</th>
<th>BEFORE COMPRESS MAX</th>
<th>AFTER COMPRESS MIN</th>
<th>AFTER COMPRESS AVG</th>
<th>AFTER COMPRESS MAX</th>
<th>SAVINGS AVG</th>
<th>INPUT RECORDS</th>
<th>BUILD RECORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Generic</td>
<td>47</td>
<td>61</td>
<td>85</td>
<td>39</td>
<td>48</td>
<td>68</td>
<td>13</td>
<td>169</td>
<td>n/a</td>
</tr>
</tbody>
</table>

The projected savings do not include IMS or access method usage.

\[
\text{PCT SAVINGS} = \frac{\text{AVG SAVINGS} \times 100}{\text{AVG BEFORE}}
\]

**Better results will be achieved by building a custom dictionary.**

---

Figure 19. Compression Projection report

The Compression Projection report that is generated by the Examine function contains the following information:

**START BYTE**

The value specified in the START BYTE parameter.

**DATA LENGTH FROM START BYTE**

Shows the following columns:

**BEFORE COMPRESS MIN**

Smallest input segment size before compression.

**BEFORE COMPRESS AVG**

Average input segment size before compression.
BEFORE COMPRESS MAX
Largest input segment size before compression.

AFTER COMPRESS MIN
Smallest input segment size after compression.

AFTER COMPRESS AVG
Average input segment size after compression.

AFTER COMPRESS MAX
Largest input segment size after compression.

SAVINGS AVG
Average savings of bytes as determined by the following formula:
(BEFORE COMPRESS AVG - AFTER COMPRESS AVG)

Bytes that are skipped as a result of the START BYTE specification are excluded.

SAVINGS PCT
The percent of storage saved as determined by the following formula:
(PCT SAVINGS = AVG SAVINGS X 100 / AVG BEFORE)

Bytes that are skipped as a result of the START BYTE specification are excluded.

If the projected saving is less than 40%, consider creating a custom dictionary.

INPUT RECORDS
The number of records that the Examine function read to generate these statistics from the HDCDIT data set.

BUILD RECORDS
Not applicable.

Compare Old/New Compression Projection report (HCO004)

The Compare Old/New Compression Projection report (HCO004) shows the report that is produced by the Examine function when compression that uses a custom dictionary is compared to compression that uses the generic dictionary.

The following figure shows the compression projection report produced by the Examine function when a custom dictionary is being examined. IMS prefix length and access method usage can be excluded from these statistics.
The Compare Old/New Compression Projection report contains the following information:

SOURCE OF STATS

Shows the following columns:

OLD

Compression statistics that were produced when the Build function created the custom dictionary.

NEW

Compression statistics that were produced by the Examine function if the new input records were compressed by the existing custom dictionary.

These statistics let you compare the compression results of your dictionary against a different data set. Before you run the Compress function, lengths might not match the DBD-specified lengths. If a segment with an odd length value is defined in the DBD, a hexadecimal zero pad byte is added at the end of each segment by the IMS Unload utility. If such records are used as input to HDCE, the lengths differ from the DBD-specified lengths.

START BYTE

The value that is specified in the START BYTE parameter.

DATA LENGTH FROM START BYTE

Shows the following columns:

BEFORE COMPRESS MIN

Smallest input segment size before compression.

BEFORE COMPRESS AVG

Average input segment size before compression.

BEFORE COMPRESS MAX

Largest input segment size before compression.

AFTER COMPRESS MIN

Smallest input segment size after compression.

AFTER COMPRESS AVG

Average input segment size after compression.

AFTER COMPRESS MAX

Largest input segment size after compression.

Figure 20. Compare Old/New Compression Projection report
SAVINGS AVG
Average savings of bytes per segment as determined by the following formula:
(BEFORE COMPRESS AVG - AFTER COMPRESS AVG)

Bytes that are skipped as a result of the START BYTE specification are excluded.

SAVINGS PCT
The percent of storage saved as determined by the following formula:
(PCT SAVINGS = AVG SAVINGS X 100 / AVG BEFORE)

Bytes that are skipped as a result of the START BYTE specification are excluded.

INPUT RECORDS
The number of records that the Examine function read to generate these statistics.
Chapter 9. Compressing your data

After you create a dictionary and an IMS compression exit routine, the Compress function is the last of the IMS HD Compression Extended functions that you must run.

Compression is done when IMS reloads the database. The Compress function can create the JCL to reload your database if you are using one of the following reload utilities:

- DFSURG10 (for full function databases)
- FABCUR3 (for DEDB databases)

If you prefer to use your existing reload JCL, you can skip the instructions for using the Compress function.

Although a dictionary and an IMS compression exit routine can be built at any time, data compression requires an unload and reload of the database. The most recent database unload must be used as input to the reload. The unload must be in uncompressed format, and the database must not have been available for update since the unload. Otherwise, database integrity is jeopardized.

The Compress function does not produce any reports other than those that are produced by the reload utilities.

The ISPF panel functions for Compress help you create and submit the reload JCL.

Topics:

- "Compressing your data by using Compress"
- "Compress panel" on page 74
- "Compress JCL requirements" on page 75

Compressing your data by using Compress

Use the Compress function to reload a database by using the new DBD and to compress your data.

Before you begin

Restrictions:

- The restrictions of the reload utility you are using apply.
- The Compress function can be run only from ISPF panels.

About this task

The following procedure describes how to run the Compress function from the ISPF panels.

Procedure

1. Provide an acceptable unload data set. Provide a reload input data set (typically the unloaded database data set name).
2. Complete the required fields on the ISPF panel.
3. Generate the JCL by exiting the ISPF panel.
4. Optional: Edit the JCL if required.
5. Submit the generated JCL.
6. Check the reload job output to make sure that the job completed successfully.
7. Run an ACBGEN before making the database available to users.

Results
Output is the reloaded database data sets.

Compress panel

The Compress panel opens when you select the Compress function on the IMS HD Compression Extended main menu panel.

The Compress panel prompts you for the parameters that are needed to run compression. The following figure shows a sample of the Compress panel.

```
---- IMS HD Compression Extended --- Function = COMPRESS -----

COMMAND ===> RELOAD DATABASE USING THE NEW DBD
         + THESE FIELDS ARE OPTIONAL
         (Y OR N)
         TAPE INPUT ==>
         RELOAD INPUT DSN ==>
         IDCAMS CNTL DSNAME ==>
         IDCAMS CNTL MEMBER ==>
         IMS RESLIB DSN ==>
         USER RESLIB ==>
         NEW DBD LIBRARY DSN ==>
         NEW DBD NAME ==>
         DATABASE DSN ==>
         DATABASE DD NAME ==>
         * DB INDEX CLUSTER ==>
         * DB INDEX DD NAME ==>
         * DURWRK DSN ==>
         * DURDBDFN DSN ==>
         * IMS RECON1-3 PREFIX ==>

Press END to exit.
```

Figure 21. Compress panel

The Compress panel contains the following fields:

**TAPE INPUT**
Type Y if your reload input data set is on tape. Otherwise, specify N.

**RELOAD INPUT DSN**
Specify the name of the reload input data set (typically the unloaded database data set name).

**IDCAMS CNTL DSNAME**
Specify the name of the IDCAMS control data set.

**IDCAMS CNTL MEMBER**
Specify an IDCAMS control data set member that contains the IDCAMS DEFINE parameters for the database that is being reloaded.
Tip: Adjust the CI and size parameters to maximize compression. For more information, see “Segment Distribution by Compressed Length report (HCO007)” on page 41.

**IMS RESLIB DSN**
Specify the name of the IMS RESLIB data set.

**USER RESLIB**
Specify the name of the data set on which the IMS compression exit routine resides.

**NEW DBD LIBRARY DSN**
Specify the name of the data set that contains the new DBD.

**NEW DBD NAME**
Specify the name of the new DBD that was built by using the compression exit routine.

**DATABASE DSN**
Specify the name of the input data set that contains the database that is to be reloaded. If there are multiple data set groups, you must add DD statements for them when the JCL is opened to be edited.

