Note:
Before using this information and the product it supports, read the "Notices" topic at the end of this information.
<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 28. Contents of the sample library file (SIHCSAMP)</td>
<td>315</td>
</tr>
<tr>
<td>Part 10. Troubleshooting</td>
<td>319</td>
</tr>
<tr>
<td>Chapter 29. Messages and codes</td>
<td>321</td>
</tr>
<tr>
<td>Return and abend codes</td>
<td>322</td>
</tr>
<tr>
<td>Trace file or trace panel</td>
<td>323</td>
</tr>
<tr>
<td>ISPF messages</td>
<td>323</td>
</tr>
<tr>
<td>Chapter 30. Gathering diagnostic information</td>
<td>355</td>
</tr>
<tr>
<td>Part 11. Appendixes</td>
<td>357</td>
</tr>
<tr>
<td>Notices</td>
<td>359</td>
</tr>
<tr>
<td>Index</td>
<td>363</td>
</tr>
</tbody>
</table>
About this information

IBM® IMS™ Database Utility Solution for z/OS® IMS High Availability Large Database Toolkit (formerly IBM IMS High Availability Large Database Conversion and Maintenance Aid for z/OS, and hereafter referred to as IMS HALDB Toolkit) provides application enabling features and system utilities for the improved management and operation of the IMS HALDB environment.

These topics provide instructions for using IMS HALDB Toolkit.

To use the procedures in this information, you must first install IMS HALDB Toolkit as described in the Program Directory for IMS Database Utility Solution for z/OS, GI13-4194, and then perform the postinstallation steps as described in the IMS Database Utility Solution: Overview and Customization, SC27-8778.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform the following tasks:
- Use IMS HALDB Toolkit to set up environments
- Define and process a conversion project
- Maintain IMS HALDB Toolkit
- Use the DBRC function for HALDBs
- Interpret reports issued by IMS HALDB Toolkit
- Troubleshoot IMS HALDB Toolkit problems

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

Always refer to the IMS Tools Product Documentation web page for complete product documentation resources:


The IMS Tools Product Documentation web page includes:
- Links to IBM Knowledge Center for the user guides ("HTML")
- PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Recent updates to the user guides, referred to as "Tech docs" ("See updates to this book!"
- Technical notes from IBM Software Support, referred to as "Tech notes"
- White papers that describe product business scenarios and solutions
Part 1. Introduction to IMS HALDB Toolkit

IBM IMS Database Utility Solution for z/OS IMS High Availability Large Database (HALDB) Toolkit (also referred to as IMS HALDB Toolkit) provides application enabling features and system utilities for the improved management and operation of the IMS HALDB environment.

Topic:

• Chapter 1, “IMS HALDB Toolkit overview,” on page 3
Chapter 1. IMS HALDB Toolkit overview

IMS HALDB Toolkit provides HALDB functions that can be used for HALDB maintenance and operation tasks. Product features include application enabling features and system utilities for the improved management and operation of the IMS HALDB environment.

A High Availability Large Database (HALDB) is a partitioned full-function DL/I database. The supported database organizations are PHDAM, PHIDAM, and PSINDEX.

IMS databases are escalating in size because of regulatory compliance requirements and because businesses are serving more customers and using more suppliers. IMS HALDB Toolkit can help reduce the time and minimize the skills that are required to perform application support tasks and provide IMS HALDB maintenance, modeling, and analysis.

IMS HALDB Toolkit helps you to maintain 24x7 data availability. The tool’s application functions enable applications to make better use of the HALDB environment. The tool’s system utilities provide capabilities that can help improve HALDB serviceability.

Topics:
- “IMS HALDB Toolkit features and benefits” on page 4
- “Usage and restrictions” on page 8
- “Service updates and support information” on page 9
- “Product documentation and updates” on page 10
- “Accessibility features” on page 12
IMS HALDB Toolkit features and benefits

IMS HALDB Toolkit helps to administer databases through a HALDB conversion with ease and efficiency. IMS HALDB Toolkit helps reduce administrative costs by easing the conversion from full-function to HALDB and assisting in transporting HALDBs to IMS test systems.

In addition, IMS HALDB Toolkit provides the following features and functions:

- **Batch conversion and maintenance offline**
  
  The database to be converted must have IMS dynamic allocation member (MDA) descriptions.

- **Simulation of new partition settings and simulation of conversion from full-function database to HALDB**

- **JCL, created by DBDLIB conversion, to convert entire DBD libraries**

  DBDLIB conversion uses batch conversion to create JCL for each primary database.

- **Offline reorganization of PSINDEX databases, including index pointer healing and reclaiming of VSAM space**

- **PSINDEX records of deleted partitions removed (DELETE PARTITION)**

  It is not necessary to rebuild the PSINDEX by scanning all partitions.

- **Performance improvement because PSINDEX records are sorted and inserted sequentially (ADD PARTITION, using LOAD processing)**

- **Merge identical HALDBs and combine user-partitioned databases into one multi-partitioned HALDB (MERGE HALDB)**

IMS HALDB Toolkit dynamically detects the existence of the following tools and uses the tools as needed:

- **IMS High Performance Image Copy for z/OS**

- **IMS High Performance Load for z/OS**

- **IMS High Performance Pointer Checker for z/OS**

- **IMS High Performance Unload for z/OS**

- **IMS Library Integrity Utilities for z/OS**

If any of these tools are unavailable on the system, IMS HALDB Toolkit replaces them with an appropriate IMS utility (if they are available).

**Application support**

IMS HALDB Toolkit provides the following support for applications:

**Dynamic DFSHALDB statement build**

Dynamic management of the DFSHALDB statement is allowed by providing a key for the starting partition and a key for the ending partition.

When you specify a key range, the utility transforms the key range into a starting and ending partition. Therefore, the application does not require a change when partitions are split or consolidated.

**Partition Selection API**

A callable interface is provided to assign a key to a partition.

This API allows applications to split their input on a partition boundary for parallel processing.
The API returns the partition name and the partition number. The API does not require a DL/I environment.

Converting to HALDB

IMS HALDB Toolkit supports the following functions to help you convert full-function databases to HALDBs:

Simulate conversion to a HALDB
Before you convert the full-function database, simulate the HALDB partition settings to determine the most effective partition settings.

Conversion to a HALDB
You can convert to HALDB using a single step batch process.

You can alternatively convert to HALDB using the ISPF user interface to generate all required steps. Help panels are available for IMS HALDB Toolkit functions while you are in an ISPF session.

Maintaining HALDBs

IMS HALDB Toolkit supports the following functions to help you maintain HALDBs:

Simulate repartitioning of a HALDB
Before you consolidate or split partitions, simulate the new partitions to determine the most effective partition settings. This utility simulates the effect of the new partition settings with ease.

Consolidate or split partitions
As the data grows or shifts, partitions grow or shrink. This utility allows you to consolidate or split partitions with ease.

Index Pointer Healer
After reorganizing the primary database, all secondary index pointers require healing (the new self healing with HALDB). However, this task might demand extra resources that could impact performance during peak time.

This utility heals the pointer offline and at a less expensive time.

Load a single partition
Databases with secondary indexes must insert the secondary indexes in random sequence when loading. This utility delays the index insert to the end of the load process, then loads the indexes in sequence.

This method improves the elapsed time.

Delete a single partition
When a single partition is deleted, the secondary indexes must be rebuilt. This utility deletes all records from the secondary indexes that reference the deleted partition.

The entire process is recoverable and fast.

Merge HALDBs
When user partitioning (multiple identical databases on different key ranges) has been used, the conversion will first transform the databases to HALDB, and then merge them together using this process.

This task is necessary when you require secondary indexing and was not possible when using multiple databases.
Add an empty partition to the end of a database
This function is appropriate for growth that occurs mainly at the end of an existing database.

DBRC handling
IMS HALDB Toolkit supports the following functions for DBRC handling:

Cloning DBRC definitions
The clone function replicates HALDB DBRC definitions to other RECONs and allows you to exchange the high-level data set name qualifier.

Copy a HALDB to a different IMS system
The copy function transfers a production HALDB to a different IMS and allocates the target data set in that system, if the data set is not already available.

Backup DBRC definitions
The backup function saves database-related DBRC definitions. The definitions can be used as input for the batch DBRC utility (DSPURX00).

For more information, see Chapter 12, “HALDB without DBRC,” on page 179.

HALDB Analyzer
IMS HALDB Toolkit supports the HALDB Analyzer utility to help you analyze HALDBs:

Analyze HALDB constructs
This function analyzes the HALDB structure and verifies that all root keys are in the correct partition, that all ILKs are valid, and that all EPS pointers can find the correct ILE. Additionally, performance-related data is collected.

This utility can be used to check the health of a HALDB.

You can provide thresholds that trigger the utility to inform you when a maintenance function is required.

Extract root keys
This utility allows the extracting of all root keys to a sequential file. A record layout is provided.

System utilities
IMS HALDB Toolkit provides the following utilities to help you with database handling tasks:

Split unload file
This utility allows you to split an unload file into single partition unload files. The reloads then can run in parallel.

The single partition unload files are standard Hierarchical Direct (HD) unload files with header and trailer statistics.

Each file contains only records for the selected partition.

Indirect list key (ILK) rebuild
This utility enables you to rebuild all ILKs and their references, and allows you to resolve conflicting ILKs.

The databases must be offline for this activity.
ACBLIB reference
This utility provides a list of all PSBs that reference a given database, and assists in online changes by identifying which PSBs require change.

Create DBD source
This utility re-creates the DBD source statements from the DBDLIB.
Usage and restrictions

There are important usage and restrictions for the IMS HALDB Toolkit product.

Logical relationships and secondary indexes

IMS HALDB Toolkit converts databases and secondary indexes with the following logical relationships:

- **Unidirectional logical relationships** remain as unidirectional.
- **Bidirectional logical relationships with physical pairing** remain as bidirectional with physical pairing.
- **Bidirectional logical relationships with virtual pairing** are converted to bidirectional relationships with physical pairing.
- **Shared secondary indexes** are converted to single secondary indexes.
- **Non-unique secondary indexes** are converted to unique secondary indexes by adding the /SX field to the SUBSEQ parameters of the XDFLD.

A few installations have defined bidirectional logical relationships without any pairing even though this capability is not documented in the IMS product. IMS HALDB Toolkit does not support a conversion of these relationships to HALDB.

When running the CONVERT function as a batch job, DBDs with logical relationships (except for indexes) are not allowed. To convert DBDs with logical relationships, the ISPF interface must be used because the database and all logically related databases must be unloaded before the conversion can be done. Batch CONVERT does only one unload followed by one convert at a time. The ISPF interface can be used to construct the JCL such that all unloads are to be done before the conversion.

The CONVERT function does work for DBDs with logically related secondary indexes. The CONVERT function always converts the related non-HALDB Secondary Index to a HALDB Secondary Index (PSINDEX for example). This conversion from non-HALDB Secondary Index to a HALDB Secondary Index is so that the secondary index remains usable. If the secondary index is no longer required, one must remove it from the DBD and run DBDGEN again before running the CONVERT function.

Database limitations

The following limitations apply to IMS HALDB Toolkit database conversions:

- **SHISAM, HSAM, and DEDB databases** cannot be converted.
- **HISAM databases** are converted to PHIDAM.

The MAINTAIN function does not support HALDB M-V data sets.

The MAINTAIN function processes PSINDEXes only when the PSINDEXes are offline.
Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:

Product documentation and updates

IMS Tools information is available at multiple places on the web. You can receive updates to IMS Tools information automatically by registering with the IBM My Notifications service.

Information on the web

Always refer to the IMS Tools Product Documentation web page for complete product documentation resources:


The IMS Tools Product Documentation web page includes:

- Links to IBM Knowledge Center for the user guides (“HTML”)
- PDF versions of the user guides (“PDF”)
- Program Directories for IMS Tools products
- Recent updates to the user guides, referred to as "Tech docs" ("See updates to this book!")
- Technical notes from IBM Software Support, referred to as "Tech notes"
- White papers that describe product business scenarios and solutions

IBM Redbooks® publications that cover IMS Tools are available from the following web page:

http://www.redbooks.ibm.com

The IBM Information Management System website shows how IT organizations can maximize their investment in IMS databases while staying ahead of today’s top data management challenges:

https://www.ibm.com/software/data/ims/

Receiving documentation updates automatically

To automatically receive automated emails that notify 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 Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications 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 Notifications page is displayed, click Subscribe to select those products that you want to receive information updates about. The IMS Tools option is located under Software > Information Management.
4. Click Continue to specify the types of updates that you want to receive.
5. Click Submit to save your profile.
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 IBM product documentation, use one of the following options:

- Use the IBM Knowledge Center Contact Us link.
- Use the online reader comment form, which is located at [http://www.ibm.com/software/data/rcf/](http://www.ibm.com/software/data/rcf/)
- Send your comments by email to comments@us.ibm.com Include the name of the book, the part number of the book, the version of the product that you are using, and, if applicable, the specific location of the text you are commenting on, for example, a page number or table number.
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 HALDB Toolkit enable users to perform the following activities:

- 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. See 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.
Part 2. Simulating HALDBs

By using the ANALYZEPART function, you can simulate HALDB partition settings to determine the effective partition settings.

The ANALYZEPART function is supported in batch mode and it can be used for offline databases.

With the ANALYZEPART function, you can:

• Simulate the HALDB partition settings before you convert a full-function database to a HALDB with the CONVERT function
• Simulate the new partition settings before you perform maintenance on HALDB partitions with the MAINTAIN function

Topic:
• Chapter 2, “Simulating HALDB using ANALYZEPART,” on page 15
Chapter 2. Simulating HALDB using ANALYZEPART

Use the ANALYZEPART command to simulate the effect of new partition settings or conversion from a full-function database to a HALDB in batch mode.

Topics:
- “ANALYZEPART data flow” on page 16
- “Simulating conversion to a HALDB” on page 17
- “Simulating repartitioning of a HALDB” on page 18
- “Simulating conversion or repartitioning using unloaded data sets as input” on page 19
- “ANALYZEPART DD statements” on page 21
- “ANALYZEPART command parameters” on page 24
- “ANALYZEPART examples” on page 33
ANALYZE PART data flow

The following figure shows the data flow for the ANALYZE PART function.

![Diagram of ANALYZE PART data flow]

**Figure 1. ANALYZE PART data flow**

The simulation result is written to the Collect DBDs report in the IHCLIST data set. If you request to generate DBRC commands, the DBRC commands are written to the data set pointed to from the `dbrcout` DD statement.
Simulating conversion to a HALDB

You can simulate conversion from a full-function database to a HALDB offline by using the `ANALYZEPART` command.

About this task

By simulating the effect of the HALDB partition settings, you can determine the most effective partition settings before you convert a full-function database to a HALDB by using the `CONVERT` command.

For a database with logical relationships, only the database that is specified on the DBD parameter is simulated. All the logically related databases are not simulated.

Procedure

1. Locate the sample JCL members in the SIHCSAMP file, members IHCEMAIx.
2. Specify the JCL DD statements. For a list of DD statements, see "ANALYZEPART DD statements" on page 21.
3. Specify the ANALYZEPART command parameters. For a list of ANALYZEPART command parameters, see "ANALYZEPART command parameters" on page 24.

   The following table lists some of the common scenarios and ANALYZEPART parameters for simulating HALDB partition settings:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>ANALYZEPART parameters</th>
<th>KEYSIN input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulate conversion to a HALDB</td>
<td><code>ANALYZEPART DBD(xxxx)</code></td>
<td></td>
</tr>
<tr>
<td>that has four partitions</td>
<td><code>PARTNUM(4)</code></td>
<td>Not applicable</td>
</tr>
<tr>
<td>Simulate conversion to a HALDB</td>
<td><code>ANALYZEPART DBD(xxxx)</code></td>
<td></td>
</tr>
<tr>
<td>that has 512 MB size partitions</td>
<td><code>PARTSIZE(512)</code></td>
<td>Not applicable</td>
</tr>
<tr>
<td>Simulate conversion to a HALDB</td>
<td><code>ANALYZEPART DBD(xxxx)</code></td>
<td></td>
</tr>
<tr>
<td>using your own key boundaries</td>
<td><code>KEYS(KEYSIN)</code></td>
<td><code>//KEYSIN DD *</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>smaller-high-key</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>middle-high-key</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>last-high-key</code></td>
</tr>
</tbody>
</table>

   You can also refer to the following examples to code ANALYZEPART JCL statements and command parameters:
   - “ANALYZEPART example: Specifying the number of partitions” on page 34
   - “ANALYZEPART example: Specifying the partition size” on page 35
   - “ANALYZEPART example: Specifying the key ranges for partitions” on page 36
   - “ANALYZEPART example: Generating DBRC commands” on page 37
   - “ANALYZEPART example: Using unloaded data sets as input” on page 39
Simulating repartitioning of a HALDB

You can simulate repartitioning of a HALDB offline by using the **ANALYZEPART** command.

**About this task**

By simulating the effect of the new partition settings, you can determine the most effective partition settings before you consolidate or split partitions by using the **MAINTAIN** command.

For a database with logical relationships, only the database that is specified on the DBD parameter is simulated. All the logically related databases are not simulated.

**Procedure**

1. Locate the sample JCL members in the SIHCSAMP file, members IHCEMAIx.
2. Specify the JCL DD statements. For a list of DD statements, see "ANALYZEPART DD statements" on page 21.
3. Specify the ANALYZEPART command parameters. For a list of ANALYZEPART command parameters, see "ANALYZEPART command parameters" on page 24.

   The following table lists some of the most common functions and ANALYZEPART parameters:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>ANALYZEPART parameters</th>
<th>KEYSIN input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulate consolidation of two partitions into one using PARTNUM</td>
<td>ANALYZEPART <strong>DBD(****) - PARTLIST(part1,part2) - PARTNUM(1)</strong></td>
<td>Not applicable</td>
</tr>
<tr>
<td>Simulate splitting of one partition into two using PARTNUM</td>
<td>ANALYZEPART <strong>DBD(****) - PARTITION(part1) - PARTNUM(2)</strong></td>
<td>Not applicable</td>
</tr>
<tr>
<td>Simulate splitting of one large partition into 2048 MB size partitions using PARTSIZE</td>
<td>ANALYZEPART <strong>DBD(****) - PARTITION(bigpart1) - PARTSIZE(2048)</strong></td>
<td>Not applicable</td>
</tr>
<tr>
<td>Simulate splitting of one partition into three using your own key boundaries</td>
<td>ANALYZEPART <strong>DBD(****) - PARTITION(part1) - KEYS(KEYSIN)</strong></td>
<td>//KEYSIN DD * smaller-high-key middle-high-key last-high-key</td>
</tr>
<tr>
<td>Simulate repartitioning with 2048 MB size partitions for all the partitions using PARTSIZE</td>
<td>ANALYZEPART **DBD(*<strong><em>) - PARTITION(</em>) - PARTSIZE(2048)</strong></td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

   You can also refer to the following examples to code ANALYZEPART JCL statements and command parameters:

   - "ANALYZEPART example: Specifying the number of partitions" on page 34
   - "ANALYZEPART example: Specifying the partition size" on page 35
   - "ANALYZEPART example: Specifying the key ranges for partitions" on page 36
   - "ANALYZEPART example: Generating DBRC commands" on page 37
   - "ANALYZEPART example: Using unloaded data sets as input" on page 39
Simulating conversion or repartitioning using unloaded data sets as input

You can use unloaded data sets of a full-function database to simulate conversion from a full-function database to a HALDB, or unloaded data sets of a HALDB to simulate repartitioning of a HALDB.

About this task

Instead of supplying a full-function database or a HALDB as input for the ANALYZEPART function, you can supply unloaded data sets of a full-function database or of a HALDB and simulate the effect of database conversion or repartitioning.

Restriction: When you simulate conversion from a full-function database to a HALDB by supplying unloaded data sets as input, the ANALYZEPART function processes only the primary DBD that is specified on the DBD parameter. It does not process any secondary index databases.

Procedure

1. Prepare the unloaded data sets. If you already have unloaded data sets, make sure your unloaded data sets satisfy the following requirements:
   • The ANALYZEPART function reads all data in the unloaded data sets that you specify on the DFSUINPT DD statement. If your unloaded data sets contain all the data of a HALDB and you want the ANALYZEPART function to process only some partitions, you must split the unloaded data sets so that the unloaded data sets that you specify on the DFSUINPT DD statement contain only the data of partitions that you want to process. You can do so by using the split unload file utility. For more information, see “Example 4: Split unloaded data sets and simulate repartitioning of a HALDB” on page 40.
   • If the data in the unloaded data sets is masked and the original database uses unique keys, you must mask the unloaded data sets by using different unique values. The ANALYZEPART function does not check for duplicate keys. If duplicate keys exist in the unloaded data sets, the result of simulation might not be as expected.
   • If the database contains compressed segments (in other words, if the SEGMENT statements of the DBDGEN utility has a COMPRTN keyword), use the FABHURG1 utility of IBM IMS High Performance Unload for z/OS to create the unloaded data sets. You must specify the DECN control statement for the FABHURG1 utility to create unloaded data sets that the ANALYZEPART function can read. For more information, see the topic “FABHURG1 unload utility” in the IMS High Performance Unload User's Guide.

2. Locate the sample JCL members in the SIHCSAMP file, members IHCEMAIx.

3. Specify the JCL DD statements. For a list of DD statements, see “ANALYZEPART DD statements” on page 21.
   a. Specify a DFSUINPT DD statement and code the names of all the unloaded data sets on the statement.
   b. Specify other JCL DD statements.

4. Specify the ANALYZEPART command parameters. For a list of ANALYZEPART command parameters, see “ANALYZEPART command parameters” on page 24. For examples, see “ANALYZEPART example: Using unloaded data sets as input” on page 39.
a. Specify the INPUT(UNLOAD) parameter. This parameter requests the ANALYZE PART function to read data from unloaded data sets.

b. If the unloaded data sets contain data from a HALDB, specify the partitions to simulate with the PARTLIST parameter or the PARTITION parameter. The ANALYZE PART function does not check the consistency of partitions against the unloaded data sets specified on the DFSUINPT DD statement. If there is inconsistency between the partitions and data in the unloaded data sets, the result of simulation might not be as expected.

c. Specify other ANALYZE PART command parameters.

**Restriction:** When unloaded data sets are provided as input, the ANALYZE PART function ignores the following parameters:

- Parameters related to secondary index databases (INDPART, INDPRI M, INDSIZE)
- READINT parameter
ANALYZEPART DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the ANALYZEPART command.

Table 3. DD statements and record format for the ANALYZEPART command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>original_dbds</td>
<td>Input</td>
<td>Depends on database organization</td>
<td>Optional</td>
</tr>
<tr>
<td>DFSUINPT</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Conditionally required</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCLIST</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

//STEPLIB DD DISP=SHR,DSN=ihcload
// DD DISP=SHR,DSN=imstools
// DD DISP=SHR,DSN=reslib

where:

- *ihcload* is the name of the library that contains the IMS HALDB Toolkit load modules.
- *imstools* is the optional name of the library that contains the IMS reorganization utilities load modules.
- *reslib* is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.
IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.

**IMS**
This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

**IMSDALIB**
This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested.

Dynamic allocation of the database data sets is attempted in the following order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

**RECONx**
These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

**Attention:** If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

**original_dbds**
This statement is optional if the database to be processed is not a HALDB. This statement defines the database data set. One statement of this type must be present for each data set in the DBD that describes this database.

The DD name must match the DD name in the DBD.

For a HIDAM database, DD statements must also exist for the data sets that represent the index. The DD statements that relate to the index must contain DD names specified in the DBD for the index database.

The data sets, if specified, must reside on DASD. If you use dynamic allocation, do not code the DD statement for the database data sets.

**Attention:**
- You must not specify the DD statements for HALDB data sets because HALDB data sets are registered in RECON and are always allocated dynamically.
- If the DD statement is coded, do not code DCB=BUFNO=n or AMP='BUFND=n' to request access method buffers for database data sets because HSSR Engine of IMS HP Unload allocates its own buffer pools.

**DFSUINPT**
This statement is required when you supply unloaded data sets as input. The statement specifies one or more data sets that contain unloaded data of a full-function database or of a HALDB.
The supported format of the unloaded data sets is HD unload format. The ANALYZEPART function processes all the data sets that are specified on this DD statement and all the data inside the data sets.

For more information about supplying unloaded data sets as input, see the following topics:

- “Simulating conversion or repartitioning using unloaded data sets as input” on page 19
- “ANALYZEPART example: Using unloaded data sets as input” on page 39

IHCSYSIN
This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “ANALYZEPART command parameters” on page 24.

MSGPRINT
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

IHCLIST
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for reports. For more information about the reports generated in this data set, see “IHCLIST data set” on page 303.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE
This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

SYSUDUMP
This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
ANALYZEPART command parameters

The following parameters are available to use with ANALYZEPART command simulation tasks.

IMS HALDB Toolkit ignores any parameters that are unavailable for the requested process.

Subsections:

- “ANALYZEPART command parameters”
- “ANALYZEPART command parameters for generating DBRC commands” on page 28

ANALYZEPART command parameters

Use the following parameters to define the conditions for simulating the database.

Table 4. ANALYZEPART command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies which DBD is to be simulated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you specify a HALDB DBD, the value must be a HALDB master name and it cannot be a partition name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>INDPART</td>
<td>Optional</td>
<td>Specifies the number of index partitions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is effective only when you simulate conversion from a full-function database to a HALDB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cannot be used with INDSIZE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(1-999)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>INDPREM</td>
<td>Optional</td>
<td>Specifies whether a secondary index is partitioned the same as the primary database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is effective only when you simulate conversion from a full-function database to a HALDB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following restrictions apply:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The index key must be the root segment key.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The primary database must be partitioned using the KEYS parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> N</td>
</tr>
</tbody>
</table>
Table 4. ANALYZEPART command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDSIZE</td>
<td>Optional</td>
<td>Specifies the size of each index partition. This parameter is effective only when you simulate conversion from a full-function database to a HALDB. INDSIZE cannot be used with INDPART.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format (1 - 4095)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value 1024</td>
</tr>
<tr>
<td>INPUT</td>
<td>Optional</td>
<td>Specifies that the input is unloaded data sets. Code the DFSUINPT DD to specify the unloaded data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format (UNLOAD)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value None.</td>
</tr>
<tr>
<td>KEYLEN</td>
<td>Optional</td>
<td>Specifies the length of the subkey when IHCPSEL0 is used. For example, to partition by country, using a country code that is 3 characters in position 7-9 of your key, specify the following parameters and values: KEYLEN(3) KEYOFF(6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format (nnn) where nnn is 1 to root keylen - 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value None.</td>
</tr>
<tr>
<td>KEYOFF</td>
<td>Optional</td>
<td>Specifies the offset within the root segment key to use when IHCPSEL0 is specified as PARTSEL. The offset must be at least 1. Offset 0 does not need a partition selection exit. For example, to partition by country, using a country code that is 3 characters in position 7-9 of your key, specify the following parameters and values: KEYLEN(3) KEYOFF(6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format (nnn) where nnn is 1 to root keylen -1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value None.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>KEYS</td>
<td>Optional</td>
<td>Specifies a DD statement that contains records for the high keys. KEYS cannot be used with PARTNUM or PARTSIZE. The KEYS parameter can be specified as a character or hexadecimal. Specifying KEYS results in as many partitions as there are keys specified in the file. The specifications are analyzed to determine whether the requested partitioning is feasible. If the number of partitions has been specified, IMS HALDB Toolkit verifies that no partition is larger than 4 GB. If the high keys are created during analysis, there must be at least as many database records as there are requested partitions. <strong>Format</strong> <em>(ddname)</em> <strong>Default value</strong> If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the default setting PARTSIZE(2048) is used.</td>
</tr>
<tr>
<td>PARTITION</td>
<td>Optional</td>
<td>Specifies which partition is to be simulated. Either specify a partition name, or specify an asterisk (*) to simulate all the partitions. Either the PARTITION or the PARTLIST parameter must be provided when you simulate repartitioning of a HALDB. PARTITION and PARTLIST are mutually exclusive parameters. Do not specify this parameter when you simulate conversion from a full-function database to a HALDB. <strong>Format</strong> *(partition_name</td>
</tr>
<tr>
<td>PARTLIST</td>
<td>Optional</td>
<td>Specifies which partitions are to be simulated. You can specify multiple partition names. Either the PARTITION or the PARTLIST parameter must be provided when you simulate repartitioning of a HALDB. PARTITION and PARTLIST are mutually exclusive parameters. Do not specify this parameter when you simulate conversion from a full-function database to a HALDB. <strong>Format</strong> <em>(part1,part2,...,partn)</em> <strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Table 4. ANALYZEPART command parameters  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARTNUM</td>
<td>Optional</td>
<td>Specifies the number of partitions to create. PARTNUM cannot be used with PARTSIZE or KEYS. Specifying the PARTNUM parameter causes IMS HALDB Toolkit to find the high keys. The specifications are analyzed to determine whether the requested partitioning is feasible. PARTNUM might not result in the exact number of partitions specified because a root segment and all of its dependent segments must be written to the same partition. Therefore, fewer than PARTNUM partitions might be created so that hierarchical integrity is preserved. If the number of partitions has been specified, IMS HALDB Toolkit verifies that no partition is larger than 8 GB. If the high keys are created during analysis, there must be at least as many database records as there are requested partitions. <strong>Format</strong> (1 - 999) <strong>Default value</strong> If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the default setting PARTSIZE(2048) is used.</td>
</tr>
<tr>
<td>PARTSEL</td>
<td>Optional</td>
<td>Specifies a partition selection routine to use. The PARTSEL parameter must be specified with the KEYS parameter. IMS HALDB Toolkit provides an exit routine for partition selection (IHCPSEL0). When you use the IHCPSEL0 exit routine, the database is partitioned using a subset of the key. If you use the IHCPSEL0 exit routine, you must also select the KEYOFF and KEYLEN parameters. <strong>Format</strong> (partition_selection_routine) <strong>Default value</strong> None.</td>
</tr>
<tr>
<td>PARTSIZE</td>
<td>Optional</td>
<td>Specifies the size of partitions to create. PARTSIZE cannot be used with PARTNUM or KEYS. Specifying this parameter causes IMS HALDB Toolkit to find the high keys. The specifications are analyzed to determine whether the requested partitioning is feasible. <strong>Format</strong> (1 - 8191) <strong>Default value</strong> If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the default setting PARTSIZE(2048) is used.</td>
</tr>
</tbody>
</table>
Table 4. ANALYZEPART command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| READINT   | Optional    | Specifies that the ANALYZEPART function reads the database with read integrity (PROCOPT=G). This parameter is effective only when IMS HALDB Toolkit dynamically detects the existence of IMS High Performance Unload for z/OS. Specify READINT(N) to simulate HALDB partitions of a database that is used by an online IMS system and that allows update access (ACCESS=UP). When READINT(N) is used, the ANALYZEPART function reads the database without read integrity (PROCOPT=GO). **Attention:**

- When you specify READINT(N), IMS does not check the ownership of the segments returned. Therefore, the ANALYZEPART function might get a segment that had been updated by another program. Such a segment might cause data integrity problems, resulting in an IMS user abend, system loop, or system abend. To learn more about read without integrity, see the topic “Read without integrity” in IMS Application Programming.

- If the database is a PSINDEX that is used by an online IMS system with ACCESS=UP, READINT(N) cannot be used because the ANALYZEPART function reads the PSINDEX with read integrity (PROCOPT=G) even if you specify READINT(N). In this case, the job terminates with a database authorization failure.

Format (Y | N)

Default value

Y

ANALYZEPART command parameters for generating DBRC commands

Use the following additional parameters to control the Database Recovery Control (DBRC) commands that IMS HALDB Toolkit generates. If you do not need to generate DBRC commands, do not use these parameters. These parameters are effective only when the DBRCOUT parameter, which specifies to generate DBRC commands, is specified.

Table 5. ANALYZEPART command parameters for generating DBRC commands

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONST</td>
<td>Optional</td>
<td>Used as a substitute for the DBD name when building partition names. See also DBDPATT and FIRSTPART parameters.</td>
</tr>
</tbody>
</table>

Format (constant)

Default value None.
Table 5. ANALYZEPART command parameters for generating DBRC commands  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| DBDPATT   | Optional    | Creates a partition name from the DBD.  
**Format** *(cccccccc)* where *c* is an asterisk (*) or a period (.)  
You can use a maximum of 6 asterisks (*) to identify the position in the DBD name. The default is 6 asterisks.  
When 6 positions are specified, only 1 character is used for the partition name, creating a limit of 36 partitions.  
If fewer than 6 positions are specified, 1 - 3 characters are used for the partition name depending on the FIRSTPART parameter.  
You can also use periods (.) to specify that certain characters of the DBD name are ignored.  
**Default value** *(******..)* |
| DBRCOUT   | Required    | Specifies the name of the DD to which IMS HALDB Toolkit prints DBRC commands.  
Specify this parameter to generate DBRC commands. You must also specify the dbrcout DD statement in the JCL.  
**Format** *(ddname)*  
**Default value** None. |
| DSNDBD    | Optional    | Specifies the suffix portion of the new data set name that you are using.  
A total of 37 characters are allowed for the entire data set name, which includes the characters used for the DSNPREF prefix and the characters used for the DSNDBD suffix.  
**Format** *(DBD | PART | DBDPART)*  
  
  **DBD** Add the DBD name to DSNPREF.  
  **PART** Add the partition name to DSNPREF.  
  **DBDPART** Add the DBD name and the partition name to DSNPREF.  
**Default value** DBD |
Table 5. `ANALYZEPART` command parameters for generating DBRC commands  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNPREF</td>
<td>Conditionally required</td>
<td>Specifies a high-level data set name to use when building new partition data set names. The DBRC DSNPREFX parameter is created from this specification and is appended with the DBD name. The final data set name is then appended by DBRC with the DBD file letter and the partition name. When you simulate conversion from a non-HALDB to a HALDB, you must specify this parameter. When you simulate repartitioning of a HALDB, this parameter is optional. <strong>Format</strong> <em>(dsnprefx)</em>  <strong>Default value</strong> If omitted, the default is to copy from the existing DSNPREF.</td>
</tr>
<tr>
<td>FBFF</td>
<td>Optional</td>
<td>Specifies a new free block frequency factor (FBFF) to use when defining a partition. <strong>Format</strong> <em>(0 - 99)</em>  <strong>Default value</strong> If omitted, the default is to copy from the existing FBFF.</td>
</tr>
<tr>
<td>FIRSTP</td>
<td>Optional</td>
<td>Identifies the partition name extension to be added to the name &quot;stub&quot; created by the DBDPATT parameter. The &quot;stub&quot; and the extension cannot exceed 7 characters. <strong>Format</strong> <em>(ccc)</em>  <strong>Default value</strong> <em>v0</em>, if six asterisks (*) are specified on the DBDPATT parameter or the DBDPATT parameter is not specified.  <em>vAA</em>, if five asterisks are specified on the DBDPATT parameter.</td>
</tr>
</tbody>
</table>

| 30 | IMS High Availability Large Database Toolkit User’s Guide |
### Table 5. ANALYZEPART command parameters for generating DBRC commands (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| FSPF       | Optional    | Specifies a new free space percentage value for each interval or block (FSPF) to use when defining a partition.  
  **Format**: (0 - 99)  
  **Default value**: If omitted, the default action is to copy from the existing FSPF. |
| INITPART   | Optional    | DDNAME in JCL that allows parameters that are not currently specified by the product to be added to the DBRC INIT.PART command.  
  IMS HALDB Toolkit uses the defaults for GENMAX, DFLTJCL, ICJCL, OICJCL, RECOVJCL, RECOVPD, RECVJCL, and NOREUSE.  
  Any DBRC statements that you specify on the *initpart_dd* DD statement are included in the statements that are generated by the product. DBRC statements coded on the *initpart_dd* DD statement must conform to the syntax of the INIT.PART command. For the syntax rules, see the topic "INIT.PART command" in *IMS Commands*.  
  **Format**: (*initpart_dd*)  
  **Default value**: None. |
| KEYSORDR   | Optional    | This parameter is effective only for HALDBs. IMS HALDB Toolkit ignores this parameter if this parameter is specified for a non-HALDB.  
  When KEYS(*ddname*) is specified, IMS HALDB Toolkit reads the high-key values from the KEYS(*ddname*) and then assigns them to partitions in ascending partition number order by issuing CHANGE.PART commands to DBRC. When KEYSORDR is also specified, IMS HALDB Toolkit subsequently reassigns the high-keys to the partitions, in both their individual ascending EBCDIC order, by issuing CHANGE.PART commands again.  
  The CHANGE.PART commands that are issued when reassigning the high-keys are in ascending EBCDIC order of partition name, which is the same order as is presented by a DBRC LIST.DB command. KEYSORDR processing is done whether adding or deleting partitions, or changing partitions' attributes.  
  This option can also be used for the single purpose of reorganizing partitions' high-keys, which is accomplished by placing unchanged high-keys in the KEYS(*ddname*).  
  When the KEYSORDR option is specified, PARTITION(*) is also required.  
  **Format**: (Y | N)  
  **Default value**: N |
| RAAINCR    | Optional    | Increases HDAM root addressable area (RAA) by a specified number of blocks.  
  **Format**: (0 - 26777215)  
  **Default value**: None. |
Table 5. ANALYZEPART command parameters for generating DBRC commands  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| RAPS      | Optional    | Specifies new PHDAM root anchor points (RAPs).  

  **Format**  
  (1 - 255)  

  **Default value**  
  If omitted, the default is to copy from the existing RAPs.
ANALYZEPART examples

Use the following examples to learn what you can do with the ANALYZEPART function.

- “ANALYZEPART example: Specifying the number of partitions” on page 34
- “ANALYZEPART example: Specifying the partition size” on page 35
- “ANALYZEPART example: Specifying the key ranges for partitions” on page 36
- “ANALYZEPART example: Generating DBRC commands” on page 37
- “ANALYZEPART example: Using unloaded data sets as input” on page 39
ANALYZEPART example: Specifying the number of partitions

If you know how many partitions you want to split the database into, you can use the PARTNUM parameter to specify this number.

To specify the number of partitions, use the sample member IHCEMAI1. The following example simulates splitting a database into two partitions:

```
//ANAPART JOB CLASS=A,MSGCLASS=X
//ANALYZEPART DBD('xxx') - PARTNUM(2)
```

Figure 2. Sample JCL (IHCEMAI1) for batch using PARTNUM

If the database that you specify for the DBD parameter is a HALDB database, you must specify the PARTITION or the PARTLIST parameter. If the DBD parameter specifies a non-HALDB database, do not specify the PARTITION or the PARTLIST parameter.

For example, if you want to split a HALDB database into four partitions, change the PARTNUM value to 4 and specify the PARTITION parameter.

```
//IHCSYSIN DD *
ANALYZEPART DBD('xxx') - PARTNUM(4) - PARTITION(*)
```
**ANALYZEPART example: Specifying the partition size**

If you know how large you want each partition to be, you can use the PARTSIZE parameter to specify this number.

To specify the size of partitions, use the member IHCEMAI2. The following example simulates splitting a database into partitions each with a size of 1500 MB.

```
//ANAPART JOB CLASS=A,MSGCLASS=X
//*----------------------------------------------------------
//* Create partition boundary report
//* read database and report on partition size
//*----------------------------------------------------------
//S2 EXEC PGM=ICHCHALDB,DYNAMNBR=999,
// REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
// DD DISP=SHR,DSN=your.SDFSRESL
// DD DISP=SHR,DSN=your.SHPSLMD0
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//IMS DD DISP=SHR,DSN=your.DBDLIB
//IMSDALIB DD DISP=SHR,DSN=your.MDALIB
//TRACE DD DUMMY
//MSGPRINT DD SYSOUT=*
//IHCLIST DD SYSOUT=
//IHCSYSIN DD *
ANALYZEPART DBD(***)
   PARTSIZE(1500)
```

**Figure 3. Sample JCL (IHCEMAI2) for batch using PARTSIZE**

If the database that you specify for the DBD parameter is a HALDB database, you must specify the PARTITION or the PARTLIST parameter. If the DBD parameter specifies a non-HALDB database, do not specify the PARTITION or the PARTLIST parameter.

For example, if you want to repartition two partitions of a HALDB database into partitions each with a size of 2000 MB, change the PARTSIZE value to 2000 and specify the PARTLIST parameter.

```
//IHCSYSIN DD *
ANALYZEPART DBD(***)
   PARTSIZE(2000)
   PARTLIST(part1,part2)
```
ANALYZEPART example: Specifying the key ranges for partitions

If you want to assign specific key ranges to partitions, you can use the KEYS parameter to specify an input data set containing the keys. The KEYS parameter specifies the high key boundaries for each partition.

To specify the ranges for partitions, use the member IHCEMAI3. The following example simulates splitting a database into 13 partitions with each partition containing the keys up to the specified high key boundary.

```assembler
//ANAPART JOB CLASS=A,MSGCLASS=X
//******************************************************************************
//* Create partition boundary report for HALDB
//* read database and report on specific key boundaries
//******************************************************************************
//S2 EXEC PGM=IHCHALDB,DYNAMNBR=999, 
// REGION=80M 
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD 
// DD DISP=SHR,DSN=your.SDFSRESL 
// DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL 
// IMS DD DISP=SHR,DSN=your.DBDLIB 
// IMSDALIB DD DISP=SHR,DSN=your.MDALIB 
//TRACE DD DUMMY 
//MSGPRINT DD SYSOUT=* 
//IHCLIST DD SYSOUT=* 
//*********************************************************************** 
//*KEYSIN DD *(documentation only) 
//* your first high-order key for partition1 
//* your second high-order key for partition2 and so on... 
//* your highest key for last partition 
//*********************************************************************** 
//KEYSIN DD * 
A999999999 
B999999999 
C999999999 
D999999999 
E999999999 
F999999999 
G999999999 
H999999999 
I999999999 
J999999999 
K999999999 
L999999999 
M999999999 
//IHCSYSIN DD * 
ANALYZEPART DBD(*** PARTITION(*) - 
   KEYS(KEYSIN)
```

Figure 4. Sample JCL (IHCEMAI3) for batch using KEYS

If the database that you specify for the DBD parameter is a HALDB database, you must specify the PARTITION or the PARTLIST parameter. If the DBD parameter specifies a non-HALDB database, do not specify the PARTITION or the PARTLIST parameter.
ANALYZEPART example: Generating DBRC commands

When you simulate repartitioning of a HALDB or conversion to a HALDB by using the ANALYZEPART command, you can optionally specify the DBRCOUT parameter to generate a set of DBRC commands for the Database Recovery Control (DBRC) utility (DSPURX00).

If you use the CONVERT or the MAINTAIN command to make changes to a database, IMS HALDB Toolkit automatically updates information in RECON data sets so that the changes made to the database are reflected in the RECON data sets. However, if you plan to do this without using these commands, manual tasks typically involve unloading data from the database, updating the information in RECON data sets, and reloading data back to the database. If you use the ANALYZEPART command, IMS HALDB Toolkit will generate a set of DBRC commands that you can use to update the information in RECON data sets.

To generate DBRC commands, add the DBRCOUT parameter to the ANALYZEPART command and the dbrcout DD statement, as shown in the following example:

```
//dbrcout DD SYSOUT=*  
//IHCSYSIN DD *  
  ANALYZEPART DBD(xxxx) -  
    PARLIST(part1,part2) -  
    PARTNUM(3) -  
    DBRCOUT(dbrcout)
```

The DBRCOUT parameter value (dbrcout) specifies the name of the DD to which IMS HALDB Toolkit writes the DBRC commands.

Example 1

If you want the DBRC commands printed in EBCDIC order based on the high-keys specified in the KEYSIN DD, specify the KEYSORDR(Y) parameter, the PARTITION(*) parameter, and the KEYSIN DD statement, as shown in the following example:

```
//DBRCOUT DD SYSOUT=*  
//KEYSIN DD *  
  A999999999  
  B999999999  
  C999999999  
  D999999999  
//IHCSYSIN DD *  
  ANALYZEPART DBD(xxxx) -  
    PARTITION(*) -  
    KEYS(KEYSIN) -  
    KEYSORDR(Y) -  
    DBRCOUT(DBRCOUT)
```

Example 2

If you want the DBRC commands include specific DBRC keyword values, specify the parameter values. The following parameters are supported: CONST, DBDPATT, DSNDBD, DSNPREF, FBFF, FIRSTPART, FSPF, INITPART, RAAINC, RAPS. For more information about these parameters, see “ANALYZEPART command parameters for generating DBRC commands” on page 28.

The following example specifies specific DBRC keyword values for FBFF, FSPF, and RAPS keywords.
Example 3

In the following example:

- `//DBRCOUT DD SYSOUT=*` specifies to print the DBRC commands to the SYSOUT stream.
- The KEYSIN DD statement and the KEYSORDR(Y) parameter specify to print the DBRC commands in EBCDIC order of the keys.
- The DBDPAT, FIRSTPART, DSNPREF, and DSNDDBD parameters specify the DBRC command keyword values to be used in the DBRC commands.

```
//DBRCOUT DD SYSOUT=*  
//KEYSIN DD  
A999999999  
B999999999  
C999999999  
D999999999  
//IHCSYSIN DD  
ANALYZEPART  
DBD(TESTDB01) -  
PARTITION(*) -  
KEYS(KEYSIN) -  
KEYSORDR(Y) -  
DBDPATT(****....) -  
FIRSTPART(001) -  
DSNPREFX(DBSMS.DB.TEST001) -  
DSNDDBD(PART) -  
DBRCOUT(DBRCOUT)
```

The resulting DBRC commands will look like the following example.

```
CHANGE.PART DBD(TESTDB01) PART(TEST001) -  
KEYSTRNG(A999999999) -  
BLOCKSZE(B192) -  
FBFF(10) -  
FSPF(20) -  
DSNPREFIXX(DBSMS.DB.TEST001)  
CHANGE.PART DBD(TESTDB01) PART(TEST002) -  
BLOCKSZE(B192) -  
FBFF(10) -  
FSPF(20) -  
DSNPREFIXX(DBSMS.DB.TEST002)  
INIT.PART DBD(TESTDB01) PART(TEST003) -  
KEYSTRNG(B999999999) -  
BLOCKSZE(B192) -  
FBFF(10) -  
FSPF(20) -  
DSNPREFIXX(DBSMS.DB.TEST003)
```
ANALYZEPART example: Using unloaded data sets as input

If you want to use unloaded data sets as input for the ANALYZEPART function, specify the unloaded data sets on the DFSUINPT DD statement and specify the INPUT(UNLOAD) parameter.

Subsections:
- “Example 1: Simulate conversion to a HALDB”
- “Example 2: Simulate repartitioning of a HALDB (all partitions)”
- “Example 3: Simulate repartitioning of a HALDB (one partition)” on page 40
- “Example 4: Split unloaded data sets and simulate repartitioning of a HALDB” on page 40

Example 1: Simulate conversion to a HALDB

To simulate conversion of a full-function database to a HALDB, add the DFSUINPT DD statement and the INPUT(UNLOAD) parameter.

In this example:
- DFSUINPT DD statement specifies the unloaded data set.
- INPUT(UNLOAD) parameter specifies to read input data from the unloaded data set.
- PARTNUM(3) parameter specifies to convert the database into a HALDB with three partitions.

```
//DFSUINPT DD DISP=SHR, DSN=your.hd.unload.dataset
//IHCSYSIN DD *
  ANALYZEPART     DBD(xxxx) -
      INPUT(UNLOAD) -
      PARTNUM(3)
```

Example 2: Simulate repartitioning of a HALDB (all partitions)

To simulate repartitioning of all the partitions in a HALDB, use the following example.

In this example:
- DFSUINPT DD statement specifies the unloaded data set. The unloaded data set contains all the data from all the partitions.
- INPUT(UNLOAD) parameter specifies to read input data from the unloaded data set.
- PARTITION(*) specifies to process all the partitions in the HALDB.
- PARTSIZE(2048) specifies to create partitions each with a size of 2048 MB.
- DBRCOUT parameter and dbrcout DD specify to generate DBRC commands for repartitioning.

```
//DFSUINPT DD DISP=SHR, DSN=your.hd.allpart.unload.dataset
//dbrcout DD SYSOUT=* 
//IHCSYSIN DD *
  ANALYZEPART     DBD(xxxx) -
      INPUT(UNLOAD) -
      PARTITION(*) -
      PARTSIZE(2048) -
      DBRCOUT(dbrcout)
```
Example 3: Simulate repartitioning of a HALDB (one partition)

To simulate repartitioning of one partition into two partitions, use the following example.

In this example:
- DFSUINPT DD statement specifies the unloaded data set. The unloaded data set contains all the data of the partition.
- INPUT(UNLOAD) parameter specifies to read input data from the unloaded data set.
- PARTLIST(part1) specifies to process only one partition.
- PARTNUM(2) specifies to simulate repartitioning into two partitions.

```
//DFSUINPT DD DISP=SHR, DSN=your.hd.part1.unload.dataset
//IHCSYSIN DD *
ANALYZEPART DBD(xxxx) -
   INPUT(UNLOAD) -
   PARTLIST(part1) -
   PARTNUM(2) -
```

Example 4: Split unloaded data sets and simulate repartitioning of a HALDB

If you have an unloaded data set that contains data of all partitions and you want to simulate only some partitions, split the unloaded data set by using the split unload file utility before you simulate repartitioning with the ANALYZEPART function.

In this example, you have an unloaded data set that contains all the data from a HALDB that has three partitions (part1, part2, and part3). You want to simulate repartitioning of two partitions (part1 and part2).

Step 1: Splitting unloaded data set

Use the split unload file utility to split the unloaded data set into multiple unloaded data sets. This utility splits unloaded data sets into single partition unloaded data sets. For more information about the split unload file utility, see Chapter 8, “Splitting an unload file into single partition files,” on page 149.