**DATABASE DD NAME**
Specify the name of the database to be reloaded (used as a DD name in the DFSURG10 execution step). If you are reloading multiple data set groups, you must add DD statements for them when the JCL is opened to be edited.

**DB INDEX CLUSTER**
Specify the cluster name of the database to be reloaded (if required).

**DB INDEX DD NAME**
Specify the name of the index to be reloaded if required. You can use the name of the index that was used as a DD name in the DFSURG10 execution step.

**DURIWRK DSN**
Specify the name of the DURIWRK data set that is used by the FABCUR3 utility (for DEDB reload only).

**DURDBDFN DSN**
Specify the name of the DURDBDFN data set that is used by the FABCUR3 utility (for DEDB reload only).

**IMS RECON1-3 PREFIX**
Specify the data set name prefix of the three IMS RECON data sets (if they are required).

---

**Compress JCL requirements**

The JCL to reload the database is opened in ISPF edit. It consists of the basic code that is needed to support the standard DFSURG10 or FABCUR3 reload utilities. You might have to complete the JCL to meet your requirements.

For additional information, see the other IMS HD Compression Extended JCL requirements.
Chapter 10. Messages, abends, and return codes

Use the information in these messages to help you diagnose and solve IMS HD Compression Extended problems.

Messages

Use the information in these messages to help you diagnose, troubleshoot, and solve IMS HD Compression Extended problems.

IMS HD Compression Extended messages have the following format:

HKTnnnx text

where:

HKT
Indicates that the message was issued by IMS HD Compression Extended.

nnnn
Is the message identification number.

x
Indicates the severity of the message as follows:

• A indicates that operator intervention is required before processing can continue.
• E indicates that the job step is about to terminate abnormally.
• I indicates that the message is for information only.
• W indicates that the message is a warning to alert you to a possible error condition.

Message Variables

The message text can contain lowercase variables (for example, xxx...). The variables represent values when the message opens such as:

• Data in a data set
• A return code
• An error code

Message Documentation

In addition to message number and message text, information for each message includes the following information:

Explanation:
The Explanation section explains what the message text means, why it occurred, and what its variable entry fields are (if any).

System Action:
The System Action section explains what the system will do next.

User Response:
The User Response section describes whether a response is necessary, what the appropriate response is, and how the response will affect the system or program.

HCO3589E OPEN FAILED FOR CONTROL FILE
Explanation: After an OPEN macro was issued for a control statement data set, DCBOFLGS is not zero.
System action: A USER 3589 abend is issued.
**HCOP0001I • HCOP1440I**

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** Check for any obvious runtime errors.

**HCOP0001I  IMAGE COPY EXTRACT ENDED SUCCESSFULLY**

**Explanation:** HCOPMA\(nn\) issues message HCOP0001I when the utility successfully completes with a return code of 0. Check the report for the number of segments that were written and any unidentified data that was encountered. If a SEGIN control card is supplied that specifies specific segment names, the number of segments that were written might be less than the number of segments that were read.

**System action:** HCOPMA\(nn\) terminates the job with RC=0. \(nn\) is the IMS/ESA\(^a\) release (for example, 71, 81, or 91).

**Operator response:** None.

**Problem determination:** None.

**HCOP0003E  IMAGE COPY EXTRACT ENDED WITH ERRORS**

**Explanation:** HCOPMA\(nn\) detected an error when analyzing an input control statement. \(nn\) is the IMS/ESA release (for example, 71, 81, or 91).

**System action:** The utility terminates normally.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** Check the error message in the PROCCTL Statement Report.

**HCOP0412E  RBA EXCEEDED 4 GIGABYTE SIZE LIMIT**

**Explanation:** The database might be corrupted. The RBA of the database that was processed by the SCAN processor has exceeded the 4 GB size limit.

**System action:** Processing continues.

**Operator response:** Segments might exist beyond the 4 GB size limit, and the pointers that point to these segments are invalid.

**Problem determination:** Run IMS System Utilities/Data Base Tools HD Pointer Checker.

**HCOP1040I  NO ERRORS DETECTED**

**Explanation:** The process is running successfully.

**System action:** Processing continues.

**Operator response:** Examine the output report to check the number of segments output and unidentified data.

**Problem determination:** None.

**HCOP1070W  EMPTY DATA SET**

**Explanation:** The OSAM overflow part of the HISAM database is empty.

**System action:** Processing continues.

**Operator response:** None.

**Problem determination:** None.

**HCOP1370I  SCAN OF DB dbdname DSG xx IN PROGRESS TIME=hh:mm:ss**

**Explanation:** The SCAN processor began processing data set group \(xx\) of HISAM database \(dbdname\). The processing started at \(hh:mm:ss\).

**System action:** Processing continues.

**Operator response:** None.

**Problem determination:** None.

**HCOP1380I  SCAN OF DB dbdname DSG xx COMPLETED TIME=hh:mm:ss**

**Explanation:** The SCAN processor completed processing data set group \(xx\) of HISAM database \(dbdname\). The data set is empty. No errors were detected by the SCAN processor. The processing completed at \(hh:mm:ss\).

**System action:** Processing continues.

**Operator response:** None.

**Problem determination:** None.

**HCOP1390I  SCAN OF DB dbdname DSG xx EMPTY DATA SET TIME=hh:mm:ss**

**Explanation:** The processing of data set group \(xx\) of the HDAM or HIDAM database \(dbdname\) by the SCAN processor is complete. The data set is empty. The processing completed at \(hh:mm:ss\).

**System action:** Processing continues.

**Operator response:** None.

**Problem determination:** Check the report output. If the HISAM prime or overflow data set is expected to have data, check the image copy step.

**HCOP1440I  SCAN OF DB dbdname DSG: xx IN PROGRESS @ BLOCK xxx TIME=hh:mm:ss**

**Explanation:** The processing of data set group \(xx\) of the HDAM or HIDAM database \(dbdname\) by the SCAN processor reached block number \(xxx\) at \(hh:mm:ss\). This message provides status to help avoid inappropriate operator cancellation of the job.
HCOP1450I SCAN OF DB: dbdname DSG: xx
COMPLETED @ BLOCK xxx
TIME=hh.mm.ss

Explanation: The processing of data set group xx of the HDAM or HIDAM database dbdname by the SCAN processor reached block number xxx, and has completed. No errors were detected by the SCAN processor. The processing completed at hh.mm.ss.

System action: Processing continues.
Operator response: None.
Problem determination: None.

HCOP1450I SCAN OF DB: dbdname DSG: xx
COMPLETED @ BLOCK xxx
TIME=hh.mm.ss

Explanation: The processing of data set group xx of the HDAM or HIDAM database dbdname by the SCAN processor reached block number xxx, and has completed. No errors were detected by the SCAN processor. The processing completed at hh.mm.ss.

System action: Processing continues.
Operator response: None.
Problem determination: None.

HCOP1460I SCAN OF DB dbdname DSG xx EMPTY
DATA SET TIME=hh.mm.ss

Explanation: The processing of data set group xx of the HDAM or HISAM database dbdname by the SCAN processor has completed. The data set is empty. The processing completed at hh.mm.ss.

System action: Processing continues.
Operator response: None.
Problem determination: None.

HCOP1460I SCAN OF DB dbdname DSG xx EMPTY
DATA SET TIME=hh.mm.ss

Explanation: The processing of data set group xx of the HDAM or HISAM database dbdname by the SCAN processor has completed. The data set is empty. The processing completed at hh.mm.ss.

System action: Processing continues.
Operator response: None.
Problem determination: None.

HCOP1470E SCAN OF DB: dbdname DSG: xx
COMPLETED @ BLOCK xxx
TIME=hh.mm.ss  nnn ERRORS DETECTED

Explanation: The processing of data set group xx of the HDAM or HIDAM database dbdname by the SCAN processor reached block number xxx, and has completed. The processing completed at hh.mm.ss. nnn error messages were generated. These messages might be caused by slack bytes in the IMS image copy.

System action: Processing continues.
Operator response: None.
Problem determination: None.