The following JCL example is for the split unload file utility. In this example:
- DFSUINPT DD statement specifies the unloaded data set that contains all the data of a HALDB that has three partitions.
- part1Z, part2Z, and part3Z DD statements specify the output data sets, which are single partition unloaded data set for each partition.

```
//S2 EXEC PGM=IHCHALDB,
   REGION=80M
//STEPLIB DD DISP=SHR, DSN=your.SIHLOAD
// DD DISP=SHR, DSN=your.SOFSRESL
//RECON1 DD DISP=SHR, DSN=your.RECON1
//RECON2 DD DISP=SHR, DSN=your.RECON2
//RECON3 DD DISP=SHR, DSN=your.RECON3
```
Step 2: Simulate repartitioning of two partitions
Specify the unloaded data sets for part1 and part2 generated by the split unload file utility and run the ANALYZEPART function.

In this example:
- `your.hd.part1.unload.dataset` on the DFSUINPT DD contains unloaded data of partition part1.
- `your.hd.part2.unload.dataset` on the DFSUINPT DD contains unloaded data of partition part2.
- INPUT(UNLOAD) parameter specifies to read input data from the unloaded data sets.
- PARTLIST(part1,part2) specifies to process two partitions.
- PARTNUM(3) specifies to simulate repartitioning into three partitions.
Part 3. Converting to a HALDB

In batch mode, you can convert to HALDB in a single step by using the `CONVERT` command. Alternatively, you can use the ISPF user interface to perform the same task.

**Topic:**
- Chapter 3, “Converting to HALDB using CONVERT,” on page 45
Chapter 3. Converting to HALDB using CONVERT

Use the CONVERT command to convert a database in batch mode. New data set allocations use the same System Managed Storage (SMS) constructs or the same volume serial when running outside of SMS.

Topics:
- “CONVERT data flow” on page 46
- “Converting databases offline” on page 47
- “CONVERT DD statements” on page 49
- “CONVERT command parameters” on page 53
- “CONVERT examples” on page 68
The CONVERT function performs all of the following processes in a single job step:

- Unload
- Change DBD
- Define HALDB
- Allocate HALDB
- Initialize HALDB
- Reload
- Backup

**Figure 5. CONVERT data flow**
Converting databases offline

You can convert full-function databases offline by using the CONVERT command.

About this task

Databases must be offline to use this method. You must manually /DBR your databases before running the conversion job.

Restriction: Databases with external logical relationships are not supported.

Procedure

1. Locate the sample JCL for this task in the SIHCSAMP file, member IHCECVTN.
2. Specify the JCL DD statements. For a list of DD statements, see "CONVERT DD statements" on page 49.
3. Specify the CONVERT command parameters. For a list of CONVERT command parameters, see "CONVERT command parameters" on page 53.
   Specify the ONLINE parameter as ONLINE(N). ONLINE(N) is the default if you do not specify this parameter. The following parameters are required to perform offline conversion:

   //IHCSYSIN DD *
   CONVERT DBD(xxx) ONLINE(N)...

   You can also refer to the following examples to code CONVERT JCL statements and command parameters:
   • "CONVERT example: Specifying the number of partitions" on page 69
   • "CONVERT example: Specifying the partition size" on page 70
   • "CONVERT example: Specifying the key ranges for partitions" on page 71
4. Manually /DBR your databases before running the conversion job.

Example

The following example shows the sample JCL for batch offline conversion:
Your databases must be registered to DBRC
Your databases must be dbr'd/stopped prior to executing this job

EXEC PGM=IHCHALDB,DYNAMNBR=999,REGION=80M
STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
                === your hcma loadlib
STEPLIB DD DISP=SHR,DSN=your.SDFSRESL
                === ims reslib
STEPLIB DD DISP=SHR,DSN=your.SHPSLM0
                === ims tools loadlib
MACLIB DD DISP=SHR,DSN=your.SDFSRESL
                === ims reslib
MSCSALIB DD DISP=SHR,DSN=your.SDFSRESL
                === ims tools loadlib
MACLIB DD DISP=SHR,DSN=your.SDFMAC
                === ims maclib
MACLIB DD DISP=SHR,DSN=your.dbdlib
                === ims dbdlib
MACLIB DD DISP=SHR,DSN=your.mdalib
                === ims mdalib
SYSPRINT DD SYSOUT=*  
MSGPRINT DD SYSOUT=*  
DBDPRINT DD SYSOUT=*  
AMSPRINT DD SYSOUT=*  
SYSGUMP DD SYSOUT=*  
IHCLIST DD SYSOUT=*  
IHCSYSIN DD *
CONVERT ONLINE(N) -
  DBDPATT(*****...) DSNPREF(DBSMS.DB) -
  DBD(xxxx) -
  PARTNUM(2) -
  ICMID(3) -
  ICHLQ(DBSMS.DB.IC) -
  ICTRLR(2) -
  IC.COMP(Y) -
  IC.COMPRCN(FABJ_CMP2) -
  PTRCHECK(Y)

Figure 6. Sample JCL (IHCECVTN) for batch conversions (Online(N))
CONVERT DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the CONVERT command.

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>MACLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input/Output</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>PSBLIB</td>
<td>Input/Output</td>
<td>Same as IMS</td>
<td>Required when ACBGEN(Y)</td>
</tr>
<tr>
<td>IMSACB</td>
<td>Output</td>
<td>Same as IMS</td>
<td>Required when ACBGEN(Y)</td>
</tr>
<tr>
<td>original_dbds</td>
<td>Input</td>
<td>Depends on database organization</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCLIST</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>DBDPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Required</td>
</tr>
<tr>
<td>LNKPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Required</td>
</tr>
<tr>
<td>AMSPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Required</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR,DSN=ihcload
//       DD DISP=SHR,DSN=imstools
//       DD DISP=SHR,DSN=reslib
```

where:

- `ihcload` is the name of the library that contains the IMS HALDB Toolkit load modules.
• *imstools* is the optional name of the library that contains the IMS reorganization utilities load modules.

• *reslib* is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

**DFSRESLB**

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.

**MACLIB**

This statement is required. The statement specifies the MACLIB data set for DBDGEN.

**IMS**

This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

**IMSDALIB**

This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested.

Dynamic allocation of the database data sets is attempted in the following order:

1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

**RECONx**

These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

**Attention:** If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

**original_dbds**

This statement is optional. This statement defines the database data set. One statement of this type must be present for each data set in the DBD that describes this database. The DD name must match the DD name in the DBD.

For a HIDAM database, DD statements must also exist for the data sets that represent the index. The DD statements that relate to the index must contain DD names specified in the DBD for the index database.

The data sets, if specified, must reside on DASD. If you use dynamic allocation, do not code the DD statement for the database data sets.
Attention: If this DD statement is coded, do not code DCB=BUFNO=n or AMP='BUFND=n' to request access method buffers for database data sets because HSSR Engine of IMS HP Unload allocates its own buffer pools.

IHCSYSIN
This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “CONVERT command parameters” on page 53.

SYSPRINT
This statement is optional. The statement defines the statistics output data set and output from other utilities that are executed under the control of IMS HALDB Toolkit.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide it.

MSGPRINT
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

IHCLIST
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for reports. For more information about the reports generated in this data set, see “IHCLIST data set” on page 303.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

DBDPRINT
This statement is required. The statement defines the output data set for the assemble step of DBDGEN.

The data set can reside on DASD, or it can be routed through the output job.

LNKPRINT
This statement is required. The statement defines the output data set for the link step of DBDGEN.

The data set can reside on DASD, or it can be routed through the output job.

AMSPRINT
This statement is required. The statement defines the output data set for the IDCAMS utility.

The data set can reside on DASD, or it can be routed through the output job.

TRACE
This statement is optional. The statement defines the output data set that
IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

**SYSUDUMP**

This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.

**IMSACB**

This statement is required when ACBGEN(Y). The statement describes the staging ACBLIB where IMS HALDB Toolkit generates the new ACBs for the converted DBDs.

This data set must reside on DASD.

**PSBLIB**

This statement is required when ACBGEN(Y). The statement describes the PSB library to rebuild all PSBs in the ACB library that reference the DBD to be converted.
**CONVERT command parameters**

The following parameters are available to use with CONVERT command conversions.

IMS HALDB Toolkit ignores any parameters that are unavailable for the requested process.

*Table 7. CONVERT command parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACBGEN</td>
<td>Optional</td>
<td>Specifies to run the ACBGEN process during the takeover phase.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> N</td>
</tr>
<tr>
<td>BYTELIM</td>
<td>Optional</td>
<td>Specifies a new maximum-insert-byte for the DBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (1 - 16777215)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If omitted, the default action is to copy from the existing DBD.</td>
</tr>
<tr>
<td>CONST</td>
<td>Optional</td>
<td>Used as a substitute for the DBD name when building partition names.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See also DBDPATT, INDPATT, and FIRSTPART parameters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (constant)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>Optional</td>
<td>Specifies the name of the data class for the new SMS-managed databases.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>that are dynamically allocated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (dataclass)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If omitted, the default action is to copy from existing data sets.</td>
</tr>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies which full-function database is to be converted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (dbdbname)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>DBDBACKUP</td>
<td>Optional</td>
<td>Backs up the existing DBD to a specified library.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter includes a DDNAME that must be included in the JCL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (ddname)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>DBDCOPY</td>
<td>Optional</td>
<td>Copies the new DBD to your DBD library.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If NEWDBD is not specified, DBDCOPY is set to Y.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> Y</td>
</tr>
</tbody>
</table>

Chapter 3. Converting to HALDB using CONVERT  53
### Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBDPATT</td>
<td>Optional</td>
<td>Creates a partition name from the DBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ccc...)</em> where <em>c</em> is an asterisk (*) or a period (.).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can use a maximum of 6 asterisks (*) to identify the position in the DBD name. The default is 6 asterisks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When 6 are specified, only 1 character is used for the partition name, which creates a limit of 36 partitions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If fewer than 6 positions are specified, 1 - 3 characters are used for the partition name depending on the FIRSTPART parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can also use periods(.) to ignore certain characters of the DBD name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> <em>(*****..)</em></td>
</tr>
<tr>
<td>DBDSOURCE</td>
<td>Optional</td>
<td>Generates the DBD source for the converted HALDB and places it in the specified library.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter includes a DDNAME that must be included in the JCL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ddname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>DBRCBACK</td>
<td>Optional</td>
<td>Backs up the existing DBRC to a specified library.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter includes a DDNAME that must be included in the JCL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ddname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>DSG1</td>
<td>Optional</td>
<td>Changes a database with multiple data set groups into a database with a single data set group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> N</td>
</tr>
<tr>
<td>DSNDDBD</td>
<td>Optional</td>
<td>Specifies the suffix portion of the new data set name that you are using.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A total of 37 characters are allowed for the entire data set name, which includes the characters used for the DSNPREF prefix and the characters used for the DSNDDBD suffix.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(DBD</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> DBD</td>
</tr>
</tbody>
</table>

---

**IMS High Availability Large Database Toolkit User's Guide**
### Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| **DSNPREF** | Required | Specifies a high-level data set name to be used when building new partition data set names.  

The DBRC DSNPREFIX parameter is created from this specification and appended with the DBD name. The final data set name is then appended by DBRC with the DBD file letter and the partition number.  

**Format** *(dsnprefix)*  

**Default value**  
None. |
| **FBFF** | Optional | Specifies a new free block frequency factor (FBFF) to be used when defining a partition.  

**Format** *(0 - 99)*  

**Default value**  
If omitted, the default action is to copy from the existing FBFF. |
| **FIRSTPART** | Optional | Identifies the partition name extension to be added to the name "stub" created by the DBDPATT and INDPATT parameters.  

The "stub" and the extension cannot exceed 7 characters.  

**Format** *(ccc)*  

*ccc* defines the character strings that are used, where:  

- **0** Defines a string from 0-9,A-Z. The partition number is used as an index. It creates a 1-character extension.  

- **A** Defines a string from A-Z,0-9. The partition number is used as an index. It creates a 1-character extension.  

- **1** Defines a string from 1-9,A-Z. The partition number is used as an index. It creates a 1-character extension.  

- **01** Creates a 2-character numeric extension from 01-99. The partition number is used for the extension.  

- **AA** Creates a 2-character extension from AA to 99. For example, AA,...,AZ,A0,...,A9,BA,...,Z9,0A,...  

- **001** Creates a 3-character numeric extension from 001-999. The partition number is used for the extension.  

**Default value**  
- 0, if six asterisks (*) are specified on the DBDPATT parameter or the DBDPATT parameter is not specified.  

- AA, if five asterisks are specified on the DBDPATT parameter. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSPF</td>
<td>Optional</td>
<td>Specifies a new free space percentage value per interval or block (FSPF) to be used when defining a partition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td>IC.COMP</td>
<td>Optional</td>
<td>Used for IMS High Performance Image Copy processing. This parameter specifies whether compression is used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the copy is a standard image copy, this parameter is ignored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td>IC.COMPRTN</td>
<td>Optional</td>
<td>Used for IMS High Performance Image Copy processing. If IC.COMP(Y) is used, this parameter specifies the name of a compression routine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the copy is a standard image copy, this parameter is ignored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td>IC.VIC</td>
<td>Optional</td>
<td>Specifies whether a virtual image copy is taken.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify Y to activate IC.VIC and suppress the creation of image copies for all PSINDEXes. A NOTIFY.UIC is used instead.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td>IC.VICDSN</td>
<td>Optional</td>
<td>Activated only when IC.VIC(Y) is also specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies a string of up to 44 characters that are used as UDATA for the NOTIFY.UIC. The string must conform to z/OS and IMS data set name rules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
</tbody>
</table>
### Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1DSN</td>
<td>Optional</td>
<td>This parameter provides an alternative to using the ICHLQ, ICID, ICMPID, and ICTRLR parameters to specify the data set name for an image copy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dsname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The data set name can contain variables. To specify a variable, use an ampersand (&amp;). The following variables are supported:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;JOBNAME</strong> Name of this job.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;DBD</strong> The DBD name. This value changes to the first partition name when not all partitions are processed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;DDNAME</strong> The DD name of the database data set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;DATE1</strong> This variable is substituted as Dyyyyddd.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;DATE2</strong> This variable is substituted as Dddddyyy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;DATE3</strong> This variable is substituted as Dddmmyy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;DATE4</strong> This variable is substituted as Dmmddyy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;DATE5</strong> This variable is substituted as Dyyymmdd.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;TIME1</strong> This variable is substituted as Thhmmss.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>&amp;TIME2</strong> This variable is substituted as Thhmm.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• yyyy is the 4-digit year</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• yy is the last 2 digits of the year</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• mm is the month</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ddd is the day of the year</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dd is the day of the month</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• hh is the hour (24-hour clock) local time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• mm is the minute</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ss is the second</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>IC2DSN</td>
<td>Optional</td>
<td>Specifies the secondary image copy data set name. To create the data set name, use the same method that is described for the IC1DSN keyword.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must specify a unique data set name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dsname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| ICHLQ     | Optional    | Specifies the high-level qualifier for the image copy data sets.  
**Format** *(ichlq)*  
**Default value**  
If omitted, the action defaults to DSNPREF. |
| ICID      | Optional    | Used to override the default (IC1 or IC2) for the suffix qualifier for dual image copy data sets.  
**Format** *(1 | 2) or (Y | N)*  
Y is equivalent to 1, and N is equivalent to 2.  
**Default value** 1 |
| ICMID     | Optional    | Specifies the mid-level data set qualifier for image copy data sets.  
**Format** *(1 - 4)*  
The ICMID parameter specifications render the following mid-level data set qualifiers:  
ICMID(1)  
*dbname.ddname.*  
ICMID(2)  
*dbname.*  
ICMID(3)  
*ddname.*  
ICMID(4)  
None.  
**Default value** 3 |
| ICNUM     | Optional    | Specifies single or dual image copies.  
**Format** *(1 | 2)*  
**Default value** 1 |
| ICOFF     | Optional    | Specifies that no image copies are to be taken, and turns off the ICNEEDED parameter.  
**Format** *(Y | N)*  
**Default value** N |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICTRLR</td>
<td>Optional</td>
<td>Specifies whether to use another final suffix qualifier for image copy data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (1 - 6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The ICTRLR parameter specifications render the following final suffix qualifiers:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ICTRLR(1)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GDG</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ICTRLR(2)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ddymmdd.Tthmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ICTRLR(3)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dddmmdd.Tthmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ICTRLR(4)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ddymmdd.Tthmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ICTRLR(5)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ICTRLR(6)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ddymdd.Tthmmss</td>
</tr>
<tr>
<td>IMSID</td>
<td>Optional</td>
<td>Specifies the IMS ID to use when updating the IMS Library Integrity Utilities partition description.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (nnnn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If omitted, the default is from current, SDFSRESL (DFSVC000).</td>
</tr>
<tr>
<td>INDPART</td>
<td>Optional</td>
<td>Specifies the number of index partitions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cannot be used with INDSIZE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (1 - 999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>None.</td>
</tr>
<tr>
<td>INDPATT</td>
<td>Optional</td>
<td>Creates a partition name &quot;stub&quot; from the DBD name of a secondary index.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (ccccccccc) where c is an asterisk (*) or a period (.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is similar to DBDPATT. The &quot;*&quot; positions of the DBD name are kept to create the &quot;name stub.&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can use a maximum of 6 asterisks (*) to identify the position. The default is 6 asterisks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBDPATT is used. If neither DBDPATT nor INDPATT is specified, the default is (******.).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| INDPRIIM  | Optional    | Specifies whether a secondary index is partitioned the same as the primary database.  

The following restrictions apply:  
- The index key must be the root segment key.  
- The primary database must be partitioned using the KEYS parameter.  

Format (Y | N)  
Default value N  
| INDSIZE   | Optional    | Specifies the size of each index partition.  

Cannot be used with INDPART.  
Format (1 - 4095)  
Default value 1024  
| INITDB    | Optional    | DDNAME in JCL that allows parameters to be added to the DBRC INIT.DB command, which are not currently specified by the product.  

This parameter can be specified in stream or in a data set.  
If no parameters are added, the defaults for the Global Service Group (GSG) are used.  
If specified, this parameter requires a DDNAME which must be included in the JCL.  
Any DBRC statements that you specify on the initdb_dd DD statement are included in the statements that are generated by the product.  
DBRC statements coded on the initdb_dd DD statement must conform to the syntax of the INIT.DB command. For the syntax rules, see the topic "INIT.DB command" in IMS Commands.  
Format (initdb_dd)  
Default value None.  
| INITPART  | Optional    | DDNAME in JCL that allows parameters to be added to the DBRC INIT.PART command, which are not currently specified by the product.  

IMS HALDB Toolkit uses the defaults for GENMAX, DFLTJCL, ICJCL, OICJCL, RECOVJCL, RECOVPD, RECVJCL, and NOREUSE.  
Any DBRC statements that you specify on the initpart_dd DD statement are included in the statements that are generated by the product. DBRC statements coded on the initpart_dd DD statement must conform to the syntax of the INIT.PART command. For the syntax rules, see the topic "INIT.PART command" in IMS Commands.  
Format (initpart_dd)  
Default value None.  

60  
IMS High Availability Large Database Toolkit User’s Guide
Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYLEN</td>
<td>Optional</td>
<td>Specifies the length of the subkey when IHCPSEL0 is used. For example, to partition by country, using a country code that is 3 characters in position 7-9 of your key, specify the following: KEYLEN(3) KEYOFF(6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong>  (nnn) where nnn is 1 to root keylen -1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>KEYOFF</td>
<td>Optional</td>
<td>Specifies the offset within the root segment key to be used when IHCPSEL0 has been specified as PARTSEL. The offset must be at least 1. Offset 0 does not need a partition selection exit. For example, to partition by country, using a country code that is 3 characters in position 7-9 of your key, specify the following: KEYLEN(3) KEYOFF(6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong>  (nnn) where nnn is 1 to root keylen -1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>KEYS</td>
<td>Optional</td>
<td>Specifies a DD statement which contains records for the high keys; cannot be used with PARTNUM or PARTSIZE. This parameter can be specified as a character or hexadecimal. Specifying KEYS results in as many partitions as there are keys specified in the file. The specifications are analyzed to determine whether the requested partitioning is feasible. If the number of partitions has been specified, IMS HALDB Toolkit verifies that no partition is larger than 4 GB. If the high keys are created during analysis, there must be at least as many database records as there are requested partitions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong>  (ddname)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the default setting PARTSIZE(2048) is used.</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>Optional</td>
<td>Specifies a new SMS management class.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong>  (mgmtclas)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> If omitted, the default action is to copy from existing data sets.</td>
</tr>
</tbody>
</table>

Chapter 3. Converting to HALDB using CONVERT 61
Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| NEWDBD    | Optional    | Specifies the DDNAME of the data set that contains the changed HALDB DBDs. This parameter is a DDNAME, which must be included in the JCL. If NEWDBD is not specified, a temporary data set is used. DBDCOPY is then reset to Y.  
**Format** *(ddname)*  
**Default value** None. |
| ONLINE    | Optional    | Specifies whether the batch conversion is to be performed while the databases remain online.  
ONLINE(Y) is not supported by IMS Database Utility Solution.  
**Format** *(Y | N)*  
**Default value** N |
| OSAM      | Optional    | Converts VSAM to OSAM.  
**Format** *(Y | N)*  
**Default value** N |
| OVFLINCR  | Optional    | Increases HDAM overflow area by a number of blocks.  
**Format** *(0 - 9999999)*  
**Default value** None. |
| PARTNUM   | Optional    | Specifies the number of partitions to be created. PARTNUM cannot be used with PARTSIZE or KEYS. Specifying this parameter causes IMS HALDB Toolkit to find the high keys. The specifications are analyzed to determine whether the requested partitioning is feasible. PARTNUM might not result in the exact number of partitions specified because a root segment and all of its dependent segments must be written to the same partition. Therefore, fewer than PARTNUM partitions might be created so that hierarchical integrity is preserved.  
If the number of partitions has been specified, IMS HALDB Toolkit verifies that no partition is larger than 4 GB. If the high keys are created during analysis, there must be at least as many database records as there are requested partitions.  
**Format** *(1 - 999)*  
**Default value**  
If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the default setting PARTSIZE(2048) is used. |
### Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARTSEL</td>
<td>Optional</td>
<td>Specifies a partition selection routine to be used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter must be used with the KEYS parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMS HALDB Toolkit provides an exit for partition selection (IHCPS0L0). When</td>
</tr>
<tr>
<td></td>
<td></td>
<td>you use this exit, the database is partitioned using a subset of the key.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you use IHCPS0L0, you must also select the KEYOFF and KEYLEN parameters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> ([partition_selection_routine])</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>None.</td>
</tr>
<tr>
<td>PARTSIZE</td>
<td>Optional</td>
<td>Specifies the size of partitions that are created.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARTSIZE cannot be specified with PARTNUM or KEYS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifying the PARTSIZE parameter causes IMS HALDB Toolkit to find the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>high keys. The specifications are analyzed to determine whether the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>requested partitioning is feasible.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (1 - 8191)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• For a VSAM, 1 - 4095.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• For a VSAM with the OSAM(Y) parameter, 1 - 8191.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• For an OSAM, 1 - 8191.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>default setting PARTSIZE(2048) is used.</td>
</tr>
<tr>
<td>PTRCHECK</td>
<td>Optional</td>
<td>Performs pointer checking during image copy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is ignored when using the standard IMS image copy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>RAAINCR</td>
<td>Optional</td>
<td>Increases HDAM root addressable area (RAA) by a specified number of blocks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (0 - 26777215)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>None.</td>
</tr>
<tr>
<td>RAPS</td>
<td>Optional</td>
<td>Specifies new PHDAM root anchor points (RAPS).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (1 - 255)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If omitted, the default action is to copy from the existing RAPS.</td>
</tr>
</tbody>
</table>
Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| RELOAD.DBIOBUF       | Optional    | Used for IMS High Performance Load processing. Specifies the number of tracks for the access method (VSAM or BSAM) buffers that are used for one database data set group when the database is read or written to.  
If the load is a standard HD Load, this parameter is ignored.  
For details, see the *IMS High Performance Load User’s Guide*.  
**Format** (1 - 999)  
**Default value** 30 |
| RELOAD.DBRLBUF       | Optional    | Used for IMS High Performance Load processing. Specifies the number of database buffers (the number of tracks) to be used for one database data set group when the database is read or written to.  
If the load is a standard HD Load, this parameter is ignored.  
For details, see the *IMS High Performance Load User’s Guide*.  
**Format** (1 - 999)  
**Default value** 45 |
| RELOAD.FRSPC         | Optional    | Used for IMS High Performance Load processing. Specifies whether IMS HP Load secures free space according to the specification of the FRSPC parameter in DBD.  
If the load is a standard HD Load, this parameter is ignored.  
For details, see the *IMS High Performance Load User’s Guide*.  
**Format** (YES | NO)  
**Default value** YES |
| RELOAD.HPIO          | Optional    | Used for IMS High Performance Load processing. Specifies whether IMS HP Load uses the High Performance I/O Driver for output database data sets.  
If the load is a standard HD Load, this parameter is ignored.  
For details, see the *IMS High Performance Load User’s Guide*.  
**Format** (YES | NO)  
**Default value** YES |
### Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELOAD.OADSPR</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies whether IMS HP Load reduces the usage of data space when processing the overflow area by using the High Performance I/O driver or by using an OVERFLOW DD. This specification is valid only for HDAM or PHDAM databases. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide.</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> NO</td>
</tr>
<tr>
<td>RELOAD.RAAFORMAT</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies whether IMS HP Load is to format HDAM or PHDAM RAA blocks regardless of the number of blocks used. Once the overflow area is used, this parameter is ignored and the entire RAA is formatted. This specification is valid only for HDAM or PHDAM databases. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide.</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> YES</td>
</tr>
<tr>
<td>RELOAD.SEARCH</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies the limit value for a space search algorithm that finds a segment that does not fit in its home block. This specification is valid for HDAM or PHDAM databases. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide.</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(0 - 999)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> 0</td>
</tr>
<tr>
<td>RELOAD.SORT</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies whether to run the physical sequential sort processing before loading. This specification is valid only for HDAM and PHDAM databases. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide.</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(YES</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> NO</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RELOAD.USEREXIT</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies whether IMS HP Load invokes a user exit routine. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide</em>.</td>
</tr>
<tr>
<td>Format</td>
<td><em>(user_exit)</em></td>
<td>Default value None.</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>Optional</td>
<td>Specifies the name of the storage class for the new SMS-managed databases that are dynamically allocated. Format <em>(storclas)</em> Default value If omitted, the default action is to copy from existing data sets.</td>
</tr>
<tr>
<td>TAKEOVER.FILE</td>
<td>Optional</td>
<td>Saves the takeover statements (DBRC and IDCAMS) to a specified data set. Format <em>(ddname)</em> Default value None.</td>
</tr>
<tr>
<td>VOLALLO</td>
<td>Optional</td>
<td>Used to override the space allocation when creating the database data sets. Format <em>(nn, pppp, ssss)</em> Default value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <em>nn</em>: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <em>pppp</em>: 300</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <em>ssss</em>: 300</td>
</tr>
</tbody>
</table>
Table 7. CONVERT command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLALLOL</td>
<td>Optional</td>
<td>Used to override the space allocation when creating the ILDS data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> $(nn,pppp,ssss)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$nn$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If $nn$ is greater than 1, candidate volumes are defined. If $nn$ is zero, VOLALLOL is ignored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$pppp$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$ssss$</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• $nn$: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• $pppp$: None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• $ssss$: None.</td>
</tr>
<tr>
<td>VOLALLOX</td>
<td>Optional</td>
<td>Used to override the space allocation when creating the primary index data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> $(nn,pppp,ssss)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$nn$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If $nn$ is greater than 1, candidate volumes are defined. If $nn$ is zero, VOLALLOX is ignored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$pppp$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$ssss$</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• $nn$: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• $pppp$: None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• $ssss$: None.</td>
</tr>
</tbody>
</table>
CONVERT examples

Use the following examples to learn what you can do with the CONVERT function.

- “CONVERT example: Specifying the number of partitions” on page 69
- “CONVERT example: Specifying the partition size” on page 70
- “CONVERT example: Specifying the key ranges for partitions” on page 71
**CONVERT example: Specifying the number of partitions**

If you know how many partitions you intend to convert to, you can use the PARTNUM parameter to specify this number.

To specify the number of partitions, use the sample member IHCECVT1. The following example converts a database into two partitions:

```plaintext
//JOBNAME JOB NAME,000,CLASS=A,MSGCLASS=X
//******************************************************************************
//* SAMPLE JCL FOR CONVERT - using PARTNUM method
//* Your databases must be registered to DBRC
//* You must /dbar your databases prior to executing this job
//* unless you use ONLINE(Y) and have Online Reorg installed
//******************************************************************************
//S2 EXEC PGM=IHCHALDB,DYNAMNBR=999,REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIMLOAD                      === your hcma loadlib
// DD DISP=SHR,DSN=your.SDFSRESL    === ims reslib
// DD DISP=SHR,DSN=your.SHPSLMD0    === ims tools loadlib
//DFSRESLBD DD DISP=SHR,DSN=your.SDFSRESL    === ims reslib
//MACLIB DD DISP=SHR,DSN=your.SDFSRESL    === ims maclib
//IMS DD DISP=SHR,DSN=your.dbdlib    === ims dbdlib
//IMSDALIB DD DISP=SHR,DSN=your.mdalib     === ims mdalib
//SYSPRINT DD SYSPRINT**
//MSGPRINT DD SYSPRINT**
//DBDPRINT DD SYSPRINT**
//LNPBPRINT DD SYSPRINT**
//SYSDUMP DD SYSPRINT**
//IHCLIST DD SYSPRINT**
//IHCSYSIN DD *
//CONVERT ONLINE(N) -
// PARTNUM(2) -
// DBDPATT(*****...) DSNPREF(DBSMS.DB) -
// DBD(xxxx) -
// ICMID(3) -
// ICHLQ(DBSMS.DB.IC) -
// ICTRLR(2) -
// IC.COMP(Y) -
// IC.COMPRTN(FABJCMP2) -
// PTRCHECK(Y)
```

*Figure 7. Sample JCL (IHCECVT1) for batch using PARTNUM*
CONVERT example: Specifying the partition size

If you know how large you want each partition to be, you can use the PARTSIZE parameter to specify this number.

To specify the size of partitions, use the member IHCECVT2. The following example converts a database into partitions each with a size of 2048 MB.

```
//JOBNAM JOB_NAME,000,CLASS=A,MSGCLASS=X
//********************************************************************
//*                SAMPLE JCL FOR CONVERT - using PARTSIZE method  *          
//*  Your databases must be registered to DBRC                       
//*  You must /dbr your databases prior to executing this job        
//*  unless you use ONLINE(Y) and have Online Reorg installed        
//********************************************************************
//S2 EXEC PGM=IHCHALDB,DYNAMNBR=999,REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD <== your icma loadlib
//   DD DISP=SHR,DSN=your.SDFSRESL <== ims reslib
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL <== ims reslib
//MACLIB DD DISP=SHR,DSN=your.SDFSRESL <== ims reslib
//IMS DD DISP=SHR,DSN=your.dbdlib <== ims dbdlib
//IMSDALIB DD DISP=SHR,DSN=your.mdalib <== ims mdalib
//SYSPRINT DD SYSOUT **
//MSGPRINT DD SYSOUT **
//DBOPRINT DD SYSOUT **
//LNKPRINT DD SYSOUT **
//AMSPRINT DD SYSOUT **
//SYSDUMP DD SYSOUT **
//IHCLIST DD SYSOUT **
//IHCSYSIN DD *
   CONVERT ONLINE(N) -
      PARTSIZE(2048) -
      DBDPATT(****...) DSNPREF(DBSMS.DB) -
      DBD(xxxx) -
      ICMIID(3) -
      ICHLQ(DBSMS.DB.IC) -
      ICTRLR(2) -
      IC.COMP(Y) -
      IC.COMPRTN(FABJCMP2) -
      PTRCHECK(Y)
```

Figure 8. Sample JCL (IHCECVT2) for batch using PARTSIZE
CONVERT example: Specifying the key ranges for partitions

If you want to assign specific key ranges to partitions, you can use the KEYS parameter to specify an input data set containing the keys. The KEYS parameter specifies the high key boundaries for each partition.

To specify the ranges for partitions, use the member IHCECVT3. The following example converts a database into two partitions with each partition containing the keys up to the specified high key boundary.

```
//JOBNAME   JOB NAME,000,CLASS=A,MSGCLASS=X
//*******************************************************************************
//*                      SAMPLE JCL FOR CONVERT - using KEYS method                *
//*******************************************************************************
//* You must /dbr your databases prior to executing this job
//*******************************************************************************
//S2        EXEC PGM=IHCHALDB,DYNAMNBR=999,REGION=80M
//STEPLIB  DD DISP=SHR,DSN=your.SIHLOAD  <== your hcmaloadlib
//         DD DISP=SHR,DSN=your.SDFSRESL  <== ims reslib
//         DD DISP=SHR,DSN=your.SHPSLMDO <== ims tools loadlib
//DFSRESLB  DD DISP=SHR,DSN=your.SDFSRESL  <== ims reslib
//MACLIB   DD DISP=SHR,DSN=your.SDFSMAC  <== ims maclib
//IMS      DD DISP=SHR,DSN=your.mdalib  <== ims mdalib
//IMSDALIB DD DISP=SHR,DSN=your.dbdlib   <== ims dbdlib
//SYSPRINT DD SYSOUT=*                       
//MSGPRINT DD SYSOUT=*                       
//DBDPRINT DD SYSOUT=*                       
//LINKPRINT DD SYSOUT=*                      
//AMSPRINT DD SYSOUT=*                      
//SYSSDUMP DD SYSOUT=*                      
//IHCLIST  DD SYSOUT=*                      
//*******************************************************************************
//KEYSIN    DD *(documentation only)
//* your first high-order key for partition1
//* your second high-order key for partition2 and so on...
//* your highest key for last partition
//*******************************************************************************

//KEYSIN DD *
M9999999999
29999999999
//IHCSYSIN DD *
CONVERT ONLINE(N) - KEYS(KEYSIN) -
DBDPATT(******...) DSNPREF(DBSMS.DB) -
DBD(xxxx) -
ICMID(3) -
ICHLQ(DBSMS.DB.IC) -
ICTRLR(2) -
IC.COMP(Y) -
IC.COMPRTN(FABJCMP2) -
PTRCHECK(Y)
```

Figure 9. Sample JCL (IHCECVT3) for batch using KEYS
Part 4. Maintaining HALDBs

In batch mode, you can perform maintenance on HALDB partitions by using the **MAINTAIN** command. Use the **MAINTAIN** command to consolidate, split, and reorganize partitions.

Instead of using the **MAINTAIN** command in batch mode, you can alternatively use the ISPF user interface to perform the same tasks.

By using batch mode, you can:
- Merge two HALDBs into one, which provides the capability to resolve user partitioning
- Provide a way to delete single partitions without rebuilding secondary indexes

**Topic:**
- Chapter 4, “Maintaining HALDB using MAINTAIN,” on page 75
Chapter 4. Maintaining HALDB using MAINTAIN

Use the **MAINTAIN** command to perform maintenance tasks on a database in batch mode.

**Topics:**
- “MAINTAIN data flow” on page 76
- “Maintaining databases offline” on page 77
- “MAINTAIN DD statements” on page 79
- “MAINTAIN command parameters” on page 82
- “How the MAINTAIN function works when adding, deleting, and naming partitions” on page 97
- “MAINTAIN examples” on page 102
The following figure shows the data flow for the MAINTAIN function.

**Figure 10. MAINTAIN data flow**

The MAINTAIN function performs all of the following processes in a single job step:
- Unload
- Define new partition settings
- Allocate DBDS
- Initialize partitions
- Reload
- Backup
Maintaining databases offline

You can maintain HALDBs offline by using the `MAINTAIN` command and the batch process.

**About this task**

Alternatively, you can use the ISPF user interface to perform the same offline maintenance tasks.

**Procedure**

1. Take the databases offline manually.
   Databases must be offline before beginning any maintenance task.

2. Specify the JCL DD statements. For a list of DD statements, see "MAINTAIN DD statements" on page 79.

3. Specify the MAINTAIN command parameters. For a list of MAINTAIN command parameters, see "MAINTAIN command parameters" on page 82.

   You can perform several functions by using the `MAINTAIN` command. The following table lists some of the most common functions:

```
Table 8. Common MAINTAIN functions

<table>
<thead>
<tr>
<th>Scenario</th>
<th>MAINTAIN parameters</th>
<th>KEYSIN input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consolidate two partitions into one using PARTNUM</td>
<td>MAINTAIN DBD(xxxx) - PARTLIST(part1,part2) - PARTNUM(1)</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>
| Consolidate all partitions into one using one key (highest key) | MAINTAIN DBD(xxxx) - PARTITION(*) - KEYS(keysin) | //KEYSIN DD *
|                         |                     | highest-key  |
| Split one partition into two using PARTNUM | MAINTAIN DBD(xxxx) - PARTITION(part1) - PARTNUM(2) | Not applicable |
| Split one large partition into 2048 MB size partitions using PARTSIZE | MAINTAIN DBD(xxxx) - PARTITION(bigpart2) - PARTSIZE(2048) | Not applicable |
| Split one partition into three using your own key boundaries | MAINTAIN DBD(xxxx) - PARTITION(part1) - KEYS(keysin) | //KEYSIN DD *
|                         |                     | smaller-high-key middle-high-key last-high-key-part1 |
| Add, change, or delete partitions such that four partitions result, and that their high-keys are assigned in EBCDIC order based on the high-keys found in KEYSIN | MAINTAIN DBD(xxxx) PARTITION(*) - KEYS(keysin) KEYSORDR(Y) | //KEYSIN DD *
|                         |                     | A999 G999 M999 Z999 |
| Reorganize one partition without any changes | MAINTAIN DBD(xxxx) - PARTITION(part3) - PARTNUM(1) | Not applicable |
```

You can also refer to the following examples to code MAINTAIN JCL statements and command parameters:

- "MAINTAIN example: Specifying the number of partitions" on page 103
“MAINTAIN example: Specifying the partition size” on page 104
“MAINTAIN example: Specifying the key ranges for partitions” on page 105
**MAINTAIN DD statements**

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the **MAINTAIN** command.

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input/Output</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input/Output</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCLIST</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>AMSPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Required</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**STEPLIB**

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR,DSN=ihcload
// DD DISP=SHR,DSN=imstools
// DD DISP=SHR,DSN=reslib
```

where:

- *ihcload* is the name of the library that contains the IMS HALDB Toolkit load modules.
- *imstools* is the optional name of the library that contains the IMS reorganization utilities load modules.
- *reslib* is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

**DFSRESLB**

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.
IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.

Even if IMS High Performance Load modules are found in the combined load library of IMS solution pack in the STEPLIB concatenation, when the MAINTAIN function processes PSINDEXes, the function uses DFSURGL0 (HD Reorganization Reload utility) in this library to reload PSINDEXes.

**IMS**

This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

**IMSDALIB**

This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested. Dynamic allocation of the database data sets is attempted in the following order:

1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

**RECONx**

These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

**Attention:** If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

**IHCSYSIN**

This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “MAINTAIN command parameters” on page 82.

**SYSPRINT**

This statement is optional. The statement defines the statistics output data set and output from other utilities that are executed under the control of IMS HALDB Toolkit.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide it.

**MSGPRINT**

This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.
The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

**IHCLIST**

This statement is optional. The statement defines the IMS HALDB Toolkit output data set for reports. For more information about the reports generated in this data set, see “IHCLIST data set” on page 303.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

**AMSPRINT**

This statement is required. The statement defines the output data set for the IDCAMS utility.

The data set can reside on DASD, or it can be routed through the output job.

**TRACE**

This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

**SYSUDUMP**

This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
MAINTAIN command parameters

The following parameters are available to use with MAINTAIN command maintenance tasks.

IMS HALDB Toolkit ignores any parameters that are unavailable for the requested process.

Table 10. MAINTAIN command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BYTELIM</td>
<td>Optional</td>
<td>Specifies a new maximum-insert-byte for the DBD.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
<td>(1 - 16777215)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
<td>If omitted, the default action is to copy from the existing DBD.</td>
</tr>
<tr>
<td>CONST</td>
<td>Optional</td>
<td>Used as a substitute for the DBD name when building partition names. See also DBDPATT, INDPATT, and FIRSTPART parameters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
<td>(constant)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
<td>None.</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>Optional</td>
<td>Specifies the name of the data class for the new SMS-managed databases that are dynamically allocated.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
<td>(dataclass)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
<td>If omitted, the default action is to copy from existing data sets.</td>
</tr>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies which HALDB DBD is to be maintained.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The value must be a HALDB master name and it cannot be a partition name.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
<td>(dbdbname)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
<td>None.</td>
</tr>
<tr>
<td>DBDPATT</td>
<td>Optional</td>
<td>Creates a partition name from the DBD.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
<td>(cccccccc) where c is an asterisk (*) or a period (.).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can use a maximum of 6 asterisks (*) to identify the position in the DBD name. The default is 6 asterisks.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>When 6 positions are specified, only 1 character is used for the partition name, creating a limit of 36 partitions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If fewer than 6 positions are specified, 1 - 3 characters are used for the partition name depending on the FIRSTPART parameter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can also use periods (.) to specify that certain characters of the DBD name are ignored.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
<td>(*****..)</td>
</tr>
</tbody>
</table>
Table 10. MAINTAIN command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNDBD</td>
<td>Optional</td>
<td>Specifies the suffix portion of the new data set name that you are using. A total of 37 characters are allowed for the entire data set name, which includes the characters used for the DSNPREF prefix and the characters used for the DSNDBD suffix.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> ( \text{(DBD} \mid \text{PART} \mid \text{DBDPART)} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{DBD} Add the DBD name to DSNPREF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{PART} Add the partition name to DSNPREF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\textbf{DBDPART} Add the DBD name and the partition name to DSNPREF.</td>
</tr>
<tr>
<td>Default value</td>
<td></td>
<td>DBD</td>
</tr>
<tr>
<td>DSNPREF</td>
<td>Required</td>
<td>Specifies a high-level data set name to use when building new partition data set names. The DBRC DSNPREFIX parameter is created from this specification and is appended with the DBD name. The final data set name is then appended by DBRC with the DBD file letter and the partition name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> ( dsnprefix )</td>
</tr>
<tr>
<td>Default value</td>
<td></td>
<td>None.</td>
</tr>
<tr>
<td>FBFF</td>
<td>Optional</td>
<td>Specifies a new free block frequency factor (FBFF) to use when defining a partition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> ( 0 - 99 )</td>
</tr>
<tr>
<td>Default value</td>
<td></td>
<td>If omitted, the default is to copy from the existing FBFF.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| FIRSTPART | Optional    | Identifies the partition name extension to be added to the name "stub" created by the DBDPATT and INDPATT parameters. The "stub" and the extension cannot exceed 7 characters.  

**Format**  
(ccc)  

- **0**: Defines a string from 0-9,A-Z. The partition number is used as an index. It creates a 1-character extension.  
- **A**: Defines a string from A-Z,0-9. The partition number is used as an index. It creates a 1-character extension.  
- **1**: Defines a string from 1-9,A-Z. The partition number is used as an index. It creates a 1 character extension.  
- **01**: Creates a 2-character numeric extension from 01-99. The partition number is used for the extension.  
- **AA**: Creates a 2-character extension from AA to 99. For example, AA,...,AZ,A0,...,A9,BA,...,Z9,0A,...  
- **001**: Creates a 3-character numeric extension from 001-999. The partition number is used for the extension.  

**Default value**  
- 0, if six asterisks (*) are specified on the DBDPATT parameter or the DBDPATT parameter is not specified.  
- AA, if five asterisks are specified on the DBDPATT parameter. |
| FSPF      | Optional    | Specifies a new free space percentage value for each interval or block (FSPF) to use when defining a partition.  

**Format**  
(0 - 99)  

**Default value**  
If omitted, the default action is to copy from the existing FSPF. |
| IC.COMP   | Optional    | Used for IMS High Performance Image Copy processing. This parameter specifies whether compression is used. If the copy is a standard image copy, this parameter is ignored.  

**Format**  
(Y | N)  

**Default value**  
N |
Table 10. MAINTAIN command parameters  (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC.COMPRTN</td>
<td>Optional</td>
<td>Used for IMS High Performance Image Copy processing. If IC.COMP(Y) is used, this parameter specifies the name of a compression routine. If the copy is a standard image copy, this parameter is ignored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(comprtn)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong>  FABJCMP1</td>
</tr>
<tr>
<td>IC.VIC</td>
<td>Optional</td>
<td>Specifies whether a virtual image copy is taken. Specify Y to activate IC.VIC and suppress the creation of image copies for all PSINDEXes. A NOTIFY.UIC is used instead.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong>  N</td>
</tr>
<tr>
<td>IC.VICDSN</td>
<td>Optional</td>
<td>Activated only when IC.VIC(Y) is also specified. Specifies a string of up to 44 characters that is used as UDATA for the NOTIFY.UIC. The string must conform to z/OS and IMS data set name rules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dsname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong>  None.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IC1DSN</td>
<td>Optional</td>
<td>This parameter provides an alternative to using the ICHLQ, ICID, ICMID, and ICTRLR parameters to specify the data set name for an image copy.</td>
</tr>
</tbody>
</table>

**Format** *(dsname)*

The data set name can contain variables. To specify a variable, use an ampersand (&). The following variables are supported:

- **&JOBNAME**
  Name of this job.
- **&DBD**
  The DBD name. This value changes to the first partition name when not all partitions are processed.
- **&DDNAME**
  The DD name of the database data set.
- **&DATE1**
  This variable is substituted as Dyyyyddd.
- **&DATE2**
  This variable is substituted as Ddyyyggg.
- **&DATE3**
  This variable is substituted as Dddmmmyy.
- **&DATE4**
  This variable is substituted as Dmmddyy.
- **&DATE5**
  This variable is substituted as Dyyyyymm.
- **&TIME1**
  This variable is substituted as Thhmmss.

**Format** *(dsname)*

where:

- **yyyy** is the 4-digit year
- **yy** is the last 2 digits of the year
- **mm** is the month
- **ddd** is the day of the year
- **dd** is the day of the month
- **hh** is the hour (24-hour clock) local time
- **mm** is the minute
- **ss** is the second

**Default value**

None.

| IC2DSN    | Optional    | Specifies the secondary image copy data set name. To create the data set name, use the same method that is described for the IC1DSN keyword. |

You must specify a unique data set name.

**Format** *(dsname)*

**Default value**

None.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICHLQ</td>
<td>Optional</td>
<td>Specifies the high-level qualifier (HLQ) for the image copy data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ichlq)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> If omitted, defaults to DSNPREF.</td>
</tr>
<tr>
<td>ICID</td>
<td>Optional</td>
<td>Used to override the default (IC1 or IC2) for the suffix qualifier for dual image copy data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y is equivalent to 1, and N is equivalent to 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> 1</td>
</tr>
<tr>
<td>ICMID</td>
<td>Optional</td>
<td>Specifies the mid-level data set qualifier for image copy data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(1 - 4)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The ICMID parameter specifications render the following mid-level data set qualifiers:</td>
</tr>
</tbody>
</table>
|           |             | **ICMID(1)** *
|           |             | *dbname.ddname.ddname.*                                  |
|           |             | **ICMID(2)** *
|           |             | *dbname.*                                               |
|           |             | **ICMID(3)** *
|           |             | *ddname.*                                               |
|           |             | **ICMID(4)** *
<p>|           |             | None.                                                   |
|           |             | <strong>Default value</strong> 3                                     |
| ICNUM     | Optional    | Specifies single or dual image copies.                  |
|           |             | <strong>Format</strong> <em>(1 | 2)</em>                                    |
|           |             | <strong>Default value</strong> 1                                     |
| ICOFF     | Optional    | Specifies that no image copies are to be taken, and turns off the ICNEEDED parameter. |
|           |             | <strong>Format</strong> <em>(Y | N)</em>                                    |
|           |             | <strong>Default value</strong> N                                    |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICTRLR</td>
<td>Optional</td>
<td>Specifies whether to use another final suffix qualifier for image copy data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(1 - 6)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The ICTRLR parameter specifications render the following final suffix qualifiers:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICTRLR(1) GDG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICTRLR(2) Dyyddd.Thhmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICTRLR(3) Dmmddyy.Thhmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICTRLR(4) Dddmmyy.Thhmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICTRLR(5) None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ICTRLR(6) Dyyddd.Thhmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> 2</td>
</tr>
<tr>
<td>IMSID</td>
<td>Optional</td>
<td>Specifies the IMS ID to use when updating the IMS Library Integrity Utilities partition description.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(nnnn)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> If omitted, the default is from current, SDFSRESL (DFSVC000).</td>
</tr>
<tr>
<td>INITPART</td>
<td>Optional</td>
<td>DDNAME in JCL that allows parameters that are not currently specified by the product to be added to the DBRC INIT.PART command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IMS HALDB Toolkit uses the defaults for GENMAX, DFLTJCL, ICJCL, OICJCL, RECOVJCL, RECOVPD, RECVJCL, and NOREUSE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Any DBRC statements that you specify on the initpart_dd DD statement are included in the statements that are generated by the product. DBRC statements coded on the initpart_dd DD statement must conform to the syntax of the INIT.PART command. For the syntax rules, see the topic &quot;INIT.PART command&quot; in IMS Commands.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(initpart_dd)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>KEYLEN</td>
<td>Optional</td>
<td>Specifies the length of the subkey when IHCPSEL0 is used. For example, to partition by country, using a country code that is 3 characters in position 7-9 of your key, specify the following parameters and values: KEYLEN(3) KEYOFF(6) <strong>Format</strong> (nnn) where nnn is 1 to root keylen -1 <strong>Default value</strong> None.</td>
</tr>
<tr>
<td>KEYOFF</td>
<td>Optional</td>
<td>Specifies the offset within the root segment key to use when IHCPSEL0 is specified as PARTSEL. The offset must be at least 1. Offset 0 does not need a partition selection exit. For example, to partition by country, using a country code that is 3 characters in position 7-9 of your key, specify the following parameters and values: KEYLEN(3) KEYOFF(6) <strong>Format</strong> (nnn) where nnn is 1 to root keylen -1 <strong>Default value</strong> None.</td>
</tr>
<tr>
<td>KEYS</td>
<td>Optional</td>
<td>Specifies a DD statement that contains records for the high keys. KEYS cannot be used with PARTNUM or PARTSIZE. The KEYS parameter can be specified as a character or hexadecimal. Specifying KEYS results in as many partitions as there are keys specified in the file. The specifications are analyzed to determine whether the requested partitioning is feasible. If the number of partitions has been specified, IMS HALDB Toolkit verifies that no partition is larger than 4 GB. If the high keys are created during analysis, there must be at least as many database records as there are requested partitions. If KEYSORDR(Y) is specified with KEYS, then the keys are sorted in EBCDIC order, then assigned to the partitions in name order (also EBCDIC order). <strong>Format</strong> (ddname) <strong>Default value</strong> If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the default setting PARTSIZE(2048) is used.</td>
</tr>
</tbody>
</table>
Table 10. MAINTAIN command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| KEYSORDR     | Optional    | When KEYS(ddname) is specified, IMS HALDB Toolkit reads the high-key values from the KEYS(ddname) and then assigns them to partitions in ascending partition number order by issuing CHANGE.PART commands to DBRC. When KEYSORDR is also specified, IMS HALDB Toolkit subsequently reassigns the high-keys to the partitions, in both their individual ascending EBCDIC order, by issuing CHANGE.PART commands again. The CHANGE.PART commands that are issued when reassigning the high-keys are in ascending EBCDIC order of partition name, which is the same order as is presented by a DBRC LIST.DB command. KEYSORDR processing is done whether adding or deleting partitions, or changing partitions' attributes. This option can also be used for the single purpose of reorganizing partitions' high-keys, which is accomplished by placing unchanged high-keys in the KEYS(ddname).