HCOP1470E SCAN OF DB: dbdname DSG: xx
COMPLETED @ BLOCK xxx
TIME=hh.mm.ss  nnn ERRORS DETECTED

Explanation: The processing of data set group xx of the HDAM or HIDAM database dbdname by the SCAN processor reached block number xxx, and has completed. The processing completed at hh.mm.ss. nnn error messages were generated. These messages might be caused by slack bytes in the IMS image copy.

System action: Processing continues.
Operator response: None.
Problem determination: None.

HCOP1500W NO SEGMENTS WRITTEN TO OUTPUT FILE

Explanation: No data was extracted from the image copy data set. This situation might be due to an invalid image copy. If SEGIN was used to select segments, the segments of this type might not be available.

Operator response: Correct the error, and rerun the job.

Problem determination: The segment name might have been entered incorrectly, or the wrong IMS image copy data set might have been specified.

HCOP1520W ERRORS WERE DETECTED DURING IMAGE COPY EXTRACT EXECUTION

Explanation: A nonzero return code was received from the Extract function. Error messages are printed in reports.

System action: Processing continues with RC=2 or 4.
Operator response: Determine the causes of the error messages. Repair the database if necessary, and rerun the job.

Problem determination: See the output report from the Extract function.

HCOP2033E SEGMENT EXCEEDS BLOCK SIZE OF OUTPUT DATA SET

Explanation: The block size of the segment output data set HCOQSAM is too small. It must be 32 bytes larger than the largest segment in the database.

System action: A USER 2033 abend is issued.
Operator response: Correct the error, and rerun the job.

Problem determination: The block size can be up to the capacity of the output device but must be less than 32 KB for the HCOQSAM data set.

HCOP2034E UNABLE TO OPEN SEGMENT OUTPUT DATA SET

Explanation: A nonzero return code was received when the segment output data set HCOQSAM was opened.

System action: A USER 2034 abend is issued.
Operator response: Correct the error, and rerun the job.

Problem determination: Check the JCL parameters of the segment output data set HCOQSAM.

HCOP2051W DB dbdname DB# mm DD ddbname IS NOT SPECIFIED IN DATABASE STATEMENT FOR MULTIPLE DATA SET GROUP

Explanation: The multiple data set group of database dbdname was not specified in the DATABASE statement, but at least one multiple data set group was specified in the DATABASE statement. This message is a warning that not all data sets in this database are being processed.

System action: Processing continues.
Operator response: If one of the multiple data set
problems are omitted, segments in those data sets are not extracted.

**Problem determination:** None.

**HCOP2052W DB dbdname DB# nn IS NOT SPECIFIED IN DATABASE STATEMENT FOR LOGICALLY RELATED DB DBLG# xx**

**Explanation:** The database `dbdname` (database number `nn`) was not specified on a DATABASE statement, but at least one other database of the logically related database group `xx` was specified on the DATABASE statement. This message is also issued when the database type specified on the DBORG parameter prevents database `dbdname` (database number `nn`) from being processed, even if the database was specified on a DATABASE statement.

**System action:** Processing continues.

**Operator response:** This message is for informational purposes.

**Problem determination:** None.

**HCOP2055E Parm-value IS INVALID FOR Parm-name PARAMETER**

**Explanation:** An invalid control statement was detected during control statement syntax checking. The `Parm-value` is invalid for the `Parm-name` parameter.

**System action:** Processing stops.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** None.

**HCOP2058E LOGICAL DATABASE CANNOT BE PROCESSED**

**Explanation:** An invalid control statement was detected during control statement syntax checking. The `dbdname` in the DATABASE statement is the name of a logical DBD.

**System action:** Processing stops.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** The PSB might contain PCBs that reference logical DBDs. Because only physical databases are processed, the control statement must refer to the physical DBD.

**HCOP2059E SPECIFIED DATABASE WAS ALREADY FOUND IN ANOTHER DATABASE STATEMENT**

**Explanation:** An invalid control statement was detected during control statement syntax checking.

**HCOP2060E DDNAME SPECIFIED IN DD PARAMETER NOT FOUND IN DMB OR DDNAME SPECIFIED IN OVERFLOW PARAMETER NOT FOUND IN DMB**

**Explanation:** The DD name in the DD parameter or OVERFLOW parameter on the control statement is not present in the IMS DMB control block.

**System action:** Processing stops.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** The DD name might have been entered incorrectly on the control statement. It is also possible that the wrong PSB name was used on the EXEC statement PARM. This error can also occur if the utility is run under a different release of IMS than the one used to install it.

**HCOP2061E Parm-name PARAMETER IS INVALID FOR SPECIFIED DATABASE**

**Explanation:** An invalid control statement was detected during control statement syntax checking. The `Parm-name` parameter on the DATABASE statement is invalid.

**System action:** Processing stops.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** Check the DATABASE statement for spelling errors and to validate the combination of parameters.

**HCOP2062E DBD NAME SPECIFIED IN "DB" PARAMETER NOT FOUND IN DBD**

**Explanation:** The `dbdname` in the DB parameter on the control statement is not present in the IMS DBD control block.

**System action:** Processing stops.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** The `dbdname` might have been entered incorrectly on the control statement, or the wrong PSB name might have been specified on the
EXEC statement PARM. This error is also issued when the product is run under a release of IMS that is different from the one that is specified in the EXEC statement.

**HCOP2063E  DD STATEMENT NOT FOUND FOR DDNAME SPECIFIED IN DD PARAMETER OR DD STATEMENT NOT FOUND FOR DDNAME SPECIFIED IN OVERFLOW PARAMETER**

**Explanation:** No DD statement is present in the JCL that corresponds to the DD name that is specified in the DD parameter or in the OVERFLOW parameter on the control statement.

**System action:** Processing stops.

**Operator response:** Correct the JCL by adding the missing DD statement, and rerun the job. If no extract is required from the IMS overflow image copy data set, then provide a DD DUMMY statement with the correct DD name.

**Problem determination:** This error is probably an issue caused by invalid JCL. Check for spelling mistakes.

**HCOP2067E  INVALID NUMBER OF OPERANDS SPECIFIED FOR parm-name PARAMETER**

**Explanation:** An invalid control statement was detected during the control statement syntax checking. The number of operands in the parm-name parameter on the control statement is invalid.

**System action:** Processing stops.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** Check the control statement for invalid operands in the parameter.

**HCOP2069E  MORE THAN ONE PROC STATEMENT SPECIFIED**

**Explanation:** An invalid control statement was detected during the control statement syntax checking. More than one PROC statement is specified in the PROCCTL data set.

**System action:** Processing terminates.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** Only one PROC statement is allowed in the PROCCTL data set. The singular PROC statement must also be the first control statement in the PROCCTL data set.

**HCOP2070E  PROC STATEMENT NOT SPECIFIED**

**Explanation:** An invalid control statement was detected during the control statement syntax checking. No PROC statement is specified in the PROCCTL data set.

**System action:** Processing terminates.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** Only one PROC statement is allowed in the PROCCTL data set. The singular PROC statement must also be the first control statement in the PROCCTL data set.

**HCOP2071E  parm-name PARAMETER IS REQUIRED FOR SPECIFIED DATABASE**

**Explanation:** An invalid control statement was detected during the control statement syntax checking. The parm-name parameter is not specified on the DATABASE statement.

**System action:** Processing terminates.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** Specify the DD name (as coded in the DBD) of the overflow data set for HISAM. Specify the dbdname of the primary database indexed by the HIDAM index or by the secondary index database in the PRIMEDB parameter.

**HCOP2073E  MISSING DATABASE STATEMENT**

**Explanation:** An invalid control statement was detected during the control statement syntax checking. The data set that is associated with the PROCCTL DD statement has no DATABASE statement, or there was no DATABASE statement that preceded an END statement.

**System action:** Processing terminates.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** The DATABASE statement must be specified.

**HCOP2074E  MISSING CONTROL STATEMENT**

**Explanation:** An invalid control statement was detected during the control statement syntax checking. The data set that is associated with the PROCCTL DD statement was empty or it was a DUMMY data set.

**System action:** Processing terminates.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** None.
HCOP2075E  MORE THAN ONE OPTION STATEMENT SPECIFIED AFTER PROC OR DATABASE STATEMENT
Explanation: An invalid control statement was detected during the control statement syntax checking. Option statements are not valid.
System action: Processing terminates.
Operator response: Correct the error, and rerun the job.
Problem determination: None.

HCOP3504E  MESSAGE TEXT NOT FOUND
Explanation: An error message cannot be found in the message table in module HCOUMSGS.
System action: A USER 3504 abend is issued.
Operator response: Verify that HCOPMAnn, including all current maintenance, is installed correctly. Reinstall HCOPMAnn, including all maintenance, if necessary. Then rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM Software Support.
Problem determination: Either HCOPMAnn or its maintenance might have been installed incorrectly.

HCOP3505E  INVALID MESSAGE FLAG
Explanation: An attempt was made to print an error message, and an invalid flag was supplied to module HCOUMSGS.
System action: A USER 3505 abend is issued.
Operator response: Verify that HCOPMAnn, including all current maintenance, is installed correctly. Reinstall HCOPMAnn, including all maintenance, if necessary. Then rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM for support.
Problem determination: Either HCOPMAnn or its maintenance might have been installed incorrectly.

HCOP3509E  NUMBER OF ENTRIES FOR BLOCK MAP EXCEED SIZE OF GETMAIN AREA
Explanation: A database block (or control interval) contains more segments and free space elements than the module internal tables allow.
System action: A USER 3509 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM for support.
Problem determination: Verify whether there are too many segments in the block. Otherwise, check for any obvious runtime errors.

HCOP3510E  MORE THAN MAXIMUM DBDS FOUND IN PSB: psbname
Explanation: The PSB used in the EXEC statement contains more database PCBs than is allowed for the IMS release being used.
System action: A USER 3510 abend is issued.
Operator response: Correct PSBGEN, and rerun the job.
Problem determination: Make sure the correct PSBLIB is being used for the IMS version being used. This error is also issued when the product is run under a release of IMS that is different from the one that is specified in the EXEC statement.

HCOP3513E  INVALID USER STATISTICS INTERVAL SPECIFIED
Explanation: An internal error occurred.
System action: A USER 3513 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all reports) and call IBM Software Support. Check for any obvious runtime errors.
Problem determination: None.

HCOP3570E  FIND FAILED FOR DBD IN DBDLIB
Explanation: A FIND macro was issued for the dbdname, but no module with that name was in the library specified on the IMS DD statement.
System action: A USER 3570 abend is issued.
Operator response: Correct the error, and rerun the job.
Problem determination: This error is probably caused by invalid JCL. Make sure the DBD library is part of the IMS DD statement. Check the control statement for spelling errors.

HCOP3571E  SEGMENT CODE NOT FOUND IN DBD SEGMENT TABLE
Explanation: An internal error occurred.
System action: A USER 3571 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM Software Support.
Problem determination: Check for any obvious runtime errors. Make sure the DBD and PSB are valid.
HCOP3572E  EXPECTED PHYSICAL PARENT NOT IN SEGMENT INFORMATION TABLE

**Explanation:** An internal error occurred.

**System action:** A USER 3572 abend is issued.

**Operator response:** Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM Software Support.

**Problem determination:** Check for any obvious runtime errors. Make sure the DBD and PSB are valid.

HCOP3574E  PHYSICAL PARENT NOT FOUND IN SEGMENT INFORMATION TABLE

**Explanation:** An internal error occurred.

**System action:** A USER 3574 abend is issued.

**Operator response:** Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM Software Support.

**Problem determination:** Check for any obvious runtime errors. Make sure the DBD and PSB are valid.

HCOP3575E  DATABASE NAME SPECIFIED IN CONTROL STATEMENT NOT FOUND IN IMAGE COPY HEADER

**Explanation:** The dbdname on the control statement is not contained in the first record of the IMS image copy data set.

**System action:** A USER 3575 abend is issued.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** The dbdname entered on the control statement, or the IMS image copy data set that was used might be wrong. The IMS image copy data set might be corrupted.

HCOP3576E  ERROR IN DETERMINING END OF BLOCK IN BUFFER OR BAD FSE

**Explanation:** An internal error occurred.

**System action:** A USER 3576 abend is issued.

**Operator response:** Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM Software Support.

**Problem determination:** Check for any obvious runtime errors. Make sure the DBD and PSB are valid.

HCOP3584E  OSAM DATA SET IS LARGER THAN 2 (or 4) GIGABYTES

**Explanation:** An OSAM data set group that is larger than 2 GB was that was processed (4 GB for IMS/ESA Version 3).

**System action:** A USER 3584 abend is issued.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** The input data set being used might be invalid.

HCOP3586E  OPEN FAILED FOR PRINT FILE

**Explanation:** After issuing an OPEN macro for a printer data set, DCBOFLGS is not zero.

**System action:** A USER 3586 abend is issued.

**Operator response:** Correct the error, and rerun the job.

**Problem determination:** Check for any obvious runtime errors.

HCOP3590E  INVALID POINTER OR SEGMENT CODE IN SEGMENT STATISTICS TABLE

**Explanation:** An internal error occurred.

**System action:** A USER 3590 abend is issued.

**Operator response:** Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM Software Support.

**Problem determination:** Check for any obvious runtime errors. Make sure the DBD and PSB are valid.

HCOP3600E  NO DMB FOUND IN DMB DIRECTORY

**Explanation:** An internal error occurred.

**System action:** A USER 3600 abend is issued.

**Operator response:** Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all reports) and call IBM Software Support.

**Problem determination:** Check for any obvious runtime errors. Make sure the DBD and PSB are valid.

HCOP3601E  LOAD FAILED FOR DDNAME ddname MODULE module-name

**Explanation:** After issuing a LOAD macro to load module module-name from the library that is specified by ddname on the DD statement, the register 15 contained a nonzero return code.
HCOP3603E • HCOP3671E

System action: A USER 3601 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. This error occurs if the compression exit routine cannot be loaded from the steplib concatenation. If the problem persists, save the entire run listing (including the dump, JCL, and all reports) and call IBM Software Support.
Problem determination: Check that the STEPLIB DD card specifies the library that contains the pertinent compression exit routines.

HCOP3603E GETMAIN FAILED. SIZE nn K
ITEM-ID xx
Explanation: After issuing a GETMAIN macro to acquire SIZE nn K bytes of storage, the register 15 contained a nonzero return code. ITEM-ID xx represents the internal location that is issuing a GETMAIN macro.
System action: A USER 3603 abend is issued.
Operator response: Increase the region size parameter on the JOB statement, and rerun the job.
Problem determination: None.

HCOP3604E FREEMAIN FAILED. ITEM-ID: xx
Explanation: An internal error occurred in HCOPMA or HCOPSDA0.
System action: A USER 3604 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all reports) and call IBM Software Support.
Problem determination: Check for any obvious runtime errors.

HCOP3605E OPEN FAILED FOR DDNAME ddname
Explanation: After issuing an OPEN macro to open the data set that is associated with the specified ddname, the register 15 contained a nonzero return code.
System action: A USER 3605 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all reports) and call IBM for support.
Problem determination: Make sure that the DD statement for ddname correctly indicates that the proper data set has been provided.

HCOP3606E CLOSE FAILED FOR DDNAME ddname
Explanation: An internal error occurred in module HCOPACTL.
System action: A USER 3606 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all reports) and call IBM Software Support.
Problem determination: Check for any obvious runtime errors.

HCOP3650E INVALID CALL TO U1212AV
Explanation: An internal error occurred in module HCOPSDA0.
System action: A USER 3650 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all reports) and call IBM for support.
Problem determination: Check for any obvious runtime errors. Make sure that the DBD and PSB are valid.

HCOP3669E DATABASE AND DATA SET GROUP MUST BE IN DMB ORDER
Explanation: The control statements are in the wrong sequence.
System action: A USER 3669 abend is issued.
Operator response: Correct the error, and rerun the job.
Problem determination: Check the DMB Directory report to find the correct order. The control statements must be in the same order as specified in the DMB Directory report.