When the KEYSORDR option is specified, PARTITION(*) is also required.

**Format**  \( (Y \mid N) \)

**Default value**  \( N \) |
| MGMTCLAS     | Optional    | Specifies a new SMS management class.

**Format**  \( (mgmtclas) \)

**Default value**  If omitted, the default action is to copy from existing data sets.

| NOOSAM8G     | Optional    | Specifies to change the maximum capacity of OSAM data sets from 8 GB of data to 4 GB of data.

NOOSAM8G(Y) is applicable only for OSAM data sets of PHDAM or PHIDAM database. The database must be registered as OSAM8G in the RECON data sets.

If you specify NOOSAM8G(Y), you must also specify PARTITION(*).

**Format**  \( (Y \mid N) \)

**Default value**  \( N \) |
| ONLINE       | Optional    | Specifies whether the batch maintenance is to be performed while the databases remain online.

ONLINE(Y) is not supported by IMS Database Utility Solution.

**Format**  \( (Y \mid N) \)

**Default value**  \( N \) |
| OVFLINCR     | Optional    | Increases HDAM overflow area by a number of blocks.

**Format**  \( (0 - 9999999) \)

**Default value**  None.
Table 10. MAINTAIN command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| PARTITION | Optional    | Specifies which partition is to be maintained. Either specify a partition name, or specify an asterisk (*) to process all the partitions. PARTLIST and PARTITION are mutually exclusive. **Format** *(partition_name | *)*  
**Default value** None. |
| PARTLIST  | Optional    | Specifies which partitions are to be maintained. You can specify multiple partition names. PARTLIST and PARTITION are mutually exclusive. **Format** *(part1,part2,...,partn)*  
**Default value** None. |
| PARTNUM   | Optional    | Specifies the number of partitions to create.  
PARTNUM cannot be used with PARTSIZE or KEYS. Specifying the PARTNUM parameter causes IMS HALDB Toolkit to find the high keys. The specifications are analyzed to determine whether the requested partitioning is feasible. PARTNUM might not result in the exact number of partitions specified because a root segment and all of its dependent segments must be written to the same partition. Therefore, fewer than PARTNUM partitions might be created so that hierarchical integrity is preserved.  
If the number of partitions has been specified, IMS HALDB Toolkit verifies the size of partitions as follows:  
• For a VSAM or OSAM HALDB whose maximum data capacity is 4 GB, no partition is larger than 4 GB.  
• For an OSAM HALDB whose maximum data capacity is 8 GB, no partition is larger than 8 GB.  
If the high keys are created during analysis, there must be at least as many database records as there are requested partitions.  
**Format** *(1 - 999)*  
**Default value** If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the default setting PARTSIZE(2048) is used. |
| PARTSEL   | Optional    | Specifies a partition selection routine to use.  
The PARTSEL parameter must be specified with the KEYS parameter. IMS HALDB Toolkit provides an exit routine for partition selection (IHCPSELO). When you use the IHCPSELO exit routine, the database is partitioned using a subset of the key. If you use the IHCPSELO exit routine, you must also select the KEYOFF and KEYLEN parameters.  
**Format** *(partition_selection_routine)*  
**Default value** None. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARTSIZE</td>
<td>Optional</td>
<td>Specifies the size of partitions to create.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARTSIZE cannot be used with PARTNUM or KEYS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifying this parameter causes IMS HALDB Toolkit to find the high keys. The specifications are analyzed to determine whether the requested partitioning is feasible.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (1 - 8191)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• For a VSAM or OSAM HALDB whose maximum data capacity is 4 GB, 1 - 4095.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• For an OSAM HALDB whose maximum data capacity is 8 GB, 1 - 8191.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If none of the parameters PARTNUM, PARTSIZE, or KEYS are specified, the default setting PARTSIZE(2048) is used.</td>
</tr>
<tr>
<td>PTRCHECK</td>
<td>Optional</td>
<td>Performs pointer checking during image copy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter is ignored when the standard IMS image copy is used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>RAAINCR</td>
<td>Optional</td>
<td>Increases HDAM root addressable area (RAA) by a specified number of blocks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (0 - 26777215)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>None.</td>
</tr>
<tr>
<td>RAPS</td>
<td>Optional</td>
<td>Specifies new PHDAM root anchor points (RAPs).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (1 - 255)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If omitted, the default is to copy from the existing RAPs.</td>
</tr>
<tr>
<td>RELOAD,DBIOBUF</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies the number of tracks for the access method (VSAM or BSAM) buffers that are used for one database data set group when the database is read or written to.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the load is a standard HD Load, this parameter is ignored.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For details, see the <em>IMS High Performance Load User’s Guide</em>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (1 - 999)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>
Table 10. MAINTAIN command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELOAD.DBRLBUF</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies the number of database buffers (the number of tracks) to be used for one database data set group when the database is read or written to. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide</em>. <strong>Format</strong> (1 - 999) <strong>Default value</strong> 45</td>
</tr>
<tr>
<td>RELOAD.FRSPC</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies whether IMS HP Load secures free space according to the specification of the FRSPC parameter in DBD. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide</em>. <strong>Format</strong> (YES</td>
</tr>
<tr>
<td>RELOAD.HPIO</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies whether IMS HP Load uses the High Performance I/O Driver for output database data sets. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide</em>. <strong>Format</strong> (YES</td>
</tr>
<tr>
<td>RELOAD.OADSPR</td>
<td>Optional</td>
<td>Used for IMS High Performance Load processing. Specifies whether IMS HP Load reduces the usage of data space when processing the overflow area by using the High Performance I/O driver or by using an OVERFLOW DD. This specification is valid only for HDAM or PHDAM databases. If the load is a standard HD Load, this parameter is ignored. For details, see the <em>IMS High Performance Load User’s Guide</em>. <strong>Format</strong> (YES</td>
</tr>
</tbody>
</table>
### Table 10. MAINTAIN command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| RELOAD.RAAFORMAT   | Optional    | Used for IMS High Performance Load processing. Specifies whether IMS HP Load is to format HDAM or PHDAM RAA blocks regardless of the number of blocks used. Once the overflow area is used, this parameter is ignored and the entire RAA is formatted. This specification is valid only for HDAM or PHDAM databases.  
  If the load is a standard HD Load, this parameter is ignored.  
  For details, see the *IMS High Performance Load User’s Guide*.  
  **Format** *(YES | NO)*  
  **Default value**  
  YES                                                                   |
| RELOAD.SEARCH      | Optional    | Used for IMS High Performance Load processing. Specifies the limit value for a space search algorithm that finds a segment that does not fit in its home block. This specification is valid for HDAM or PHDAM databases.  
  If the load is a standard HD Load, this parameter is ignored.  
  For details, see the *IMS High Performance Load User’s Guide*.  
  **Format** *(0 - 999)*  
  **Default value**  
  0                                                                     |
| RELOAD.SORT        | Optional    | Used for IMS High Performance Load processing. Specifies whether to run the physical sequential sort processing before loading. This specification is valid only for HDAM and PHDAM databases.  
  If the load is a standard HD Load, this parameter is ignored.  
  For details, see the *IMS High Performance Load User’s Guide*.  
  **Format** *(YES | NO)*  
  **Default value**  
  NO                                                                    |
| RELOAD.USEREXIT    | Optional    | Used for IMS High Performance Load processing. Specifies whether IMS HP Load invokes a user exit routine.  
  If the load is a standard HD Load, this parameter is ignored.  
  For details, see the *IMS High Performance Load User’s Guide*.  
  **Format** *(user_exit)*  
  **Default value**  
  None.                                                                 |
| STORCLAS           | Optional    | Specifies the name of the storage class for the new SMS-managed databases that are dynamically allocated.  
  **Format** *(storclas)*  
  **Default value**  
  If omitted, the default action is to copy the SMS storage class from existing data sets. |
Table 10. MAINTAIN command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAKEOVER.FILE</td>
<td>Optional</td>
<td>Saves the takeover statements (DBRC and IDCAMS) to a specified data set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ddname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>VOLALLO</td>
<td>Optional</td>
<td>Used to override the space allocation when creating the database data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(nn,pppp,ssss)</em></td>
</tr>
</tbody>
</table>
|               |             | *nn* The number of volumes to be used. This parameter is for SMS controlled allocations. 
|               |             | *pppp* The number of cylinders for the primary allocation. The maximum is 2000. |
|               |             | *ssss* The number of cylinders for the secondary allocation. The maximum is 2000. |
|               |             | **Default value**                                                          |
|               |             | *nn*: 0                                                                    |
|               |             | *pppp*: 300                                                                |
|               |             | *ssss*: 300                                                                |
| VOLALLOL      | Optional    | Used to override the space allocation when creating the ILDS data sets.    |
|               |             | **Format** *(nn,pppp,ssss)*                                                |
|               |             | *nn* The number of volumes to be used. This parameter is for SMS controlled allocations. 
|               |             | *pppp* The number of cylinders for the primary allocation.                 |
|               |             | *ssss* The number of cylinders for the secondary allocation.               |
|               |             | **Default value**                                                          |
|               |             | *nn*: 0                                                                    |
|               |             | *pppp*: None.                                                              |
|               |             | *ssss*: None.                                                             |
### Table 10. MAINTAIN command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLALLOX</td>
<td>Optional</td>
<td>Used to override the space allocation when creating the primary index data sets.</td>
</tr>
</tbody>
</table>

**Format** \((nn,pppp,ssss)\)

- **nn**: The number of volumes to be used. This parameter is for SMS controlled allocations. If nn is greater than 1, candidate volumes are defined. If nn is zero, VOLALLOX is ignored.
- **pppp**: The number of cylinders for the primary allocation.
- **ssss**: The number of cylinders for the secondary allocation.

**Default value**

- \(nn\): 0
- \(pppp\): None.
- \(ssss\): None.
How the MAINTAIN function works when adding, deleting, and naming partitions

Learn how the MAINTAIN function adds, deletes, and names partitions.

Adding or deleting partitions

When adding partitions (splitting partitions) or deleting partitions (merging partitions), the MAINTAIN function processes the partitions according to the following rules:

- If the high key of an existing partition matches one of the new high keys, IMS HALDB Toolkit keeps that existing partition unchanged.
- IMS HALDB Toolkit assigns new high keys to partitions in ascending order, firstly to the new partitions (if created) and then to the existing partitions that have no high keys assigned. When there are multiple existing partitions with no high keys assigned, IMS HALDB Toolkit assigns high keys to the partitions in ascending order of the original high keys.
- IMS HALDB Toolkit deletes unnecessary partitions in ascending order of the original high keys.

The following examples demonstrate specific cases to help you understand the changes that the MAINTAIN function makes to partitions.

**Example 1: Splitting one partition**

HALDB PHDOA has three partitions; Part_A (high key is 200), Part_B (high key is 400), and Part_C (high key is 600). You want to split partition Part_B into two partitions so you specify the MAINTAIN command as follows:

```
MAINTAIN DBD(PHDOA) PARTITION(Part_B) PARTNUM(2)
```

In this case, IMS HALDB Toolkit processes the partitions as follows:

1. Creates a new partition, Part_D.
2. Calculates the new high keys for splitting partition Part_B and resulted with high keys of 300 and 400.
3. Checks whether the existing partition (in this case Part_B) has one of the new high keys. IMS HALDB Toolkit detected that Part_B has the high key of 400 so it keeps Part_B unchanged.
4. Assigns the new high key of 300 to the new partition Part_D.
Example 2: Splitting multiple partitions

HALDB PHDOB has four partitions; Part_A (high key is 150), Part_B (high key is 300), Part_C (high key is 450), and Part_D (high key is 600). You want the HALDB PHDOB to have six partitions instead of four partitions so you specify the MAINTAIN command as follows:

```
MAINTAIN DBD(PHDOB) PARTITION(*) PARTNUM(6)
```

In this case, IMS HALDB Toolkit processes the partitions as follows:

1. Creates two new partitions, Part_E and Part_F.
2. Calculates the new high keys for repartitioning and resulted with high keys of 100, 200, 300, 400, 500, and 600.
3. Checks whether any of the existing partitions has one of the new high keys. It detected that Part_B has the high key of 300 and Part_D has the high key of 600 so it keeps these two partitions unchanged.
4. Assigns a high key to each of the remaining partitions in ascending order of the high keys, starting with the new partition; high key 100 to Part_E and high key 200 to Part_F. Then it assigns the remaining high keys to the existing partitions that have no high keys assigned; high key 400 to Part_A and high key 500 to Part_C.
Example 3: Merging multiple partitions into one partition

HALDB PHDOC has three partitions; Part_A (high key is 200), Part_B (high key is 400), and Part_C (high key is 600). You want the HALDB PHDOC to have only one partition instead of three partitions so you specify the MAINTAIN command as follows:

```
MAINTAIN DBD(PHDOC) PARTITION(*) PARTNUM(1)
```

In this case, IMS HALDB Toolkit processes the partitions as follows:

1. Checks whether any of the existing partitions has the new high key, which is 600. It detected that Part_C has the high key of 600 so it keeps Part_C unchanged.
2. Deletes the other two partitions.
Example 4: Merging multiple partitions into fewer partitions

HALDB PHDOD has six partitions; Part_A (high key is 100), Part_B (high key is 200), Part_C (high key is 300), Part_D (high key is 400), Part_E (high key is 500), and Part_F (high key is 600). You want the HALDB PHDOD to have four partitions instead of six partitions so you specify the MAINTAIN command as follows:

```
MAINTAIN DBD(PHDOD) PARTITION(*) PARTNUM(4)
```

In this case, IMS HALDB Toolkit processes the partitions as follows:

1. Calculates the new high keys for the partitions and resulted with high keys of 150, 300, 450, and 600.
2. Checks whether any of the existing partitions has one of the new high keys. It detected that Part_C has the high key of 300 and Part_F has the high key of 600 so it keeps these two partitions unchanged.
3. Deletes two partitions in ascending order of original high keys. In this case, Part_A, which had a high key of 100, and Part_B, which had a high key of 200.
4. Assigns the remaining high keys to each of the remaining partitions in ascending order of the high keys; high key 150 to Part_D and high key 450 to Part_E.

![Original partitions](image1)

![New partitions](image2)

Assigning partition names

The MAINTAIN function does not change the names of existing partitions. It only assigns names to new partitions.

New partition names are determined based on the CONST, DBDPATT, and FIRSTPART parameters and the current partition ID in the RECON data sets. For more information about these parameters, see "MAINTAIN command parameters" on page 82.

The following examples demonstrate how IMS HALDB Toolkit assigns partition names when the FIRSTPART parameter is specified.
Example 1: Assigning partition names when FIRSTPART(01)

HALDB PHDOE has five partitions; PHDOE01, PHDOE02, PHDOE03, PHDOE04, and PHDOE05. The value of CURRENT PARTITION ID in RECON is 5 (index=5). You want to split partition PHDOE01 into three partitions so you specify the MAINTAIN command as follows:

```
MAINTAIN DBD(PHDOE) DBDPATT(*****...) PARTITION(PHDOE01) PARTNUM(3) FIRSTPART(01)
```

In this case, IMS HALDB Toolkit processes the partitions as follows:
1. Creates two new partitions.
2. Reads the DBDPAT(*****...) parameter and uses the first five letters (PHDOE) of the DBD name for the first five characters of the new partition names.
3. Reads the FIRSTPART(01) parameter and obtains the value of CURRENT PARTITION ID from RECON, which is 5. Adds two character numeric extension of 06 and 07 to the end of the new partition names.
4. The two new partitions have the names of PHDOE06 and PHDOE07 respectively.

Example 2: Assigning partition names when FIRSTPART(AA)

HALDB PHDOF has five partitions; PHDOFAA, PHDOFAB, PHDOFAC, PHDOFAD, and PHDOFAE. The value of CURRENT PARTITION ID in RECON is 5 (index=5). You want to split partition PHDOFAA into three partitions so you specify the MAINTAIN command as follows:

```
MAINTAIN DBD(PHDOF) DBDPATT(*****...) PARTITION(PHDOFAA) PARTNUM(3) FIRSTPART(AA)
```

In this case, IMS HALDB Toolkit processes the partitions as follows:
1. Creates two new partitions.
2. Reads the DBDPAT(*****...) parameter and uses the five letters (PHDOF) of the DBD name for the first five characters of the new partition names.
3. Reads the FIRSTPART(AA) parameter and obtains the value of CURRENT PARTITION ID from RECON, which is 5. Adds two character extension of AF and AG to the end of the new partition names.
4. The two new partitions have the names of PHDOFAF and PHDOFAG respectively.
MAINTAIN examples

Use the following examples to learn what you can do with the MAINTAIN function.

- “MAINTAIN example: Specifying the number of partitions” on page 103
- “MAINTAIN example: Specifying the partition size” on page 104
- “MAINTAIN example: Specifying the key ranges for partitions” on page 105
MAINTAIN example: Specifying the number of partitions

If you know how many partitions you intend to maintain, you can use the PARTNUM parameter to specify this number.

To specify the number of partitions, use the sample member IHCEMTN1. The following example splits a database into four partitions:

```plaintext
//JOBNAME JOB NAME,000,CLASS=A,MSGCLASS=X
//******************************************************************************
// SAMPLE JCL FOR MAINTAIN - using PARTNUM method
//******************************************************************************
//* SLEETS PARTB AND PARTC INTO 4 PARTITIONS
//* Your old HALDB databases/partitions must be registered to DBRC
//* You must /db the affected partitions prior to executing this job
//* unless you use ONLINE(Y) and have Online Reorg installed
//******************************************************************************
//S2 EXEC PGM=IHCHALDB,DYNAMNBR=999,REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD <=== your hcma loadlib
//   DD DISP=SHR,DSN=your.SDFSRESL <=== ims reslib
//DFSRESLB DD DISP=SHR,DSN=your.SHPSLMDB <=== ims tools loadlib
//IMS DD DISP=SHR,DSN=your.dbdlib <=== ims dbdlib
//IMSDALIB DD DISP=SHR,DSN=your.mdalib <=== ims mdalib
//SYSPRINT DD SYSOUT**
//AMSPRINT DD SYSOUT**
//SYSDUMP DD SYSOUT**
//IHCLIST DD SYSOUT**
//IHCSYSIN DD *
MAINTAIN ONLINE(N) -
   PARTNUM(4) -
   DDB(masterdbd) -
   PARTLIST(partb,partc) -
   DBDPATT(*****...) DSNPREF(DBSMS.DB) -
   ICMID(3) -
   ICHLQ(DBSMS.DB.IC) -
   ICTRLR(2) -
   IC.COMP(Y) -
   IC.COMPRTN(FABJCMP2) -
   PTRCHECK(Y)
```

Figure 11. Sample JCL (IHCEMTN1) for batch using PARTNUM
MAINTAIN example: Specifying the partition size

If you know how large you want each partition to be, you can use the PARTSIZE parameter to specify this number.

To specify the size of partitions, use the member IHCEMTN2. The following example splits a database into partitions each with a size of 2048 MB.

```bash
//JOBNAMC JOB NAME,000,CLASS=A,MSGCLASS=X
//********************************************************************
//*                 SAMPLFC FOR MAINTAIN - using PARTSIZE method
//*                  SPLITC PARTC INTO AS MANY 2048 SIZED PARTITIONS
//*   Your old HALDB databases/partitions must be registered to DBRC
//*                  unless you use ONLINE(Y) and have Online Reorg installed
//********************************************************************
//S2 EXEC PGM=IHCHALDB,DYNAMNBR=999,REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD <== your hcma loadlib
// DD DISP=SHR,DSN=your.SDFSRESL <== ims reslib
// DD DISP=SHR,DSN=your.SHPSLMDO <== ims tools loadlib
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL <== ims reslib
//IMS DD DISP=SHR,DSN=your.dbdlib <== ims dbdlib
//IMSDALIB DD DISP=SHR,DSN=your.mdalib <== ims mdalib
//SYSPRINT DD SYSDUT**
//MSGPRINT DD SYSDUT**
//AMSPRINT DD SYSDUT**
//SYSUDUMP DD SYSDUT**
//IHCLIST DD SYSDUT**
//IHCYSIN DD *
MAINTAIN ONLINE(N) -
   PARTSIZE(2048) -
   DBD(masterdbd) -
   PARTITION(partc) -
   DBDPATT(*****...) DSNPREF(DBSMS.DB) -
   ICMID(3) -
   ICHLQ(DBSMS.DB.IC) -
   ICTRLR(2) -
   IC.COMP(Y) -
   IC.COMPRTN(FABJCMP2) -
   PTRCHECK(Y)
```

Figure 12. Sample JCL (IHCEMTN2) for batch using PARTSIZE
MAINTAIN example: Specifying the key ranges for partitions

If you want to assign specific key ranges to partitions, you can use the KEYS parameter to specify an input data set containing the keys. The KEYS parameter specifies the high key boundaries for each partition.

To specify the ranges for partitions, use the member IHCEMTN3. The following example splits the partition that is specified on the PARTITION parameter into 13 partitions with specific key ranges.
//JOBNAME JOB NAME,000,CLASS=A,MSGCLASS=X
//**********************************************************
//SAMPLE JCL FOR MAINTAIN - using KEYS method
//**********************************************************
// SPLITs the first partition into specific key-boundaries
// (note: the first partition had a high key of M9999999999
// and has records for all key ranges)
// Your old HALDB databases/partitions must be registered to DBRC
// You must /dbr the affected partitions prior to executing this job
// unless you use ONLINE(Y) and have Online Reorg installed
//**********************************************************
//S2 EXEC PGM=IHCHALDB,DYNAMNBR=999,REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD <== your hcma loadlib
// DD DISP=SHR,DSN=your.SDFSRESL <== ims reslib
// DD DISP=SHR,DSN=your.SHPSLMD0 <== ims tools loadlib
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL <== ims reslib
//IMS DD DISP=SHR,DSN=your.dbdlib <== ims dbdlib
//IMSDALIB DD DISP=SHR,DSN=your.mdalib <== ims mdalib
//SYSPRINT DD SYSOUT**
//MSGPRINT DD SYSOUT**
//AMSUPRINT DD SYSOUT**
//SYSUDUMP DD SYSOUT**
//IHCLIST DD SYSOUT**
//**********************************************************
//KEYSIN DD *(documentation only)
//your first high-order key for partition1
//your second high-order key for partition2 and so on...
//your highest key for last partition
//**********************************************************
//KEYSIN DD *
A999999999
B999999999
C999999999
D999999999
E999999999
F999999999
G999999999
H999999999
I999999999
J999999999
K999999999
L999999999
M999999999
//IHCSYSIN DD *
MAINTAIN ONLINE(N) -
  KEYS(KEYSIN) -
  DBD(masterdbd) -
  PARTITION(parta) -
  DBDPATT(*****...) DSNPREF(DBSMS.DB) -
  ICMID(3) -
  ICHLQ(DBSMS.DB.IC) -
  ICTRLR(2) -
  IC.COMP(Y) -
  IC.COMPRTN(FABJCMP2) -
  PTRCHECK(Y)

Figure 13. Sample JCL (IHCEMTN3) for batch using KEYS
Part 5. HALDB utilities

JCL for HALDB utilities can be created from the examples in the sample library file, or through the ISPF user interface.

The topics in this section describe how to use the sample JCL provided in the SIHCSAMP file. The sample member is named for each specific utility.