HCOP3671E OFFSET OF KEY IN VLS ROOT SEGMENT CANNOT BE ZERO
Explanation: An internal error occurred in module HCOPBLBD.
System action: A USER 3671 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all reports) and call IBM for support.
Problem determination: Check for any obvious runtime errors. Make sure that the DBD and PSB are valid.
HCOP3672E LOGICAL CHILD NOT FOUND IN SEGMENT INFORMATION TABLE

Explanation: An internal error occurred.
System action: A USER 3672 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMA* reports) and call IBM for support.
Problem determination: Check for any obvious runtime errors. Make sure that the DBD and PSB are valid.

HCOP3673E LOGICAL PARENT NOT FOUND IN SEGMENT INFORMATION TABLE

Explanation: An internal error occurred.
System action: A USER 3673 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMA* reports) and call IBM for support.
Problem determination: Check for any obvious runtime errors. Make sure that the DBD and PSB are valid.

HCOP3674E SEGMENT CODE OF INDEX TARGET NOT FOUND IN DMB DIRECTORY

Explanation: An internal error occurred in module HCOPLBLD.
System action: A USER 3674 abend is issued.
Operator response: Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMA* reports) and call IBM for support.
Problem determination: Check for any obvious runtime errors. Make sure that the DBD and PSB are valid.

HCOP3830E TWO DIFFERENT LEVELS OF IMS ARE BEING USED: **mmm** IMS LEVEL OF THIS RUN (REGION CONTROLLER) **nnn** IMS LEVEL OF THIS HCO ASSEMBLY

Explanation: The IMS batch region controller (DFSRC00) being used is at a different level than the IMS macros used in the image copy extract assembly. **mmm** and **nnn** represent the IMS level.
System action: Results are unpredictable. Typically an immediate 0C4 or other abend is issued.
Operator response: Run module HCOPMA* with the correct version of IMS.

Problem determination: If different levels of IMS are installed, check the JOBLIB or STEPLIB and the name of the HCOPMA* (for example, HCOPMA71, HCOPMA81, or HCOPMA91) program that was run. The last two digits are the IMS release that HCOPMA* was assembled on.

HCOP3999E UNKNOWN ERROR OCCURRED IN module-name MODULE (**CC = nnn**) 

Explanation: An internal error occurred in module-name module.
System action: A USER 3999 abend is issued.
Operator response: Save the entire run listing (including the dump, JCL, and reports) and call IBM Software Support.
Problem determination: None.

HCOP8001E CONTROL STATEMENT FORMAT ERROR

Explanation: An invalid control statement was detected during the control statement syntax checking.
System action: Processing stops.
Operator response: Correct the error, and rerun the job.
Problem determination: Check the control statement.

HCOP8002E parm-name IS INVALID AS STATEMENT PARAMETER

Explanation: An invalid control statement was detected during the control statement syntax checking.
System action: Processing stops.
Operator response: Correct the error, and rerun the job.
Problem determination: Check the control statement.

HCOP8003E parm-name PARAMETER IS INVALID FOR ctl-stmt-name STATEMENT

Explanation: An invalid control statement was detected during the control statement syntax checking.
System action: Processing stops.
Operator response: Correct the error, and rerun the job.
Problem determination: Check the control statement.

HCOP8004E NUMBER OF parm-name PARAMETERS EXCEEDED THE LIMIT, MAX IS **nn**

Explanation: An invalid control statement was detected during the control statement syntax checking.
System action: Processing stops.
HCOP8005E • HCOZ0004E

Operator response: Correct the error, and rerun the job.

Problem determination: Check the control statement.

HCOP8005E parm-name PARAMETER IS REQUIRED FOR ctl-stmt-name STATEMENT

Explanation: An invalid control statement was detected during the control statement syntax checking.

System action: Processing stops.

Operator response: Correct the error, and rerun the job.

Problem determination: Check the control statement.

HCOP8006E NUMBER OF OPERAND FOR parm-name PARAMETER EXCEED THE LIMIT, MAX IS nn

Explanation: An invalid control statement was detected during the control statement syntax checking.

System action: Processing stops.

Operator response: Correct the error, and rerun the job.

Problem determination: Check the control statement.

HCOP8007E LENGTH ERROR IN n-th OPERAND OF PARAMETER parm-name

Explanation: An invalid control statement was detected during the control statement syntax checking.

System action: Processing stops.

Operator response: Correct the error, and rerun the job.

Problem determination: Check the control statement.

HCOP8008E n-th OPERAND IS REQUIRED FOR PARAMETER parm-name

Explanation: An invalid control statement was detected during the control statement syntax checking.

System action: Processing stops.

Operator response: Correct the error, and rerun the job.

Problem determination: Check the control statement.

HCOP8999E UNKNOWN ERROR OCCURRED IN module-name MODULE (CC = nn)

Explanation: An internal error occurred in module-name module.

System action: A USER 8999 abend is issued.

Operator response: Save the entire run listing (including the dump, JCL, and HCOPMAnn reports) and call IBM Software Support.

Problem determination: None.

HCOZ0001I cccccccc HDC DICTIONARY BUILT IN DD HDCDOUT FILE

Explanation: HCOZLDU0 issues the message HCOZ0001I when the utility successfully completes with a return code of 0. cccccccc is the dictionary name that was passed as a parameter to the utility.

System action: The utility terminates normally.

Operator response: None.

System programmer response: None.

Problem determination: No error.

HCOZ0002E cccccccc HDC DICTIONARY NOT BUILT, BECAUSE SOURCE DATA DOES NOT MATCH EXPANDED DATA. THE FIRST 10 MISMATCHED RECORDS WILL BE PRINTED BEFORE THE COMPRESSION STATISTICS

Explanation: Message HCOZ0002E is issued when mismatched source and expanded data is detected. A dictionary object data set is not built. cccccccc is the dictionary name that was passed as a parameter to the utility.

System action: The utility terminates normally.

Operator response: None.

System programmer response: Contact your DBA or IBM service.

Problem determination: No error.

HCOZ0003W HDC DICTIONARY NOT BUILT, BECAUSE MVS CSRCMPSC SERVICE IS NOT INSTALLED

Explanation: Message HCOZ0003W is issued when the MVS™ compression and expansion service CSRCMPSC is not installed. An HDC dictionary cannot be built.

System action: The utility terminates normally.

Operator response: None.

System programmer response: Install MVS CSRCMPSC service, and rerun the utility.

Problem determination: No error.

HCOZ0004E HDC DICTIONARY NOT BUILT, BECAUSE MVS CSRCMPSC SERVICE HAD A LOGIC ERROR

Explanation: Message HCOZ0004E is issued when a logic error is detected while using the MVS...
CSRMPSC compression and expansion service macro.
An HDC dictionary cannot be built.

**System action:** The utility terminates normally.

**Operator response:** None.

**System programmer response:** Contact IBM support.

**Problem determination:** No error.

---

**HCOZ0005E**  
**ddname** INPUT FILE EMPTY OR INCOMPLETE. PROCESSING TERMINATED

**Explanation:** Message HCOZ0005E is issued when an empty data set is detected. The ddname of the empty data set is named in the message.

**System action:** The utility terminates normally.

**Operator response:** Notify the DBA.

**System programmer response:** None.

**Problem determination:** Check the output report for details of the errors.

---

**HCOZ0006E**  
HCOGDIC1 DICTIONARY NAME RESERVED FOR “EXAMINE” USE ONLY

**Explanation:** Message HCOZ0006E is issued if the generic dictionary reserved name of HCOGDIC1 is used with the Build or Analyze function. The HCOGDIC1 name can only be used for the Examine function.

**System action:** The utility terminates normally.

**Operator response:** Notify the DBA.

**System programmer response:** None.

**Problem determination:** None.

---

**HCOZ0007E**  
ssssssss HAD INVALID SYSIN CONTROL INPUT. SEE REPORT OUTPUT.

**Explanation:** HCOZLDH0 issues this message when invalid, duplicate or no segment names are supplied in the SYSIN input data set.

HCOZLDE0 issues this message when invalid segment names, key end, fixed data, or variable data is read in the SYSIN input data set.

ssssssss is the SYSIN segment name.

**System action:** The utility terminates before dictionary processing begins.

**Operator response:** None.

**System programmer response:** None.

**Problem determination:** None.

---

**HCOZ0008E**  
START BYTE SPECIFIED IN PARAMETER LONGER THAN QSAM LL FIELD. PROGRAM TERMINATED.

**Explanation:** HCOZLDU0 issues this message if a record shorter than the value specified in the START BYTE parameter is detected in either the HDCDIT (statistics generation) or HDCDIN (dictionary creation) data set.

**System action:** The utility terminates before dictionary creation or analysis is completed.

**Operator response:** None.

**System programmer response:** None.

**Problem determination:** If the HDCDIN or the HDCDIT data set is an IMS Unload database data set, the short record is probably an unload header or trailer record. Use the HDCE preparation utility HCOZLDH0 to remove these records and any segment records not needed for the dictionary. If the HDCDIN or HDCDIT record has been created from another source, ensure that the START BYTE parameter correctly reflects the record lengths in the input data sets.

---

**Abend codes**

The explanations and user responses provided in this abend code reference can help you diagnose, troubleshoot, and solve IMS HD Compression Extended problems.

For each abend code, the following information is provided where applicable:

**Explanation:**

The Explanation section explains what the abend code means, why it occurred, and what its variable entry fields are (if any)

**System Action:**

The System Action section explains what the system will do next...
Reason Code 1 • Reason Code 6

User Response:
The User Response section describes whether a response is necessary, what
the appropriate response is, and how the response will effect the system or
program

Abend U3315
Abend issued for HCOZLDH0, HCOZLDE0, and HCOZLDU0.

Abend code U3315 is issued with the following reason codes:

Reason Code 1
Explanation: The utility failed to get storage in the
initialize dictionary module, or an internal error
occurred in that module.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Analyze the reason
code.
Problem determination: A storage shortage occurred.
Increase the region size and rerun the utility, and if the
problem persists contact IBM Software Support.

Reason Code 2
Explanation: The utility failed because it passed
invalid parameters to the build work dictionary
module.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Contact IBM Software
Support.
Problem determination: Invalid parameters were
passed to the build work dictionary module (an
internal logic error occurred).

Reason Code 3
Explanation: The utility failed because it passed
invalid parameters to the build dictionary module.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Contact IBM Software
Support.
Problem determination: Invalid parameters were
passed to the build dictionary module (an internal logic
error occurred).

Reason Code 4

Abend U3325
Abend issued for HCOZLDX0 and HCOZLDXV.
Abend code 3325 is used by the compression exit routine and issues one of the following reason codes. The reason code can be found in the last byte of Register 14.

Reason Code 1
Explanation: Invalid function code was passed to the compression macro.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Contact IBM Software Support.
Problem determination: A logic error occurred while attempting to perform data compression or expansion.

Reason Code 2
Explanation: Sequence field is not completely in the segment.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Contact IBM Software Support.
Problem determination: A logic error occurred while attempting to perform data compression or expansion.

Reason Code 3
Explanation: Compression or expansion input length is negative.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Contact IBM Software Support.
Problem determination: A logic error occurred while attempting to perform data compression or expansion.

Reason Code 4
Explanation: INIT processing is not set for open.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Notify the database administrator.
Problem determination: A logic error occurred while attempting to perform data compression or expansion. Check that the link edit for the IMS segment edit or compression exit has been performed correctly.

Reason Code 5
Explanation: No dictionary address.

Reason Code 6
Explanation: Dictionary not on page boundary.
System action: The utility terminates abnormally.
Operator response: Notify the database administrator.
System programmer response: Notify the database administrator.
Problem determination: A logic error occurred while attempting to perform data compression or expansion. Check that the link edit for the IMS segment edit or compression exit has been performed correctly.

Reason Code 7
Explanation: No eye catcher in dictionary.
System action: The utility terminates abnormally.
Operator response: Notify the database administrator.
System programmer response: Notify the database administrator.
Problem determination: A logic error occurred while attempting to perform data compression or expansion. Check that the link edit for the IMS segment edit or compression exit has been performed correctly.

Reason Code 8
Explanation: No compression service available.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Ensure that the application is running on a machine that supports hardware data compression.
Problem determination: An error occurred while attempting to access the hardware data compression microcode or the MVS software simulation code.

Reason Code 9
Explanation: GETMAIN failure for full function database call.
System action: The utility terminates abnormally.
Reason Code 10 • Reason Code F

Operator response: Notify the system programmer.
System programmer response: Contact IBM Software Support.
Problem determination: A logic error occurred while attempting to perform data compression or expansion.

Reason Code 10
Explanation: Minimum segment size for Fast Path database is invalid.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Notify the database administrator.
Problem determination: An invalid segment length was passed to the data compression process. Check that the DBD is correct for the database minimum and maximum segment lengths.

Reason Code A
Explanation: GETMAIN failure for DEDB Fast Path database call.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Contact IBM Software Support.
Problem determination: A logic error occurred while attempting to perform data compression or expansion.

Reason Code B
Explanation: Compression input length < 2 bytes.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Notify the database administrator.
Problem determination: A segment shorter than 2 bytes was detected during compression processing.

Reason Code C
Explanation: Expansion input length < 2 bytes
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Notify the database administrator.
Problem determination: A segment shorter than 2 bytes was detected during expansion processing.

Reason Code D
Explanation: Hardware instruction issued a nonzero return code.
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Contact IBM Software Support.
Problem determination: A logic error occurred while attempting to perform data compression or expansion.

Reason Code E
Explanation: Software simulation issued nonzero return code
System action: The utility terminates abnormally.
Operator response: Notify the system programmer.
System programmer response: Contact IBM Software Support.
Problem determination: A logic error occurred while attempting to perform data compression or expansion.

Reason Code F
Explanation: INIT processing is not set for close.
System action: The utility terminates abnormally.
Operator response: Notify the database administrator.
System programmer response: Notify the database administrator.
Problem determination: A logic error occurred while attempting to perform data compression or expansion. Check that the DBD is correct for the database access requirements and database type.
Abends for HCOPMAnn

Abend U2033  The block size of the segment output data set is too small. It must be 14 bytes larger than the largest segment in the database.

**Explanation:** The abend is accompanied by the following message:
HCOP2033E - SEGMENT EXCEEDS BLOCK SIZE OF OUTPUT DATA SET

**Operator response:** Correct the error and rerun the job.

**Problem determination:** The block size can be up to the capacity of the output device, but must be less than 32 KB for the HCOQSAM data set.

Abend U2034  Nonzero return code was received from open on segment output data set HCOQSAM.

**Explanation:** The abend is accompanied by the following message:
HCOP2034E - UNABLE TO OPEN SEGMENT OUTPUT DATA SET

**Operator response:** Correct the error and rerun the job.

**Problem determination:** Check the JCL parameters of the segment output data set HCOQSAM.

Abend U3504  An error message cannot be found in the message table in module HCOUMSGS.

**Explanation:** The abend is accompanied by the following message:
HCOP3504E - MESSAGE TEXT NOT FOUND

**Operator response:** Verify that HCOPMAnn, including all current maintenance, is installed correctly. Reinstall HCOPMAnn, including all maintenance, if necessary, then rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all HCOPMAnn reports) and call IBM Software Support.

**Problem determination:** Either HCOPMAnn or its maintenance might have been installed incorrectly.

Abend U3509  A database block (or control internal) contains more segments and free space elements than module internal tables allow.

**Explanation:** The abend is accompanied by the following message:
HCOP3509E - NUMBER OF ENTRIES FOR BLOCK MAP EXCEED SIZE OF GETMAIN AREA

**Operator response:** Correct any obvious errors, and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL and all HCOPMAnn reports) and call IBM Software Support.

**Problem determination:** Verify whether there are too many segments in the block. Otherwise, check for any obvious runtime errors.

Abend U3510  The PSB in which the number of database data sets is referred to in PSB psbname must be less than 256. This error is often caused by running with a different level of IMS than the one you assembled with.

**Explanation:** The abend is accompanied by the following message:
HCOP3510E - MORE THAN MAXIMUM DBDS FOUND IN PSB: +psbname%

**Operator response:** Correct the PSBGEN and rerun the job.

**Problem determination:** Make sure that the correct PSBLIB is being used for the IMS version that is being used. This error is also issued when the product is run under a release of IMS that is different from the one that is specified in the EXEC statement.