A comment is made if an equivalent ISPF-generated JCL option is available for the utility. See the ISPF section of this reference for ISPF-related information.

Topics:
- Chapter 5, “Using DBRC enhancement utilities,” on page 109
- Chapter 6, “Using the HALDB Analyzer,” on page 129
- Chapter 7, “Reorganizing a PSINDEX,” on page 143
- Chapter 8, “Splitting an unload file into single partition files,” on page 149
- Chapter 9, “Recovering by using the ILK reset utility,” on page 155
- Chapter 10, “Healing index pointers,” on page 161
- Chapter 11, “System utilities,” on page 167
Chapter 5. Using DBRC enhancement utilities

DBRC maintains information that is required for database recoveries, generates recovery control (RECON) statements, verifies recovery input, maintains a separate change log for database data sets, and supports sharing of IMS databases and areas by multiple IMS subsystems.

IMS HALDB Toolkit provides several DBRC enhancement utilities to support HALDB movement between multiple RECONs.

Topics:
- “Backing up DBRC definitions for HALDB DBD” on page 110
- “Cloning DBRC definitions for test environments” on page 115
- “Copying HALDBs to a different IMS system” on page 123
Back up DBRC definitions for HALDB DBD

The DBRC definitions backup utility allows you to save the current DBRC definition of a HALDB to a sequential file.

About this task

You would perform a backup each time before you make changes. In this case, you want to change the partition layout.

The utility creates DBRC statements that can be used as input for the batch DBRC utility (DSPURX00).

Procedure

1. Locate the sample JCL in the SIHCSAMP file, member IHCEDBBK.
2. Specify the JCL DD statements. For a list of DD statements, see “BACKUPDBRC DD statements” on page 111.
3. Specify the BACKUPDBRC command parameters.
   - For the DBD parameter, you must specify the DBD for the HALDB (primary or index).
   - For other parameters, see “BACKUPDBRC command parameters” on page 114.

Example

The following example shows the sample JCL for backing up DBRC definition for HALDB DBD:

```plaintext
//JOBNAME JOB 00,000,CLASS=A,MSGCLASS=X
 بتاريخ-------------------------------
 /* BACKUP DBRC DEFINITION FOR DB
 /*-------------------------------
 //S2 EXEC PGM=IHCALDB,
 // REGION=80M
 // STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
 // DD DISP=SHR,DSN=your.SDFSRESL
 // IMSDALIB DD DISP=SHR,DSN=your.recon.mdalib
 // DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
 // IMS DD DISP=SHR,DSN=your.DBDLIB
 // TRACE DD SYSOUT**
 // MSGPRINT DD SYSOUT**
 // SYSUDUMP DD SYSOUT**
 // IHCSYSIN DD *
 BACKUPDBRC DBD(ddddd) - /* specify HALDB DBD
 INCLIND(Y) - /* include secind
 INCLIC(Y) - /* include batch IC
 TODD(OUTDBRC) /* specify output DDname
 //OUTDBRC DD SYSOUT**
```

Figure 14. JCL for backing up DBRC definition for HALDB DBD
BACKUPDBRC DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the BACKUPDBRC command.

Table 11. DD statements and record format for the BACKUPDBRC command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>DBRCOUT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR,DSN=ihclib
//   DD DISP=SHR,DSN=imstools
//   DD DISP=SHR,DSN=reslib
```

where:

- `ihclib` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.
IMS  This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

   This data set must reside on DASD.

IMSDALIB  This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

   If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

   For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested.

   Dynamic allocation of the database data sets is attempted in the following order:

   1. Dynamic allocation members in the IMSDALIB concatenation
   2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECONx  These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

   Attention: If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

IHCSYSIN  This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

   For more information about the format of IHCSYSIN DD statements, see "BACKUPDBRC command parameters” on page 114.

SYSPRINT  This statement is optional. The statement defines the statistics output data set and output from other utilities that are executed under the control of IMS HALDB Toolkit.

   The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide it.

MSGPRINT  This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

   The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

DBRCOUT  This statement is required if TODD(ddname) parameter is not specified. The statement defines the output data set for generating DBRC statements.

TRACE  This statement is optional. The statement defines the output data set that
IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

SYSUDUMP

This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
## BACKUPDBRC command parameters

The following table summarizes the `BACKUPDBRC` command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DBD</strong></td>
<td>Required</td>
<td>Specifies the name of the HALDB DBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td><strong>INCLIC</strong></td>
<td>Optional</td>
<td>Specifies whether to include the non-fuzzy image copies.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> Y</td>
</tr>
<tr>
<td><strong>INCLIND</strong></td>
<td>Optional</td>
<td>Specifies whether to include the secondary index.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> Y</td>
</tr>
<tr>
<td><strong>TODD</strong></td>
<td>Optional</td>
<td>Specifies the output file where the DBRC statements are written.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must specify either the TODD <em>(ddname)</em> parameter in the IHCSYSIN DD statement or the DBRCOUT DD statement in the JCL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ddname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Cloning DBRC definitions for test environments

In a test environment, the clone DBRC definitions utility allows you to ease DBRC problems by cloning DBRC definitions to an alternate RECON.

About this task

The utility replicates HALDB DBRC definitions to other RECONs and allows you to exchange the high-level data set name qualifier.

You must have your own data sets for testing the batch program, or testing online programs with IMS Batch Terminal Simulator. This requires you to have your own RECONs.

When batch jobs are run with DBRC active, you must accommodate recovery and authorization issues as well. For example, if an application program fails, the database is flagged by DBRC to indicate the need for some recovery action. This utility helps eliminate those problems.

To use this utility, include the sample JCL in your test JCL.

Alternatively, you can create the JCL by using the ISPF user interface.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCECLON.
2. Specify the JCL DD statements. For a list of DD statements, see “COPYDBRC DD statements” on page 118.
3. Specify the COPYDBRC command parameters. For a list of COPYDBRC command parameters, see “COPYDBRC command parameters” on page 120.
4. Include the sample JCL in your test JCL.

Example

In the following sample JCL:

- the.other.RECON is the RECON for the programmer.
- your.RECON is the RECON to be copied from.
- your.DBDLIB is the DBD library for the DBD.
/*------------------------------------------------------------
//*/
// Clone DBRC definitions to alternate RECON
//*/
/*------------------------------------------------------------
//*/
// Allocate RECON datasets
/*------------------------------------------------------------
*/

//ALLO EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
DELETE the.other.RECON1
DELETE the.other.RECON2
DELETE the.other.RECON3
SET MAXCC=0
DEFINE CLUSTER(NAME(the.other.RECON1) VOL(volser) KEYS(320) -  
  CISZ(8192) RCSZ(8100 8192) CYLINDERS(2 2) -  
  FREESPACE(30,10) INDEXED SHR(3 3) SPEED)
DEFINE CLUSTER(NAME(the.other.RECON2) VOL(volser) KEYS(320) -  
  CISZ(8192) RCSZ(8100 8192) CYLINDERS(2 2) -  
  FREESPACE(30,10) INDEXED SHR(3 3) SPEED)
DEFINE CLUSTER(NAME(the.other.RECON3) VOL(volser) KEYS(320) -  
  CISZ(8192) RCSZ(8100 8192) CYLINDERS(2 2) -  
  FREESPACE(30,10) INDEXED SHR(3 3) SPEED)
/*------------------------------------------------------------
//*/

//CRE EXEC PGM=IHCHALDB,  
// REGION=60M  
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD  
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL  
//IMS DD DISP=SHR,DSN=your.DBDLIB  
//RECON1 DD DISP=SHR,DSN=your.RECON1  
//RECON2 DD DISP=SHR,DSN=your.RECON2  
//RECON3 DD DISP=SHR,DSN=your.RECON3  
//MSGPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//IHCSYSIN DD *  
COPYDBRC DBD(nnnnn) -  
  DSNPREF(new.hlq) -  
  DSNDBD(DBD) -  
  INCLIND(YES) -  
  TODD(DBRCOUT)
/*
//DBRCOUT DD DISP=(PASS),SPACE=(TRK,(1,1)),UNIT=SYSALLDA
/*------------------------------------------------------------
//*/

//INIT.RECON
/*------------------------------------------------------------
*/

//INITRCN EXEC PGM=DSPURX00,REGION=50M,COND=(4,LE)  
//STEPLIB DD DISP=SHR,DSN=your.SDFSRESL  
//RECON1 DD DISP=SHR,DSN=the.other.RECON1  
//RECON2 DD DISP=SHR,DSN=the.other.RECON2  
//RECON3 DD DISP=SHR,DSN=the.other.RECON3  
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL  
//IMS DD DISP=SHR,DSN=your.DFSDSRESL  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
INIT.RECON NOFORCER CATDS TAPEUNIT(3480)
/*

Figure 15. Sample JCL from member IHCECLON (Part 1 of 2)
Figure 16. Sample JCL from member IHCECLON (Part 2 of 2)
COPYDBRC DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the COPYDBRC command.

Table 13. DD statements and record format for the COPYDBRC command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMS DALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR,DSN=ihcload
//          DD DISP=SHR,DSN=imstools
//          DD DISP=SHR,DSN=reslib
```

where:

- `ihcload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMS DALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.

IMS

This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.
IMSDALIB
This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested. Dynamic allocation of the database data sets is attempted in the following order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECONx
These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

Attention: If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

IHCSYSIN
This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see "COPYDBRC command parameters" on page 120.

MSGPRINT
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see "MSGPRINT data set" on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE
This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

SYSUDUMP
This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
# COPYDBRC command parameters

The following table summarizes the `COPYDBRC` command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPYIC</td>
<td>Optional</td>
<td>Specifies whether to create NOTIFY.IC statements. COPYIC and DFDSS parameters are mutually exclusive. Format (Y</td>
</tr>
<tr>
<td>DATACLAS</td>
<td>Optional</td>
<td>Represent the SMS constructs to be used for the IDCAMS statements. Format (dataclas) Default value None.</td>
</tr>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the name of the HALDB DBD. Format (dbdname) Default value None.</td>
</tr>
<tr>
<td>DELDBD</td>
<td>Optional</td>
<td>Specifies whether to create DELETE.DB statements. Format (Y</td>
</tr>
<tr>
<td>DFDSS</td>
<td>Optional</td>
<td>Specifies whether to create DFDSS statements for the new data sets. DFDSS parameter is mutually exclusive with COPYIC and IDCAMS parameters. Format (Y</td>
</tr>
<tr>
<td>DFDSSOUT</td>
<td>Conditionally required</td>
<td>Specifies the file name that will contain the DFDSS statements. You must specify this parameter if DFDSS(Y). Format (ddname) Default value None.</td>
</tr>
</tbody>
</table>
Table 14. COPYDBRC command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNDBD</td>
<td>Optional</td>
<td>Specifies the suffix portion of the new dsnname you are using.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A total of 37 characters are allowed for the entire dsnname, which includes the characters used for the DSNPREF prefix and the characters used for the DSNDBD suffix.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NONE   Do not add a suffix to DSNPREF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBD    Add the DBD name to DSNPREF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PART   Add the partition name to DSNPREF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBDPART Add the DBD name and the partition name to DSNPREF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> DBDPART</td>
</tr>
<tr>
<td>DSNPREF</td>
<td>Required</td>
<td>Specifies the prefix portion of the new dsnname you are using.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A total of 37 characters are allowed for the entire dsnname, which includes the characters used for the DSNPREF prefix and the characters used for the DSNDBD suffix.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dsnprefx)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>IDCAMS</td>
<td>Optional</td>
<td>Specifies whether to create IDCAMS statements for the new data sets.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDCAMS and DFDSS parameters are mutually exclusive.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> N</td>
</tr>
<tr>
<td>IDCOUT</td>
<td>Conditionally required</td>
<td>Specifies the file name that will contain the IDCAMS statements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You must specify this parameter if IDCAMS(Y) or DFDSS(Y).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ddname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>INCLIND</td>
<td>Optional</td>
<td>Specifies whether to include the secondary index.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> Y</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>Optional</td>
<td>Represent the SMS constructs to be used for the IDCAMS statements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(storclas)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Table 14. COPYDBRC command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TODD</td>
<td>Required</td>
<td>Specifies the output file where the DBRC statements are written.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This file will be the input to the DBRC utility.</td>
</tr>
<tr>
<td></td>
<td>Format</td>
<td>(ddname)</td>
</tr>
<tr>
<td></td>
<td>Default value</td>
<td>None.</td>
</tr>
<tr>
<td>VOLSER</td>
<td>Conditionally</td>
<td>Specifies the volume serial to be used for IDCAMS.</td>
</tr>
<tr>
<td></td>
<td>required</td>
<td>This parameter is required for non-SMS allocation.</td>
</tr>
<tr>
<td></td>
<td>Format</td>
<td>(volume)</td>
</tr>
<tr>
<td></td>
<td>Default value</td>
<td>None.</td>
</tr>
</tbody>
</table>
Copying HALDBs to a different IMS system

You can copy HALDBs to different IMS systems by using either image copies or DFSMSdss.

In this topic:
- “Copying HALDBs by using an image copy”
- “Copying HALDBs by using DFSMSdss” on page 125

Copying HALDBs by using an image copy

The HALDB copy utility allows you to copy a HALDB, using the latest non-fuzzy image copy, to a different IMS.

About this task

The utility transfers a production HALDB to a different IMS and allocates the target data set in that system, if the data set is not already available.

This task is typically performed when a problem is discovered in the production database, where the database cannot be kept offline until the error in the application has been identified and resolved.

This utility performs the following operations:
- Copies the most recent HALDB definitions across to a different RECON by changing the hlq portion of the data set name.
- Creates the necessary allocation statements.
- Registers all non-fuzzy image copies to the new RECON.

After these operations have been performed, you can run a recovery using the new RECON.

Alternatively, you can create the JCL by using the ISPF user interface.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCECOPY.

2. Specify the JCL DD statements. For a list of DD statements, see “COPYDBRC DD statements” on page 118.

3. Specify the COPYDBRC command parameters. For a list of COPYDBRC command parameters, see “COPYDBRC command parameters” on page 120.
Example

```plaintext
//**----------------------------- Copy a database to different RECON **-----------------------------//
// Once this is complete, do GENJCL.RECOV for that RECON

//CRE EXEC PGM=IHCWHALD,
// REGION=60M
//STELIB DD DISP=SHR,DSN=your.SIHCLOD
// DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//IMS DD DISP=SHR,DSN=your.DBDLIB
//RECON1 DD DISP=SHR,DSN=your.RECON1
//RECON2 DD DISP=SHR,DSN=your.RECON2
//RECON3 DD DISP=SHR,DSN=your.RECON3
//MSGPRINT DD SYSOUT**
//SYSUDUMP DD SYSOUT**
//IHCSYSIN DD *
COPYDBRC DBD(nnnnn) -
   DSNPREF(new.hlq) -
   DSNDBD(DBD) -
   INCNDO(YES) -
   COPYIC(YES) -
   IDCAMS(YES) -
   IDCOUT(IDCOUT) -
   DATACLAS(DCL) -
   STORCLAS(STOCL) -
   VOLSER(VOL001) -
   TODD(DBRCOUT)
/
//DBRCOUT DD DISP=(,PASS),SPACE=(TRK,(1,1)),UNIT=SYSALLDA
//IDCOUT DD DISP=(,PASS),SPACE=(TRK,(1,1)),UNIT=SYSALLDA

//**----------------------------- APPLY TO OTHER DBRC **-----------------------------//

//DBRC EXEC PGM=DSPURX00,REGION=50M,COND=(4,LE)
//STELIB DD DISP=SHR,DSN=your.SDFSRESL
//RECON1 DD DISP=SHR,DSN=the.other.RECON1
//RECON2 DD DISP=SHR,DSN=the.other.RECON2
//RECON3 DD DISP=SHR,DSN=the.other.RECON3
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//IMS DD DISP=SHR,DSN=your.DBDLIB
//SYSIN DD DISPP=(OLD,DELETE),DSN=*.CRE.IDCOUT

//**----------------------------- ALLOCATE NEW FILES **-----------------------------//

//IDC EXEC PGM=IDCAMS,DYNAMNBR=99,
//   REGION=50M,COND=(4,LE)
//SYSIN DD DISPP=(OLD,DELETE),DSN=*.CRE.IDCOUT
```

Figure 17. Sample JCL from member IHCECOPY
Copying HALDBs by using DFSMSdss

You can use DFSMSdss to copy a HALDB to a different IMS.

About this task

Using DFSMSdss works like image copy recovery. The target RECON must reflect any deleted partitions and you must /DBR the database.

You can create the partition definition for the target RECON and create the DFSMSdss commands that are necessary to make the copy.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEDFDS.
2. Specify the JCL DD statements. For a list of DD statements, see "COPYDBRC DD statements" on page 118.
3. Specify the COPYDBRC command parameters. For a list of COPYDBRC command parameters, see "COPYDBRC command parameters" on page 120.
4. Make sure that the target RECON reflects any deleted partitions.
5. Run /DBR on the database.
Example

```c
/*-----------------------------------------------*/
/* Copy a database to different RECON */
/* The copy function is done via DFDSS */
/*-----------------------------------------------*/
//CRE EXEC PGM=IHCWHALD, REGION=60M
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
// DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//IMS DD DISP=SHR,DSN=your.DBDLIB
//RECON1 DD DISP=SHR,DSN=your.RECON1
//RECON2 DD DISP=SHR,DSN=your.RECON2
//RECON3 DD DISP=SHR,DSN=your.RECON3
//MSGPRINT DD SYSOUT**
//SYSUDUMP DD SYSOUT**
//IHCSYSIN DD *
COPYDBRC DBD(nn.nn) -
   DSNPREF(new.hlq) -
   DSNDBD(DBD) -
   INCLIND(YES) -
   DFDSS(YES) -
   DFDSSOUT(DSSOUT) -
   IDCOUT(IDCOUT) -
   T00D(DBRCOUT)
/*
//DBRCOUT DD DISP=(,PASS),SPACE=(TRK,(1,1)),UNIT=SYSALLDA
//DSSOUT DD DISP=(,PASS),SPACE=(TRK,(1,1)),UNIT=SYSALLDA
//IDCOUT DD DISP=(,PASS),SPACE=(TRK,(1,1)),UNIT=SYSALLDA
*/
/*-----------------------------------------------*/
/* APPLY TO OTHER DBRC */
/*-----------------------------------------------*/
//DBRC EXEC PGM=IHCWHALD, REGION=60M
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
// DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//RECON1 DD DISP=SHR,DSN=the.other.RECON1
//RECON2 DD DISP=SHR,DSN=the.other.RECON2
//RECON3 DD DISP=SHR,DSN=the.other.RECON3
//IMS DD DISP=SHR,DSN=your.DBDLIB
//TRACE DD DUMMY
//SYSPRINT DD SYSOUT**
//MSGPRINT DD SYSOUT**
//SYSIN DD DISP=(OLD,DELETE),DSN=*.CRE.DBRCOUT
//IHCSYSIN DD *
RUN PGM(IHCYDBR0)
```

Figure 18. Sample JCL from member IHCEDFDS (Part 1 of 2)
Figure 19. Sample JCL from member IHCEDFDS (Part 2 of 2)
Chapter 6. Using the HALDB Analyzer

The HALDB Analyzer utility checks the health of a HALDB by verifying HALDB constructs within a database and its indexes.

About this task

The utility analyzes the HALDB structure and verifies that all root keys are in the correct partition, that all ILKs are valid, and that all EPS pointers will find the correct ILE. Additional performance-related data is also collected.

The HALDB Analyzer monitors specific HALDB components. You can provide thresholds that trigger the utility to issue critical warnings when these components are not performing properly. These components include:

Partition selection
A root segment in a partition is not found in the same partition when searched by partition selection.

Randomizer error
A call to the randomizer using an existing root segment does not return a zero return code. This warning is performed for the standard IMS randomizer (DFSHDCnn) only.

ILKs in error
The ILK of a segment is invalid.

Duplicate ILKs
There should not be any duplicate ILKs.

Missing ILEs
EPS pointer requires pointer healing, but there is no corresponding ILE.

A return code of 8 is set when any of these situations is detected.

Performance and administration parameters allow you to set threshold levels for the components being monitored by the HALDB Analyzer.

The threshold values for performance and administration parameters are set as percentage numbers from 0 to 100, unless otherwise stated.

The parameters produce two levels of warning operands:

Normal warning level
The warning level operand notifies you of a problem situation, but allows you some time to resolve the problem.

Critical warning level
The critical level operand informs you that an immediate action needs to be taken to correct the problem.

A value of 0 in any operand suppresses the notification. However, the report still shows the data. There are no default settings for the parameters. When the keyword is omitted, no threshold warning is issued.
Procedure

1. Locate the sample JCL for the HALDB Analyzer utility in the SIHCSAMP file, member IHCEANA.

2. Specify the JCL DD statements. For a list of DD statements, see "CHECK DD statements" on page 131.

3. Specify the CHECK command parameters. For a list of CHECK command parameters, see "CHECK command parameters" on page 134.

Example

```//*-----------------------------------------------------------
//*
//*    Analyze HALDB
//*
//*-----------------------------------------------------------
//S2    EXEC PGM=IHCHALDB,
//      REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
//RECON1 DD DISP=SHR,DSN=your.RECON1
//RECON2 DD DISP=SHR,DSN=your.RECON1
//RECON3 DD DISP=SHR,DSN=your.RECON1
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//IMS    DD DISP=SHR,DSN=your.DBDLIB
//TRACE  DD SYSOUT**,//KEYSOUT DD DISP=(,PASS),SPACE=(CYL,(1,1)),UNIT=SYSDA,
//MSGPRINT DD SYSOUT**,//SYSUDUMP DD SYSOUT**
//IHCSYSIN DD *
  CHECK DBD(PHDO2) -
    MINROOT(500) -
    VERIFYILK(Y) -
    VERIFYILE(Y) -
    KEYS(KEYSOUT) -
    SPACE(300,2500) -
    4GBWARN(3000,3500) -
    EXTENT(20,40) -
    ROOTOVFL(1,30) -
    ROOTHOME(2,20) -
    SPLIT(5,10) -
    PTRHEAL(20,80) -
    AVGFSE(1,30) -
    RAPUSE(60,80)
  //KEYSOUT DD DISP=(,PASS),SPACE=(CYL,(1,1)),UNIT=SYSDA,
  //  DSN=&&KEYS```

Figure 20. JCL member IHCEANA
CHECK DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the CHECK command.

Table 15. DD statements and record format for the CHECK command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCLIST</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR,DSN=ihcload
//    DD DISP=SHR,DSN=imstools
//    DD DISP=SHR,DSN=reslib
```

where:

- `ihcload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.
IMS  This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.
This data set must reside on DASD.

IMSDALIB  This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.
If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.
For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested.
Dynamic allocation of the database data sets is attempted in the following order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECONx
These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

Attention: If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

IHCSYSIN
This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.
For more information about the format of IHCSYSIN DD statements, see “CHECK command parameters” on page 134.

MSGPRINT
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.
The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

IHCLIST
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for reports. For more information about the reports generated in this data set, see “IHCLIST data set” on page 303.
The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE
This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.
This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

**SYSUDUMP**  
This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
CHECK command parameters

The following table summarizes the CHECK command parameters.

Table 16. CHECK command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| 4GBWARN   | Optional    | Sets the warning level for used space. None of the data sets can use more than 4 GB (gigabytes). When the data set tries to go beyond this limit, an abend 844 will occur. This is not the high allocation point, but rather the high use point. This point can be indicated by the threshold setting. 4GBWARN and 8GBWARN are mutually exclusive.  
  
  **Format** (llll,hhhh)  
  
  **llll** Sets the low threshold. Specify a number (in MB) in the range of 0 - 4095. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.  
  
  **hhhh** Sets the high threshold. Specify a number (in MB) in the range of 0 - 4095. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.  
  
  **Default value** None.  
  
  **Usage** A recommended setting is: 4GBWARN(2500,3500)  
  
  **Resolution** When the partition grows too large, you should split it into two partitions. |
| 8GBWARN   | Optional    | Sets the warning level for used space. None of the data sets can use more than 8 GB (gigabytes). When the data set tries to go beyond this limit, an abend 844 will occur. This is not the high allocation point, but rather the high use point. This point can be indicated by the threshold setting. 4GBWARN and 8GBWARN are mutually exclusive.  
  
  **Format** (llll,hhhh)  
  
  **llll** Sets the low threshold. Specify a number (in MB) in the range of 0 - 8191. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.  
  
  **hhhh** Sets the high threshold. Specify a number (in MB) in the range of 0 - 8191. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.  
  