Abend U3513  An internal error occurred.

**Explanation:** The abend is accompanied by the following message:
HCOP3513E - INVALID USER STATISTICS INTERVAL SPECIFIED

**Operator response:** Correct any obvious errors and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, reports) and call IBM Software Support. Check for any obvious runtime error.

**Problem determination:** None.
Abend U3570 • Abend U3600

Abend U3570 A FIND macro was issued for the 
dbdname, but no module with that name 
was in the library specified on the IMS 
DD statement.

Explanation: The abend is accompanied by the 
following message:
HCOP3570E - FIND FAILED FOR DBD IN DBOLIB

Operator response: Correct the error and rerun the 
job.

Problem determination: This error is typically caused 
by invalid JCL. Make sure that the DBD library is part 
of the IMS DD statement. Check the control statement 
for spelling errors.

Abend U3571 An internal error occurred.

Explanation: The abend is accompanied by the 
following message:
HCOP3571E - SEGMENT CODE NOT FOUND IN DBD 
SEGMENT TABLE

Operator response: Correct any obvious errors and 
rerun the job. If the problem persists, save the entire 
run listing (including the dump, JCL and all 
HCOPMAnn reports) and call IBM Software Support.

Problem determination: Check for any obvious 
runtime errors. Make sure that the DBD and the PSB 
are valid.

Abend U3572 An internal error occurred.

Explanation: The abend is accompanied by the 
following message:
HCOP3572E - EXPECTED PHYSICAL PARENT NOT IN 
SEGMENT INFORMATION TABLE

Operator response: Correct any obvious errors and 
rerun the job. If the problem persists, save the entire 
run listing (including the dump, JCL and all 
HCOPMAnn reports) and call IBM Software Support.

Problem determination: Check for any obvious 
runtime errors. Make sure that the DBD and the PSB 
are valid.

Abend U3574 An internal error occurred.

Explanation: The abend is accompanied by the 
following message:
HCOP3574E - PHYSICAL PARENT NOT IN SEGMENT 
INFORMATION TABLE

Operator response: Correct any obvious errors and 
rerun the job. If the problem persists, save the entire 
run listing (including the dump, JCL and all 
HCOPMAnn reports) and call IBM Software Support.

Problem determination: Check for any obvious 
runtime errors. Make sure that the DBD and the PSB 
are valid.

Abend U3575 The 
dbdname on the control statement 
is not contained in the first record of the 
image copy data set.

Explanation: The abend is accompanied by the 
following message:
HCOP3575E - DATABASE NAME SPECIFIED IN CONTROL 
STATEMENT NOT FOUND IN IMAGE COPY HEADER

Operator response: Correct the error and rerun the 
job.

Problem determination: An incorrect 
dbdname might 
be on the control statement, or an incorrect image copy 
data set might have been used. The image copy data 
data set might be corrupted. Check for any obvious runtime 
errors. Make sure that the DBD and the PSB are valid.

Abend U3576 An internal error occurred

Explanation: The abend is accompanied by the 
following message:
HCOP3576E - ERROR IN DETERMINING END OF BLOCK IN 
BUFFER OR BAD FSE

Operator response: Correct any obvious errors and 
rerun the job. If the problem persists, save the entire 
run listing (including the dump, JCL and all 
HCOPMAnn reports) and call IBM Software Support.

Problem determination: Check for any obvious 
runtime errors. Make sure that the DBD and the PSB 
are valid.

Abend U3590 An internal error occurred.

Explanation: The abend is accompanied by the 
following message:
HCOP3590E - INVALID POINTER OR SEGMENT CODE IN 
SEGMENT STATISTIC TABLE

Operator response: Correct any obvious errors and 
rerun the job. If the problem persists, save the entire 
run listing (including the dump, JCL and all 
HCOPMAnn reports) and call IBM Software Support.

Problem determination: Check for any obvious 
runtime errors. Make sure that the DBD and the PSB 
are valid.

Abend U3600 An internal error occurred.

Explanation: The abend is accompanied by the 
following message:
HCOP3600E - NO DMB FOUND IN DMB DIRECTORY

Operator response: Correct any obvious errors and 
rerun the job. If the problem persists, save the entire 
run listing (including the dump, JCL and all reports) 
and call IBM Software Support.

Problem determination: Check for any obvious 
runtime errors. Make sure that the DBD and the PSB 
are valid.
Abend U3601  After issuing a LOAD macro to load
module module-name from the library
specified by ddname on the DD
statement, register 15 contained a
nonzero return code.

Explanation:  The abend is accompanied by the
following message:
HCOP3601E - LOAD FAILED FOR DDNAME+ddname+%MODULE
+module-name

Operator response:  Correct any obvious errors and
rerun the job. If the problem persists, save the entire
run listing (including the dump, JCL and all reports)
and call IBM Software Support.

Problem determination:  Make sure that the DD
statement for ddname correctly indicates the proper data
set has been provided.

Abend U3606  An internal error occurred in
HCOPACTL.

Explanation:  The abend is accompanied by the
following message:
HCOP3606E - CLOSE FAILED FOR DDNAME+ddname%

Operator response:  Correct any obvious errors and
rerun the job. If the problem persists, save the entire
run listing (including the dump, JCL and all reports)
and call IBM Software Support.

Problem determination:  Check for any obvious
runtime errors.

Abend U3650  An internal error occurred in
HCOPSDA0.

Explanation:  The abend is accompanied by the
following message:
HCOP3650E - INVALID CALL TO U1212AV

Operator response:  Correct any obvious errors and
rerun the job. If the problem persists, save the entire
run listing (including the dump, JCL and all reports)
and call IBM Software Support.

Problem determination:  Check for any obvious
time errors. Make sure that the DBD and the PSB
are valid.

Abend U3671  An internal error occurred in
HCOPPLBLD.

Explanation:  The abend is accompanied by the
following message:
HCOP3671E - OFFSET OF KEY IN VLS ROOT SEGMENT
CANNOT BE ZERO

Operator response:  Correct any obvious errors and
rerun the job. If the problem persists, save the entire
run listing (including the dump, JCL and all reports)
and call IBM Software Support.

Problem determination:  Check for any obvious
runtime errors. Make sure that the DBD and the PSB
are valid.

Abend U3672  An internal error occurred.

Explanation:  The abend is accompanied by the
following message:
HCOP3672E - LOGICAL CHILD NOT FOUND IN
SEGMENT INFORMATION TABLE
Abend U3673 • Abend U8999

Operator response: Correct any obvious errors and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL and all HCOPMAnn reports) and call IBM Software Support

Problem determination: Check for any obvious runtime errors. Make sure that the DBD and the PSB are valid.

Abend U3673  An internal error occurred.
Explanation: The abend is accompanied by the following message:
HCOP3673E - LOGICAL PARENT NOT FOUND IN SEGMENT INFORMATION TABLE

Operator response: Correct any obvious errors and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL and all HCOPMAnn reports) and call IBM Software Support

Problem determination: Check for any obvious runtime errors. Make sure that the DBD and the PSB are valid.

Abend U3674  An internal error occurred in HCOPLBLD.
Explanation: The abend is accompanied by the following message:
HCOP3674E - SEGMENT CODE OF INDEX TARGET NOT FOUND IN DMB DIRECTORY

Operator response: Correct any obvious errors and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL and all HCOPMAnn reports) and call IBM Software Support

Problem determination: None.

Abend U8999  An internal error occurred in the indicated module.
Explanation: The abend is accompanied by the following message:
HCOP8999E - UNKNOWN ERROR OCCURRED IN+module-name%MODULE (CC = nnn)

Operator response: Save the entire run listing (including the dump, JCL and reports) and call IBM Software Support.

Problem determination: None.

Return codes

IMS HD Compression Extended messages are often accompanied by return codes.