  **Default value** None.  
  
  **Usage** A recommended setting is: 8GBWARN(6500,7500)  
  
  **Resolution** When the partition grows too large, you should split it into two partitions. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVGFSE</td>
<td>Optional</td>
<td>Describes the number of free space elements (FSE) per block. This parameter can indicate that a reorganization is needed. An FSE describes a portion of free space in a block. There should only be one FSE that describes one large chunk of free space. However, after any update activity, that free space can become multiple smaller pieces. This is especially true with compressed segments.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(lll,hhh)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>lll</em> Sets the low threshold for the number of FSEs per block. Specify a number in the range of 0 - 999. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>hhh</em> Sets the high threshold for the number of FSEs per block. Specify a number in the range of 0 - 999. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Usage</strong> The low and high values are dependent on the segment size. You want to set this value smaller with large segments. A recommended setting is: <em>AVGFSE(5,10)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Resolution</strong> A reorganization will return the free space back to one block.</td>
</tr>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the name of the HALDB DBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| EXTENT    | Optional    | Indicates the extent number that triggers a warning. The maximum number of extents allowed depends on the data management type. **VSAM** The maximum number of extents per data set is 251. The maximum number of extents per volume is 123. **OSAM** The maximum number of extents per data set depends on the version of IMS; approximately 60 for IMS V13, and approximately 120 for IMS V14 and later. The maximum number of extents per volume is 16. When no more extents are possible, an attempt to obtain a new one creates an out-of-space condition, which leads to an 844 IMS abend. The purpose of the EXTENT parameter settings is to prevent this occurrence. **Format** (lll,hhh)  
  
  \[lll\] Sets the low threshold. Specify a number in the range of 0 - 251. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.  
  
  \[hhh\] Sets the high threshold. Specify a number in the range of 0 - 251. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed. **Default value** None. **Usage** A recommended setting is: EXTENT(25,35) **Resolution** Consider dividing this partition into two partitions using the functions provided with this utility. If the space usage is not very high, adjust the primary and the secondary space parameter to allow fewer extents. It is also possible that your SMS pool volumes are heavily split, so that each space request must be satisfied using multiple extents. In this situation, your volumes should be maintained.
Table 16. CHECK command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYS</td>
<td>Optional</td>
<td>The designated ddname will receive a file containing the root keys.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ddname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Usage</strong> Use this parameter if you want to perform a root key analysis; otherwise, do not specify this parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The KEYS file is created upon request only. The file only exists when the KEYS parameter has been specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following example shows the layout of the file:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PID DS XL2  Current partition id</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REORGNO DS XL2  Current reorg number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SEGRBA DS XL4  Current RBA (data portion if split)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARGPID DS XL2  Target partition id</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMRESULT DS XL4  Randomizer result</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KEY DS 0C  Start of root key</td>
</tr>
<tr>
<td>MINROOT</td>
<td>Optional</td>
<td>Defines the minimum amount of root segments in the database for the segment-oriented thresholds. If there are less than the specified root segments, some warnings are suppressed. This is because the percentage values do not perform well with small databases. The thresholds are suppressed for ROOTHOME, ROOTOVFL, SPLIT, and PTRHEAL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(nnnnnn)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> 0</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| PTRHEAL   | Optional    | Indicates the amount of EPS pointer (PSINDEX ONLY) that needs pointer healing. The secondary index EPS pointer is not maintained when reorganizing the primary database (or a partition). In this situation, pointer healing (updating the direct pointer from the index to its target) must be done when the pointer is used. The ILDS data set is used.

This process creates many I/Os. Therefore, it is recommended that you have all pointers healed. This parameter can help you decide when to perform this task.

**Format** \((ll, hh)\)

\(ll\) Sets the low threshold. Specify a percentage number in the range of 0 - 99. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.

\(hh\) Sets the high threshold. Specify a percentage number in the range of 0 - 99. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.

**Default value**
None.

**Usage**
A recommended setting is: \(\text{PTRHEAL}(20, 60)\)

**Resolution**
You can use the pointer healer function provided by IMS HALDB Toolkit or rebuild the indexes using an index builder utility.

| RAPUSE | Optional | Indicates the usage of root anchor points (RAPs) in a PHDAM database. This parameter is used only for PHDAM. RAP usage is very dependent on the randomizing module. This parameter is designed for DFSHDC40, but can be used for others as well. A performance criteria for DFSHDC40 is that it should have more RAPs than root segments. This condition leads to RAPs not being used. The RAPUSE parameter monitors this usage.

**Format** \((ll, hh)\)

\(ll\) Sets the low threshold. Specify a percentage number in the range of 0 - 99. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.

\(hh\) Sets the high threshold. Specify a percentage number in the range of 0 - 99. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.

**Default value**
None.

**Usage**
A recommended setting is: \(\text{RAPUSE}(60, 80)\)

**Resolution**
The randomizing parameter must be changed. See ROOTHOME for details.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOTHOME</td>
<td>Optional</td>
<td>Indicates the number of root segments that are not in home block. This parameter is effective for PHDAM only. It is ignored for PHIDAM. PHDAM has a randomizing module that addresses a root anchor point (RAP). The root segment is chained off. For performance reasons, the root segment should be residing in the same block. However, this is not always the case. This parameter allows you to set thresholds on how many root segments are not in that block.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format ((ll, hh))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ll) Sets the low threshold. Specify a percentage number in the range of 0 - 99. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(hh) Sets the high threshold. Specify a percentage number in the range of 0 - 99. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Usage Do not allow too many roots in the home block. A recommended setting is: (\text{ROOTHOME}(10, 30))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Resolution The randomizing parameter must be changed. In most cases, you must increase the RAASIZE. You might also have too many RAPs per block. In this case, decrease the RAPs per block and increase the RAASIZE accordingly. Remember that the randomizing parameter is defined with the partition, so you must perform this task using the DBRC utility and CHANGE.PART. You can use the MAINTAIN function to resolve this.</td>
</tr>
<tr>
<td>ROOTOVFL</td>
<td>Optional</td>
<td>Indicates how many root segments are in overflow. This parameter applies to PHDAM only. It is like the ROOTHOME parameter. However, this situation occurs when many new root segments are inserted. All roots in the overflow area are also reflected by ROOTHOME, but these roots could be brought back by a simple reorganization.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Format ((ll, hh))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ll) Sets the low threshold. Specify a percentage number in the range of 0 - 99. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(hh) Sets the high threshold. Specify a percentage number in the range of 0 - 99. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default value None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Resolution The resolution is to change the randomizing parameter for the indicated partition.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SPACE</td>
<td>Optional</td>
<td>Sets thresholds for partition size. This parameter can be used to make decisions about partitions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (llll,hhhh)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>llll</strong> Sets the lowest preferred partition size. Specify a number (in MB) in the range of 0 - 8191. You will get a warning when the partition size is smaller than this value. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>hhhh</strong> Sets the high partition size. Specify a number (in MB) in the range of 0 - 8191. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Usage</strong> A recommended setting is: SPACE(512,3500)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Resolution</strong> When a partition is too small, you might want to combine it with the left or right (or both) neighboring partition. The IMS HALDB Toolkit provides the functionality to do this. When a partition becomes too large, it should be split into two partitions.</td>
</tr>
<tr>
<td>SPLIT</td>
<td>Optional</td>
<td>Indicates how many of the split segments are split to a different block. A variable length segment could be split into the prefix and the data portion. This can happen during an update when the size of the data increases, but there is not enough room to update this in place. The prefix portion cannot be moved and needs to stay in place. The data portion is now stored at a different location and is pointed to by an additional pointer in the prefix. This becomes a problem when the different location is in a different block. That situation will cause an additional I/O when retrieving the segment. The parameter threshold settings address this specific situation. The parameter is active for each segment type that has variable length or is compressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> (ll,hh)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ll</strong> Sets the low threshold for all segments of a segment type that is split to a different block. Specify a percentage number in the range of 0 - 99. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>hh</strong> Sets the high threshold. Specify a percentage number in the range of 0 - 99. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Usage</strong> A recommended setting is: SPLIT(5,20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Resolution</strong> A reorganization will resolve this problem.</td>
</tr>
</tbody>
</table>
### Table 16. CHECK command parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNONYM</td>
<td>Optional</td>
<td>Indicates the number of root segment occurrences in the synonym chain. This parameter is effective for PHDAM only. It is ignored for PHIDAM.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(ll, hh)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>ll</em> Sets the low threshold. Specify a percentage number in the range of 0 - 99. You will get a warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>hh</em> Sets the high threshold. Specify a percentage number in the range of 0 - 99. You will get a critical warning when this value is exceeded. If you specify a value of 0, this warning is suppressed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Resolution</strong> The randomizing parameter must be changed. In most cases, you must increase the RAASIZE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You might also have too few RAPs per block. In this case, increase the RAPs per block and increase the RAASIZE accordingly. Remember that the randomizing parameter is defined with the partition, so you must perform this task using the DBRC utility and CHANGE.PART. You can use the MAINTAIN function to resolve this.</td>
</tr>
<tr>
<td>VERIFYILE</td>
<td>Optional</td>
<td>When EPS pointers are present, they might require pointer healing. Pointer healing requires that the targeted partition has an ILE (a record in the ILDS) to be used for healing. If this record is not there, the situation is the equivalent of a pointer error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> N</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Usage</strong> This condition should be checked regularly.</td>
</tr>
<tr>
<td>VERIFYILK</td>
<td>Optional</td>
<td>In HALDB, each segment has a unique indirect list key (ILK) that can never be the same for the same segment type in the entire database. The pointer healing process relies on this uniqueness. This condition should be treated like a pointer, so its verification is important. The function provided will check for duplicate and invalid ILK.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> N</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Usage</strong> This condition should be checked regularly.</td>
</tr>
</tbody>
</table>
Chapter 7. Reorganizing a PSINDEX

The reorganize PSINDEX utility helps you reorganize a partitioned secondary index (PSINDEX) to improve space management due to heavy update activity, where the number of CI splits and CA splits would be high.

About this task

The following tasks can be performed by this utility:

- Reclaim space
- Reclaim split indexes
- Remove deleted index records
- Remove index records for a dropped primary partition

Although this reorganization task could be performed with IDCAMS REPRO, the task would not be registered with DBRC.

The following example shows another way to use the REORGIND command to reorganize a PSINDEX:

```
REORGIND DBD(dbname) PARTITION(psindex_partition_name)
```

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCERE0I.
2. Specify the JCL DD statements. For a list of DD statements, see "REORGIND DD statements" on page 145.
3. Specify the REORGIND command parameters. For a list of REORGIND command parameters, see "REORGIND command parameters" on page 148.
Example

//JOBNAME JOB 00,000,CLASS=A,MSGCLASS=X
//****************************************************************************
//** Reorg partition
//****************************************************************************
//REO EXEC PGM=IHCHALDB,
  //  REGION=60M
  //STEPLIB DD DISP=SHR,DSN=your.SIHCLD
  //  DD DISP=SHR,DSN=your.SDFSRESL
  //RECON1 DD DISP=SHR,DSN=your.RECON1
  //RECON2 DD DISP=SHR,DSN=your.RECON2
  //RECON3 DD DISP=SHR,DSN=your.RECON3
  //DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
  //IMS DD DISP=SHR,DSN=your.DBDLIB
  //MSGPRINT DD SYSOUT**
  //SYSUDUMP DD SYSOUT**
  //IHCSYSIN DD *
    REORGIND DBD(SHO22)
  /*

Figure 21. Sample JCL for reorganizing a PSINDEX
REORGIND DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the REORGIND command.

Table 17. DD statements and record format for the REORGIND command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**STEPLIB**

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```bash
//STEPLIB DD DISP=SHR, DSN=ihcload
// DD DISP=SHR, DSN=imstools
// DD DISP=SHR, DSN=reslib
```

where:

- *ihcload* is the name of the library that contains the IMS HALDB Toolkit load modules.
- *imstools* is the optional name of the library that contains the IMS reorganization utilities load modules.
- *reslib* is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

**DFSRESLB**

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.
IMS  This statement is required. The statement describes the library that contains
the DBDs of the database that you are processing.
This data set must reside on DASD.

IMSDALIB  This statement is optional. The statement specifies a partitioned data set or
data sets that contain the dynamic allocation members for the database
data sets of non-HALDB and the RECON data sets.
If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is
always referred to by DBRC, and DBRC determines the data sets to be
allocated for each RECON DD.
For the database data sets of non-HALDB, IMSDALIB is referred to only
when the dynamic allocation for the original data sets is requested.
Dynamic allocation of the database data sets is attempted in the following
order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECONx  These statements are optional. The statements define the database recovery
control (DBRC) RECON data sets. RECON data sets dictate which
databases and IMS online systems to use during the processing.

Attention: If you specify IMSDALIB DD dynamic allocation, do not use
these RECON DD statements. If you used JCL to allocate RECON1 as a
spare data set, you must restart the database.

IHCSYSIN  This statement is required. The statement defines a control statement or
data set that contains control statements that specify the functions of IMS
HALDB Toolkit.
For more information about the format of IHCSYSIN DD statements, see
“REORGIND command parameters” on page 148.

SYSPRINT  This statement is optional. The statement defines the statistics output data
set and output from other utilities that are executed under the control of
IMS HALDB Toolkit.
The data set can reside on DASD, or it can be routed through the output
job. IMS HALDB Toolkit dynamically allocates this data set if you do not
provide it.

MSGPRINT  This statement is optional. The statement defines the IMS HALDB Toolkit
output data set for progress messages issued during the job. For more
information, see “MSGPRINT data set” on page 302.
The data set can reside on DASD, or it can be routed through the output
job. IMS HALDB Toolkit dynamically allocates this data set if you do not
provide this DD statement.

TRACE  This statement is optional. The statement defines the output data set that
IMS HALDB Toolkit uses to write diagnostic trace records. This data set
might be required for problem diagnosis.
This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

SYSUDUMP
This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
## REORGIND command parameters

The following table summarizes the `REORGIND` command parameters.

*Table 18. REORGIND command parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| DBD       | Required    | Specifies the name of a primary DBD or a PSINDEX.  
If you specify a primary DBD, PARTITN must not be specified. All PSINDEX partitions for the specified primary DBD are reorganized.
If you specify a PSINDEX DBD, you can specify a partition. If a partition is not specified or specified with an asterisk (*), then all partitions of that PSINDEX are reorganized.  
**Format** (dbname)  
**Default value** None. |
| PARTITN   | Optional    | Specifies the name of the PSINDEX partition. If the DBD is PSINDEX, the default name value is an asterisk (*).  
This parameter is invalid when a primary DBD is specified.  
**Format** (psindex_partition_name | *)  
**Default value** * |
Chapter 8. Splitting an unload file into single partition files

The split unload file utility allows you to split or consolidate an unload file into single partition unload files. The reloads then can run in parallel.

About this task

A HALDB unload file can contain data from multiple partitions. As the data grows or shifts, partitions grow or shrink. When reloading partitions, it would be faster to reload each partition in parallel.

The single partition unload files are standard Hierarchical Direct (HD) unload files with header and trailer statistics. Each file contains only records for the selected partition.

This utility separates the HALDB unload file on the partition boundary and maintains the unload header and trailer records for each file.

The sample JCL for this utility is in the SIHCSAMP file, member IHCESPLT.

You must specify the DBD and the unload file you want to split.

The DD name for the single partition HD unload files are constructed using the partition name appended with a Z. These files are required.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCESPLT.
2. Specify the JCL DD statements. For a list of DD statements, see "IHCUSPLT DD statements" on page 151.
3. Specify the IHCUSPLT command parameters. For a list of IHCUSPLT command parameters, see "IHCUSPLT command parameters" on page 154.

Example

In the following example, each partition unload file is named PARTxxZ.
//JOBNAME JOB 00,000,CLASS=A,MSGCLASS=X
//-----------------------------------------------
// Split unload file
// Input DD is "DFSUINPT"
// Output DD is partition name + "Z"
//-----------------------------------------------
//S2 EXEC PGM=IHCHALDB,
// REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIWCHLOAD
// DD DISP=SHR,DSN=your.SDFSRESL
//RECON1 DD DISP=SHR,DSN=your.RECON1
//RECON2 DD DISP=SHR,DSN=your.RECON2
//RECON3 DD DISP=SHR,DSN=your.RECON3
//IMS DD DISP=SHR,DSN=your.DBDLIB
//TRACE DD SYSOUT=
//SYSABEND DD SYSOUT=
//MSGPRINT DD SYSOUT=
//IHCSYSIN DD *
// RUN PGM(IHCSPLT) -
// DBD(nnnnn) */ === DBD name
//DFSUINPT DD DISP=SHR,DSN=your.hd.unload.dataset
//PART01Z DD DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(50,20)),
// DSN=&PART01
//PART02Z DD DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(50,20)),
// DSN=&PART02

Figure 22. Sample JCL member IHCESPLT
IHCUSPLT DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the IHCUSPLT command.

Table 19. DD statements and record format for the IHCUSPLT command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>DFSUINPT</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IHCYSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>partition_nameZ</td>
<td>Output</td>
<td></td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```//STEPLIB DD DISP=SHR,DSN=ihclload // DD DISP=SHR,DSN=imstools // DD DISP=SHR,DSN=reslib```

where:

- `ihclload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.
IMS  This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

IMSDALIB  This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested. Dynamic allocation of the database data sets is attempted in the following order:

1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECONx  These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

Attention:  If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

DFSUINPT  This statement is required. The statement defines the unload file you want to split.

IHCSYSIN  This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “IHCSPLT command parameters” on page 154.

partition_nameZ  These statements are required. Each of these statements defines the HD unload file for each partition. The DD names for the single partition HD unload files are constructed using the partition name appended with a Z.

MSGPRINT  This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE  This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.
This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

**SYSUDUMP**
This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
IHCUSPLT command parameters

The following table summarizes the IHCUSPLT command parameters.

Table 20. IHCUSPLT command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the name of a primary DBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Chapter 9. Recovering by using the ILK reset utility

Recovery is the process of rebuilding databases after a system failure. IMS HALDB Toolkit provides the ILK utility to help you recover from the database rebuild process.

About this task

In a HALDB, an indirect list entry key (ILK) is a unique token that is assigned to a segment in PHDAM and PHIDAM databases when the segment is created. Eight bytes in length and stored in the prefix of the segment, the ILK uniquely identifies every segment in PHDAM and PHIDAM databases.

The ILK reset utility enables you to rebuild all ILKs and their references, and allows you to resolve conflicting ILKs.

The ILK reset utility resets the ILK of each segment. The utility offers a solution when duplicate or invalid ILKs are encountered. Run this utility when the HALDB Analyzer detects invalid ILKs.

This utility does not support logical relations. It creates an HD unload file, which contains the new ILKs. This file can be used as input to reload.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEILK.
2. Specify the JCL DD statements. For a list of DD statements, see “RESETILK DD statements” on page 157.
3. Specify the RESETILK command parameters. For a list of RESETILK command parameters, see “RESETILK command parameters” on page 160.
4. Make sure that an index builder utility is available because secondary indexes must be rebuilt.
5. Take the databases offline.
Example

//JOBNAME JOB 00,000,CLASS=A,MSGCLASS=X
//*******************************************************************************
//    Reset ILK
//  A new HD unload file is created containing
//  new ILKs. Reload database using this unload file.
//  Secondary indexes must be rebuild
//*******************************************************************************
//S2 EXEC PGM=IHCHALDB,
//    REGION=50M
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
//    DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESL DD DISP=SHR,DSN=your.SDFSRESL
//RECON1 DD DISP=SHR,DSN=your.RECON1
//RECON2 DD DISP=SHR,DSN=your.RECON2
//RECON3 DD DISP=SHR,DSN=your.RECON3
//IMS DD DISP=SHR,DSN=your.DBDLIB
//TRACE DD SYSOUT=*  
//MSGPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//IHCSYSIN DD *
//    RESETILK DBD(ddd)
//DFSUGU1 DD UNIT=SYSDA,DISP=(,PASS),
//    DSN=&&UNL,SPACE=(CYL,(1,1))
RESETILK DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the RESETILK command.

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>DFSURGU1</td>
<td>Output</td>
<td></td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**STEPLIB**

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```//STEPLIB DD DISP=SHR,DSN=ihcload
// DD DISP=SHR,DSN=imstools
// DD DISP=SHR,DSN=reslib```

where:

- `ihcload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

**DFSRESLB**

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.
IMS  This statement is required. The statement describes the library that contains the DBDs of the database that you are processing. This data set must reside on DASD.

IMSDALIB  This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested. Dynamic allocation of the database data sets is attempted in the following order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECONx  These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

Attention:  If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

IHCSYSIN  This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “RESETILK command parameters” on page 160.

DFSURGU1  This statement is required. The statement defines a new HD unload file for storing new ILKs.

MSGPRINT  This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE  This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis. This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

SYSUDUMP  This statement is required only if a dump is requested by IBM Software
Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
RESETILK command parameters

The following table summarizes the RESETILK command parameters.

Table 22. RESETILK command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the name of a primary DBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Chapter 10. Healing index pointers

After reorganizing the primary database, all secondary index pointers require healing. HALDB can handle this situation with its self-healing feature. Indexes can provide direct access to data, which otherwise can be found only by scanning the entire database.

About this task

HALDB identifies the new direct pointer by using the indirect list data set (ILDS) and replaces the invalid pointer with the current pointer. This process is called pointer healing.

However, this task might demand extra resources that can affect performance during peak time.

The index pointer healer utility allows you to heal index pointers offline and at a less expensive time. The utility allows you to replace, or heal, all index pointers that have become invalid after a database reorganization. The utility updates the index using the ILDS, which is an efficient process because it works sequentially.

When the primary database (or its partitions) is reorganized, the secondary indexes are not rebuilt. However, the target segment in the primary database has moved to a different place (the relative byte address or RBA). This condition makes the direct pointer in the index invalid.

Pointer healing takes place only when required (for example, when the secondary index is used to locate the target). This process requires many I/Os. It is recommended that you avoid pointer healing during peak transaction time.

The following conditions apply to using the index pointer healer utility:

• The primary database must not be open for update.
• The secondary indexes must be offline.

Alternatively, you can create the JCL by using the ISPF user interface.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEIXHL.
2. Specify the JCL DD statements. For a list of DD statements, see "PTRHEAL DD statements" on page 163.
3. Specify the PTRHEAL command parameters. For a list of PTRHEAL command parameters, see "PTRHEAL command parameters" on page 166.
4. Make sure that the primary database is closed for update.
5. Make sure that the secondary indexes are offline.
Example

```c
//JOBNAME  JOB 00,000,CLASS=A,MSGCLASS=X
//@*------------------------------------------------------------
//@* Pointer healing for secondary indexes
//@* for index DBD specified.
//@*------------------------------------------------------------
//@S2 EXEC PGM=IHCHALDB,
//@    REGION=50M
//@STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
//@    DD DISP=SHR,DSN=your.SDFSRESL
//@DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//@RECON1 DD DISP=SHR,DSN=your.RECON1
//@RECON2 DD DISP=SHR,DSN=your.RECON2
//@RECON3 DD DISP=SHR,DSN=your.RECON3
//@IMS DD DISP=SHR,DSN=your.DBDLIB
//@TRACE DD SYSDUMP==
//@MSGPRINT DD SYSDUMP==
//@SYSUDUMP DD SYSDUMP==
//@IHCSYSIN DD *
//@    PTRHEAL DBD(nnnnn)
```

Figure 24. Sample Index Pointer Healer JCL
PTRHEAL DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the PTRHEAL command.

Table 23. DD statements and record format for the PTRHEAL command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

//STEPLIB DD DISP=SHR,DSN=ihclod
//    DD DISP=SHR,DSN=imstools
//    DD DISP=SHR,DSN=reslib

where:

- *ihclod* is the name of the library that contains the IMS HALDB Toolkit load modules.
- *imstools* is the optional name of the library that contains the IMS reorganization utilities load modules.
- *reslib* is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.

IMS

This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.
This data set must reside on DASD.

**IMSDALIB**

This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested. Dynamic allocation of the database data sets is attempted in the following order:

1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

**RECONx**

These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

**Attention:** If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

**IHCSYSIN**

This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “PTRHEAL command parameters” on page 166.

**MSGPRINT**

This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

**TRACE**

This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

**SYSUDUMP**

This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.
This data set can reside on DASD, or it can be routed through the output job.
PTRHEAL command parameters

The following table summarizes the PTRHEAL command parameters.

Table 24. PTRHEAL command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the PSINDEX DBD name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Chapter 11. System utilities

IMS HALDB Toolkit provides several utilities to help you with database handling tasks.

Topics:
• “Rebuilding DBD source from DBDLIB” on page 168
• “Creating a PSB list for selected DBDs” on page 172
• “Creating OSAM multi-volume data sets” on page 175
• “Listing VIO and unit names” on page 176
Rebuilding DBD source from DBDLIB

The rebuild DBD source utility rebuilds DBD source statements from DBDLIB. If you are not sure that your DBD source reflects the DBDLIB, you can use this utility to rebuild the DBD source. The DBDLIB contains the most recent DBD description.

About this task

DBD source statements are compiled to create the DBD description in DBDLIB. The rebuild DBD source utility reverses that process.

A database description (DBD) is the collection of macro parameter statements that define the characteristics of a database, such as the database's organization and access method, the segments and fields in a database record, and the relationship between types of segments.

The output from this job is written to the OUTPUT file.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEDBD.
2. Specify the JCL DD statements. For a list of DD statements, see “IHCUDBD DD statements” on page 169.
3. Specify the IHCUDBD command parameters. For a list of IHCUDBD command parameters, see “IHCUDBD command parameters” on page 171.

Example

```//JOBNAME JOB 00,000,CLASS=A,MSGCLASS=X //************************************************************************ // Rebuild DBD source //************************************************************************ //S2 EXEC PGM=IHCHALDB, // REGION=50M //STEPLIB DD DISP=SHR,DSN=your.SIHLOAD // DD DISP=SHR,DSN=your.SDFSRESL //RECON1 DD DISP=SHR,DSN=your.RECON1 //RECON2 DD DISP=SHR,DSN=your.RECON2 //RECON3 DD DISP=SHR,DSN=your.RECON3 //IMS DD DISP=SHR,DSN=your.DBDLIB //TRACE DD DUMMY //MSGPRINT DD SYSOUT=* //OUTPUT DD SYSOUT=* //IHCSYSIN DD * RUN PGM(IHCUDBD) - DBD(ddddd) <== specify DBD name```

Figure 25. Sample JCL member IHCEDBD
**IHCUDBD DD statements**

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the IHCUDBD command.