From the Image Copy Extract utility

The following return codes are issued by the Image Copy Extract utility:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The utility ended successfully and issued the accompanying HCOZ0001I message.</td>
</tr>
<tr>
<td>2</td>
<td>Warning messages were issued, but the requested operation completed. If HCOP1520W was issued, slack bytes were encountered and caused return code 2. Message HCOP0003E is issued.</td>
</tr>
<tr>
<td>4</td>
<td>The utility did not execute successfully. No output was written due to warnings. Message HCOP0003E is issued.</td>
</tr>
<tr>
<td>8</td>
<td>The utility did not execute successfully. Operations were skipped or unsuccessfully executed because severe errors were detected. No output was written to the segment output data set because of errors. Message HCOP0003E is issued.</td>
</tr>
</tbody>
</table>
From the HD Unload Extract utility

The following return codes are issued by the HD Unload Extract utility:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The utility ended successfully.</td>
</tr>
<tr>
<td>8</td>
<td>The utility did not execute successfully. No data was found for the segments that are specified in the SYSIN control statements.</td>
</tr>
<tr>
<td>20</td>
<td>The utility ended unsuccessfully and issued the accompanying HCOZ0005E message because an input data set was empty. Processing could not continue.</td>
</tr>
<tr>
<td>24</td>
<td>The utility ended unsuccessfully and issued the accompanying HCOZ0006E message. The SYSIN segment name was invalid or missing.</td>
</tr>
</tbody>
</table>

From the Detail Analyze function

The following return codes are issued by the Detail Analyze function:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The utility ended successfully.</td>
</tr>
<tr>
<td>12</td>
<td>The utility ended unsuccessfully and issued the accompanying HCOZ0003W message because MVS CSRCMPSC is not installed on the machine.</td>
</tr>
<tr>
<td>20</td>
<td>The utility ended unsuccessfully and issued the accompanying HCOZ0005E message because an input data set is empty.</td>
</tr>
<tr>
<td>24</td>
<td>The utility ended unsuccessfully and issued the accompanying HCOZ0006E message. The SYSIN segment name was invalid or missing.</td>
</tr>
</tbody>
</table>

From the Analyze, Build, and Examine functions

The following return codes are issued by the Analyze, Build, and Examine functions:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The utility ended successfully and issued the accompanying HCOZ0001I message.</td>
</tr>
<tr>
<td>8</td>
<td>The utility ended successfully and issued the accompanying HCOZ0002E message, but did not build a dictionary object data set because data integrity checks were detected between a source sequential variable-length input record and its equivalent re-expanded record.</td>
</tr>
<tr>
<td>12</td>
<td>The utility ended unsuccessfully and issued the accompanying HCOZ0003W message because z/OS CSRCMPSC is not installed on the machine.</td>
</tr>
<tr>
<td>20</td>
<td>The utility ended unsuccessfully and issued the accompanying HCOZ0005E message because an input data set was empty. Processing could not continue.</td>
</tr>
</tbody>
</table>

Gathering diagnostic information

Before you report a problem with IMS HD Compression Extended to IBM Software Support, gather the appropriate diagnostic information.
Procedure

Provide the following information for all IMS HD Compression Extended problems:

- A clear description of the problem and the steps that are required to re-create the problem
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of IMS that you are using and the type and version of the operating system that you are using

Provide additional information based on the type of problem that you experienced:

For online abends, provide the following information

- A screen capture of the panel that you were using when the abend occurred
- The job log from the TSO session that encountered the abend
- The job log from the server
- A description of the task that you were doing before the abend occurred

For errors in batch processing, provide the following information

- The complete job log
- Print output
- Contents of any data sets that were used during the processing
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright IBM Corp. 1996, 2014
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

**Programming interface information**

This publication documents information that is NOT intended to be used as Programming Interfaces of IMS HD Compression Extended.

This book is intended to help you use IMS HD Compression Extended Version 1 Release 3.
This book also documents Associated Guidance Information and Diagnosis, Modification or Tuning Information, which is provided to help you administer compression of IMS databases.

Warning: Do not use this Diagnosis, Modification or Tuning Information as a programming interface.

This book documents no programming interfaces for use by customers in writing programs that request or receive the services of IMS HD Compression Extended Version 1 Release 3.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml.

Other company, product, and service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

Index

A
abend codes 87
ACBGEN 3, 53
accessibility overview 14
accounting information changing 13
Analyze function
description 9, 43
execution 11
generating statistics 43
HCOZLDU0 program 11
input and output data sets 43, 46
JCL 43, 46
panel 45
report 48
using 43
Analyze panel 45
analyzer component overview 6

B
Build function
creating a compression exit routine 53
description 9, 51
execution 11
generating statistics 52
HCOZLDU0 program 11
input and output data sets 52, 60
JCL generating 52
requirements for Build function 60
requirements for compression exit routine and DBD 61
modifying or creating a DBD 53
panels Build 55
Compression Exit Routine 57
Edit DBD 58
report 62
restrictions 52
using 52
Build panel 55

C
codes 87
comments methods for providing xi
Compare Old/New Compression Projection report 70
components 6
Compress function
description 9, 73
execution 11
HCOZLDX0 program 11, 54
HCOZLDXV program 11, 54
Compress function (continued)
JCL 75
panel 74
restrictions 73
using 73
Compress panel 74
compression exceptions 53
compression exit routine creating 53
description 3
JCL 61
relinking 54
Compression Exit Routine panel 57
Compression Projection report for Analyze 48
for Build 62
for Examine 69
COMPRVTN keyword 53, 59
conventions highlighting ix
custom dictionary 7
customized database dictionary 7
customized segment dictionary 7

D
data sets 23
HCOQSAM 24
HDCDDICT
for Build 60
for Detail Analyze 38
for Examine 68
HDDCIT 45, 60
HDDCOUT 60
PRIMAPRT 24
PROCCTL 27
QSAM 23
SEGIN 27
SGOUTnnn 30, 32
STATIRT 24
SYSIN 32
data sharing support scenario 3
database administration change management comprehensive solutions 2
comprehensive solutions 2
database administration and change management solutions 2
DBD (database description)
description 3
JCL 61
modifying or creating 53
modifying the MAXBYTES field 53
DEDB unload data set 43
Detail Analyze function
description 9, 35
execution 11
generating statistics 35
HCOZLD60 program 11
input and output data sets 35, 38
Detail Analyze function (continued)
JCL 35, 38
panel 37
reports
Detail Segment Compression Analysis overview 35
Segment Distribution by Compressed Length 41
using 35
Detail Analyze panel 37
device types changing 13
DFSURG10 utility 73
diagnostic information gathering 96
dictionary creating 52
customized database 7
customized segment 7
tenure 7
generic 7
input data 52
object data set 7
strategies 7
z/OS data set 7
Ziv-Lempel technique 5
dictionary driver 3
documentation accessing 13

E
Edit DBD panel 58
enterprise dictionary 7
error messages 77
Examine function
description 9, 65
execution 11
generating statistics 66
HCOZLDU0 program 11
input and output data sets 66, 68
JCL 66, 68
panel 67
reports 69, 70
restrictions 66
using 66
exceptions, compression 53
exit routines, compression 53
Extract function
description 9, 23
HD Unload Extract utility 30
Image Copy Extract utility 23

F
FABCUR3 utility 73
feedback methods for providing xi
restrictions (continued)

Image Copy Extract utility 24
return codes 94
REXX procedure 20

S
sample members
HCORLNK 54
scenarios
changing accounting information 13
changing device types 13
screen readers and magnifiers 14
SEGIN data set 27
SEGM statement 53, 59
Segment Compression Analysis
report 35, 40
Segment Distribution by Compressed Length report 35, 41
segments
maximum length 53
not compressible 53
selecting for extraction 24
selective compression 52, 66, 67
service information ix
SGOUTnnn data set 30, 32
space considerations 3
START BYTE option
Analyze function 43
Build function 51
Examine function 65
starting
IMS HD Compression Extended 20
STATIPRT data set 24
statistics
Analyze function 43
Build function 52
Detail Analyze function 35
strategies for dictionary 7
support
required information 96
support information ix
syntax diagrams
how to read x
SYSIN data set 32

T
technotes 13
trademarks 99

U
uncompressable data 53
upgrading
Tools Base 54
user-written unload program 5
utilities
DFSURG10 73
FABCUR3 73
HD Unload Extract 30
Image Copy Extract 23

Z
z/OS
data set 7
hardware data compression 1
Ziv-Lempel, explained 5