*Table 25. DD statements and record format for the IHCUDBD command*

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**STEPLIB**

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```verbatim
//STEPLIB DD DISP=SHR,DSN=ihcload
//       DD DISP=SHR,DSN=imstools
//       DD DISP=SHR,DSN=reslib
```

where:

- *ihcload* is the name of the library that contains the IMS HALDB Toolkit load modules.
- *imstools* is the optional name of the library that contains the IMS reorganization utilities load modules.
- *reslib* is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

**IMS**

This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

**IMSDALIB**

This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.
If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested. Dynamic allocation of the database data sets is attempted in the following order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

**RECONx**
These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

**Attention:** If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

**IHCSYSIN**
This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see "IHCUDBD command parameters" on page 171.

**OUTPUT**
This statement is required. The statement defines the output data set for storing the rebuilt DBD source.

The output file can be allocated as SYSOUT=* or PDS.

**MSGPRINT**
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see "MSGPRINT data set" on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

**TRACE**
This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

**SYSUDUMP**
This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
**IHCUDBD command parameters**

The following table summarizes the **IHCUDBD** command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the DBD name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Creating a PSB list for selected DBDs

A program specification block (PSB) is a control block that describes databases and logical message destinations used by an application program. A PSB consists of one or more program communication blocks (PCB).

About this task

A program communication block (PCB) is an IMS control block that describes an application program’s interface to and view of an IMS database or, additionally for message processing and batch message processing programs, to the source and destinations of messages. PCBs are defined by the user during PSB generation.

The show PSB utility can be used to create a list of program specification blocks (PSB) that reference a given database, and assists in online changes by identifying which PSBs require change.

The references to both the DBDs and the PSBs reside in the application control block (ACB) library. The ACB library is used during online and database batch (DBB) region type execution of IMS.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEACBL.
2. Specify the JCL DD statements. For a list of DD statements, see “IHCWPSBL DD statements” on page 173.
3. Edit the list of the DBD names on the DBDLIST DD.

Results

The list of PSB names can be found in the MSGPRINT file.

Example

```
//JOBNAME JOB 0,000,CLASS=A,MSGCLASS=X
//*******************************************************
//** SHOW PSBS FROM ACBLIB FOR SPECIFIED DBD(S) **
//*******************************************************
/PSBS EXEC PGM=IHCYUTIL,
// REGION=50M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
// IMS DD DISP=SHR,DSN=your.DBD.LIBRARY
// IMSACB DD DISP=SHR,DSN=your.ACBLIB
//TRACE DD SYSOUT*
//MSGPRINT DD SYSOUT*
//SYSUDUMP DD SYSOUT*
//IHCYSYSIN DD *
//RUN PGM(IHCWPSBL)
//DBDLIST DD *
DBDNAME1 <=>== specify DBD name
DBDNAME2

Figure 26. Sample JCL member IHCEACBL
```
IHCWPSBL DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the IHCWPSBL command.

Table 27. DD statements and record format for the IHCWPSBL command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSACB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>DBDLIST</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR,DSN=ihcload
// DD DISP=SHR,DSN=imstools
// DD DISP=SHR,DSN=reslib
```

where:

- `ihcload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

IMS

This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

IMSACB

This statement is required. The statement defines the ACB library that contains the DBD that the DBDLIST DD statement specifies.

This data set must reside on DASD.
IHCSYSIN
This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

DBDLIST
This statement is required. This statement defines the DBDs. To specify multiple DBDs, write one DBD name on each line.

MSGPRINT
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE
This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

SYSUDUMP
This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
Creating OSAM multi-volume data sets

This utility creates the additional VTOC records on the subsequent volumes using the secondary allocation. Overflow Sequential Access Method (OSAM) is an IMS data management access method that combines selected characteristics of BSAM and BDAM. OSAM is used by the following IMS database access methods: HIDAM and HDAM if VSAM is not used. OSAM is also used by some of the online pool management routines.

About this task

The OSAM multi-volume utility allows you to create OSAM multi-volume data sets using the **ALLOCATE** command in TSO/IDCAMS.

You do not need to create OSAM multi-volume data sets for z/OS Storage Management Subsystem (SMS) managed data sets. This utility is appropriate only for non-SMS data sets.

The **ALLOCATE** command creates the catalog entries for all specified volumes, but only builds a VTOC record on the first volume. IMS OSAM does not allow this and will abend.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEOSAM.
2. Edit the JCL as necessary.

Example

```bash
//JOBNAME   JOB 00,000,CLASS=A,MSGCLASS=X
/*/-----------------------------------------------------------
/*/  OSAM MULTI VOLUME UTILITY
/*/  MAKE SURE THAT ALL VOLUMES HAVE A VTOC
/*/  NOT REQUIRED WHEN SMS
/*/-----------------------------------------------------------
/IMVOL   EXEC PGM=IHCEOSAM
/STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
/MSGPRINT DD SYSOUT=*
/SYSIN DD *
dsname1 <<< specify your DSN
dsname2

Figure 27. Sample JCL member IHCEOSAM
```
Listing VIO and unit names

It is possible for an installation parameter to ask you for non-VIO (virtual I/O) unit names. This list DASD unit names utility shows all unit names that do not reflect device types (such as 3390) and marks those that are defined as VIO.

Procedure
1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEUNIT.
2. Edit the JCL as necessary.

Results
The list of unit names can be found in the MSGPRINT file.

Example

```
//JOBNAME   JOB 0,000,CLASS=A,MSGCLASS=X
//*****************************************************************************
//* LIST DASD UNIT NAMES
//*****************************************************************************
//UNIT EXEC PGM=IHCYUTIL,
//    REGION=50M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
//    DD DISP=SHR,DSN=your.SDFSRESL
//TRACE DD SYSOUT="
//MSGPRINT DD SYSOUT="
//SYSDUMP DD SYSOUT="
//IHCSYSIN DD *
    RUN     PGM(IHCWEDTI)
```

Figure 28. Sample JCL member IHCEUNIT
Part 6. HALDB without DBRC

The HALDB without DBRC feature provides the ability to perform data set name substitutions outside of DBRC. This feature is intended for use in test installations.

You can also convert non-HALDB DBDs and data sets to HALDB DBDs and data sets.

Topic:
- Chapter 12, “HALDB without DBRC,” on page 179
Chapter 12. HALDB without DBRC

You can use this feature to perform HALDB data set name substitutions outside of DBRC. It is intended for use in test installations. The feature can process multiple database data sets with one definition.

You can substitute HALDB data set names by using either of the following methods:
- Substituting data set names by specifying the DD statements in the JCL.
- Modifying the global table definitions by using the ISPF user interface or batch definitions.

You can also convert non-HALDB DBDs and data sets to HALDB DBDs and data sets by using the ISPF user interface.

When you use the HALDB without DBRC utility, the database is treated as if it is not registered to DBRC.

The underlying HALDB structure comes from the RECON. This RECON could be the same RECON that is used in the associated online system or the RECON for that online system in a test environment. The RECON holds the HALDB definitions, such as partition names and numbers, key ranges, and file attributes. The HALDB without DBRC utility does not update the RECON, it only refers to it to obtain the HALDB definitions.

All the functions of IMS HALDB Toolkit, except for the HALDB without DBRC function, use DBRC (DBRC=Y). In a HALDB environment without DBRC, all IMS utilities that rely on the RECON content are not usable. However, the backup and recovery utilities continue to work.

Topics:
- “Configuring HALDB for use without DBRC” on page 180
- “Substituting data set names by using JCL” on page 182
- “Substituting data set names by using ISPF or batch definitions” on page 183
- “Converting non-HALDB DBDs to HALDB DBDs” on page 201
Configuring HALDB for use without DBRC

You can configure HALDB without DBRC to perform data set name substitutions by modeling them after the DD statements that are provided in the JCL or by using the ISPF interface or batch definitions.

Before you begin

In the IMS batch JCL, the DBRC parameter in the parameter string must be set to Y to obtain the partition information from the RECON.

CAUTION:
• DSPCTR0, DFSISVI0, and IHCUTIDD, which are customized modules of IMS HALDB Toolkit, must reside on a different library from the library where other IMS Tools product modules reside.
• You must not concatenate the library that contains DSPCTR0, DFSISVI0, and IHCUTIDD modules to the STEPLIB statement of IMS online (DBRC region) JCL.
• Remember to back up the database as necessary. The HALDB without DBRC utility processes the database as if it is not registered to RECON and therefore no recovery point is created.

About this task

During the configuration, set the JCLALLO parameter in the IHCEDBRI member to indicate whether you want to substitute data set names by modeling them after the DD statements in the JCL or by using the ISPF user interface or batch definitions to modify the global tables.

Procedure

To substitute data set names by using the ISPF user interface or batch definitions, complete the following steps:

1. Add the DBRCNODD=YES | NO parameter to the IHCEDBRI member by choosing one of the following options:
   • To deactivate DBRC for this particular run only, add DBRCNODD=YES and specify the following DD statement: //$$DBRCNO DD DUMMY.
   • To deactivate DBRC entirely, add DBRCNODD=NO. If necessary, you can temporarily disable this bypass by specifying the following statement: //$$DBRCYY DD DUMMY.

2. Add the JCLALLO=YES | NO parameter to the IHCEDBRI member by choosing one of the following options:
   • To create rules for substituting HALDB data set names by using the ISPF or batch definitions, add JCLALLO=NO.
   • To substitute data set names by specifying JCL, add JCLALLO=YES.

3. Run SIHCSAMP member IHCEDBRI. The information KSDS is created, and its data set name is saved in a program.
   CAUTION:
   Make sure you create the IHCUTIDD module in a library that is different from where other IMS Tools product libraries exist.

4. Run SIHCSAMP member IHCEDBRC. IHCEDBRC generates modules DSPCTR0 and DFSISVI0.
CAUTION:
Make sure you create DSPCRTR0 and DFSISVI0 modules in a library that is different from where other IMS Tools product libraries exist.

This step re-links the DBRC module DSPCRTR0 to add the intercept module. The module can reside in the IMS RESLIB, but for testing purposes, add it to the output data set (your_output_dataset in IHCEDBRC) and not to the IMS HALDB Toolkit library.

5. Concatenate the output data sets generated by IHCEDBRI and IHCEDBRC to the STEPLIB DD. These data sets must be concatenated before IMS RESLIB.

6. For ISPF users: Copy the CLIST IHCCNOTY into your CLIST library. The CLIST contains the ISPF invocation. You must adjust the data set names to your names. You can change the CLIST name after you copy it into your CLIST library.
Substituting data set names by using JCL

You can substitute data set names by adding DD statements to the JCL.

Procedure

To substitute data set names by using JCL:

Add a DD statement for each HALDB that is used. The following example shows the valid form for the DD statements that you add.

```
//haldbdn DD DISP=SHR,DSN=your.first.partition.A00001
```

All other partition data sets must use the same dsname prefix, `your.first.partition`. The HALDB master DBD name must be `haldbdn`. If you have a PSINDEX, each index must be specified.

Results

Data set names are modeled after the DD statements that you specified in the JCL.
Substituting data set names by using ISPF or batch definitions

You can create data set naming rules by using the ISPF user interface or batch update.

If you are unfamiliar with JCL or batch scripts, you can use the ISPF interface to gain an understanding of the types of rules you can create.

In this topic:
- “Substituting data set names by using the ISPF user interface” on page 184
- “Substituting data set names by using batch functions” on page 193

Global tables

The HALDB without DBRC feature includes three preinstalled global tables. You can use the ISPF interface or batch update to modify these tables to create rules for substituting HALDB data set names.

These global tables are used when special tables do not exist. Each global table has one row defined by the key "Global." These rows cannot be deleted because they function as a fallback for row selection errors. However, these rows might be the only rows you need, and they can be modified.

The following global tables are preinstalled:

SYSTEM table
Each row in the SYSTEM table defines a certain system. If you need different user definitions per system, the SYSTEM table can have a user table assigned. If the global DBD table cannot be used for certain DBDs, it can have a DBD table assigned to it.

USER table
Each row in the USER table defines a certain user. Use it only if the global user row is not sufficient. If special definitions for that user and that DBD are required, the user row can have a DBD table attached to it.

DBD table
Each row in the DBD table defines a certain DBD. This row is required only if the data set name creation rule is kept at the DBD level. A DBD table can be assigned to a user, a system, or a system and a user.

Data set names

Data set names are created by modeling them after DD statements in the JCL or by using the ISPF user interface or batch update to modify table definitions that are identified by the IMS system (IMSID), the user (USERID), and the DBD.

Data set names in HALDB are specified by using the DSNPREFIX parameter in DBRC, which is limited to 37 characters.

You can use any combination of the following five elements to create the DSNPREFIX parameter:

- **HLQ** The high-level qualifier. It can contain up to 37 characters.
- **SYS** The name of the system. It can contain up to eight characters.
- **USR** The name of the user. It can contain up to eight characters. You can specify
whether to use the content of the USR field, the TSO user ID, or a TSO user ID that is created by using a pattern.

<table>
<thead>
<tr>
<th>DBD</th>
<th>The name of the DBD. It can contain up to eight characters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART</td>
<td>The name of the partition. It can contain up to seven characters.</td>
</tr>
</tbody>
</table>

Here is an example of the DSNPREFIX parameter:

```
Rule=HLQ, SYS, DBD

HLQ=DB.PROD
SYS=IMSA
DBD=DI21PART

DSNPREFIX = DB.PROD.IMSA.DI21PART
```

You conduct data set substitution at the DBD level. The rule for creating data set names is the same for all partitions of the same DBD.

**Substituting data set names by using the ISPF user interface**

You can use the ISPF user interface to create rules for substituting HALDB data set names.

To use the ISPF user interface, begin by opening the IMS HALDB Toolkit panel, as shown in the following figure.

![IMS HALDB Toolkit panel](image)

You can use the ISPF user interface to do the following tasks:

- "Modifying the system definition"
- "Modifying the system selection method" on page 186
- "Maintaining the RECON table" on page 187
- "Maintaining the IMSID table" on page 188
- "Specifying the level where rules are active" on page 189
- "Maintaining SYSTEMs" on page 190
- "Specifying the USER for creating data set names" on page 191
- "Maintaining USERS" on page 192

**Modifying the system definition**

You can modify the system definition to change the IMS libraries and DBD libraries.
Procedure

To modify the system definition, complete the following steps:

1. On the IMS HALDB Toolkit panel, specify option 1, System Setup. The System Setup panel is displayed, as shown in the following figure.

![Figure 30. System Setup panel](image)

2. Specify option 1, System definition. The System definition panel is displayed, as shown in the following figure.

![Figure 31. System definition panel](image)

3. Specify 1, Specify IMS libraries. The Update IMS Libraries panel is displayed, as shown in the following figure. The specified IMS libraries are used as default settings and are required for specific functions.

![Figure 32. Update IMS libraries panel](image)

4. Verify or update the IMS libraries and press Enter. You can specify the following DD names:

**MACLIB**

Specifies the MACLIB data set. Use MACLIB2 to specify an additional macro library.

**RESLIB**

Specifies the RESLIB data set. Use RESLIB2 and RESLIB3 if you need a concatenation to the RESLIB.
MDALIB
Specifies the MDALIB data set. Use MDALIB2 if you have more than one MDALIB. MDALIB might hold the RECON data set that contains the HALDB layout that you will use.

The System definition panel is displayed again.

5. On the System definition panel, specify option 2, Specify DBD libraries. The Update DBD libraries panel is displayed, as shown in the following figure.

6. Verify or update the DBD libraries and press Enter. The DBD libraries contain the non-HALDB and the HALDB DBDs. To convert multiple data sets, both libraries must be available.

Results
The system definition is updated with the IMS libraries and DBD libraries that you specified.

Modifying the system selection method
You can use a system selection method other than the default setting by modifying the system selection method.

About this task
The default setting for the system selection method is IMSID. In this setting, an IMSID selects the system row. Selecting a system row assigns a key for the system table. If the system table does not have a key, the default setting is the global key. After the system row has been selected, all other table selections are made based on that table.

To use a system selection method other than IMSID, complete the following steps:

Procedure
1. On the IMS HALDB Toolkit panel, specify option 1, System Setup. The System Setup panel is displayed, as shown in the following figure.
2. Specify option 2, System selection method. The System Selection Method panel is displayed, as shown in the following figure.

![System Setup panel](image)

**Figure 34. System Setup panel**

3. Specify one of the four following options to locate the system row in the system table. These options are used to define the rules for naming data sets.

- **RECON**
  
  If you select RECON, you must create a table that assigns RECON data set names to a system row by its name.

- **IMSID**
  
  If you select IMSID, you must create a matching table that assigns an IMSID to the name of a system row. If the table does not contain an assignment for the specified IMSID, the IMSID is used to locate the system row.

- **FILE**
  
  If you select FILE, you must provide a DD statement that has a record that contains the name of the system table row.

- **User exit**
  
  If you cannot use one of the other three methods, select User exit. The user exit provides the system name.

**Results**

The system selection method is updated so that the method that you selected will locate the system row and assign a key for the system table.

**What to do next**

You can maintain the RECON table and the IMSID table or specify the level where the rules are active.

**Maintaining the RECON table**

You can display, add, update, or delete a RECON.
Procedure

To display, add, update, or delete a RECON, complete the following steps:

1. On the IMS HALDB Toolkit panel, specify option 1, System Setup. The System Setup panel is displayed, as shown in the following figure.

![Figure 36. System Setup panel](image)

2. Specify option 3, Maintain RECON table. The System Selection by RECON data set name panel is displayed, as shown in the following figure.

![Figure 37. System Selection by RECON data set name panel](image)

3. Specify the appropriate option to display, add, update, or delete a RECON.

**Maintaining the IMSID table**

You can display, add, update, or delete an IMSID.

**Procedure**

To display, add, update, or delete an IMSID, complete the following steps:

1. On the IMS HALDB Toolkit panel, specify option 1, System Setup. The System Setup panel is displayed, as shown in the following figure.

![Figure 38. System Setup panel](image)
2. Specify option 4, Maintain IMSID table. The System Selection by IMSID panel is displayed, as shown in the following figure.

![Figure 39. System Selection by IMSID panel](image)

3. Specify whether you want to display, add, update, or delete an IMSID.

**Specifying the level where rules are active**

You can specify whether the rules for creating the DSNPREFX parameter are active on the SYSTEM level, the USER level, or the DBD level.

**Procedure**

1. On the IMS HALDB Toolkit panel, select option 2, Maintain SYSTEMs. The SYSTEM Definition panel is displayed, as shown in the following figure.

![Figure 40. SYSTEM Definition panel](image)
2. Select option 2, Display SYSTEM. The panel for the SYSTEM Definition of the SYSTEM you selected is displayed, as shown in the following figure.

![System Definition Panel](image)

**Figure 41. SYSTEM Definition of the selected SYSTEM panel**

On this panel, the SYSTEM status indicates whether the SYSTEM you selected and the dname substitution are active.

3. Specify a Y next to the level where you want the rules for creating a data set name to be active. You can specify the SYSTEM level, the USER level, or the DBD level. In the previous figure, the rules are active on the SYSTEM level.

In the previous figure, the rules use the HLQ and the name of the DBD to create the DSNPREFIX parameter.

**Results**

The DSNPREFIX parameter is created based on the rules that are specified and the level where those rules are active.

**Maintaining SYSTEMs**

You can select, add, update, and delete SYSTEMs.

**About this task**

The systems are kept in the SYSTEM table. Each row in the table represents a system. Systems are selected by searching the SYSTEM table for the name of the system. If the name of the system is not found, the global system row, which is represented by "Global," is selected. All other table selections are based on the selected system.

**Procedure**

1. On the IMS HALDB Toolkit panel, select option 2, Maintain SYSTEMs. The SYSTEM Definition panel is displayed, as shown in the following figure. You are automatically connected to the SYSTEM that you previously selected.
2. Specify the appropriate option to select, add, update, or delete a SYSTEM.

**Specifying the USER for creating data set names**
You can specify whether you want to use the content of the USR field, a TSO user ID, or a TSO user ID that is generated by a pattern as the USER.

**Procedure**
1. On the IMS HALDB Toolkit panel, select option 3, Maintain USERS. The USER Definition panel is displayed, as shown in the following figure.

2. Select option 2, Display USER. The panel for the USER Definition of the USER that you selected is displayed, as shown in the following figure.
3. Specify a Y next to the method that you want to use to create the USER field when you build the data set name. You can create the USER field by choosing one of the following methods:

**Use USR field**

The definition of the USR field.

**Use TSO User ID**

The current USER.

**Create from TSO User ID using pattern**

A user ID that is generated by a pattern.

In the previous figure, the USER field is created by using a TSO user ID that is generated by a pattern.

In the previous figure, the rule uses all five elements to create the DSNPREFX parameter.

### Maintaining USERs

You can select, add, update, and delete USERs.

### Procedure

To select, add, update, and delete USERs, complete the following steps:

1. On the IMS HALDB Toolkit panel, select option 3, Maintain USERs. The USER Definition panel is displayed, as shown in the following figure. You are automatically connected to the current user.
2. Specify the appropriate option to select, add, update, or delete a USER.

**Substituting data set names by using batch functions**

You can use batch update to create rules for substituting HALDB data set names.

In batch update, each table is identified by the NAME parameter, and the data comes from an input file that is specified by the FILE parameter. You can do the following tasks, which use the same input file:

- **ADD**
- **DELETE**
- **UPDATE**

The SYSTEM parameter and the USER parameter specify the association of the table rows. An asterisk in parentheses (*) defines a row as global.

The samples contain comment rows to help you align parameters. These rows begin with an asterisk and are followed by a field name. You enter data directly underneath these rows, starting at the asterisk.

The rule for constructing data set names is defined as 12345, where:

1. Identifies the high-level qualifier (HLQ).
2. Identifies the name of the system.
3. Identifies the name of the user.
4. Identifies the name of the DBD.
5. Identifies the name of the partition.

The SIHCSAMP file provides members IHCESAM1 to IHCESAM6. These sample members provide example JCL for specifying rules for creating the DSNPREFIX parameter and adding a row to each table.

<table>
<thead>
<tr>
<th>SIHCSAMP member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHCESAM1</td>
<td>Adds a system to the SYSTEM table. For more information, see &quot;Adding a system to the SYSTEM table&quot; on page 194.</td>
</tr>
<tr>
<td>IHCESAM2</td>
<td>Adds a user to the USER table of SYSTEM IVP1. For more information, see &quot;Adding an IMS ID to the IMSID table of SYSTEM IVP1&quot; on page 196.</td>
</tr>
<tr>
<td>SIHCSAMP member</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IHCESAM3</td>
<td>Adds an IMS ID to the IMSID table of SYSTEM IVP1. For more information, see “Adding a row to the DBD table” on page 197.</td>
</tr>
<tr>
<td>IHCESAM4</td>
<td>Adds a row to the DBD table. For more information, see “Adding a user to the USER table of SYSTEM IVP1” on page 195.</td>
</tr>
<tr>
<td>IHCESAM5</td>
<td>Adds DD names to the DBDLIB table of SYSTEM IVP1. For more information, see “Adding DD names to the DBDLIB table of SYSTEM IVP1” on page 198.</td>
</tr>
<tr>
<td>IHCESAM6</td>
<td>Adds DD names to the IMSLIB table of SYSTEM IVP1. For more information, see “Adding DD names to the IMSLIB table of SYSTEM IVP1” on page 199.</td>
</tr>
</tbody>
</table>

### Adding a system to the SYSTEM table

You can add a system to the SYSTEM table by modifying the member IHCESAM1 in the SIHCSAMP file.

#### Procedure

1. Locate the sample JCL in the SIHCSAMP file, member IHCESAM1.
2. Modify the following parameters:
   - **SYSID**
     
     The ID of the system. In the example, IVP1 is the specified SYSID.
   - **SYSNAM**
     
     The name of the system, which is used to create the data set name. In the example, IMSP is the specified SYSNAM. You can also specify IVP1.
   - **RULE**
     
     The rules that are used to create the DSNPREFIX parameter. In the example, 14 is specified. According to these rules, the DSNPREFIX parameter is created from the HLQ and the name of the DBD.
   - **A**
     
     Indicates whether the system is activated. In the example, Y is specified, indicating that the SYSTEM is activated. You can specify Y or N.
   - **HLQ**
     
     The high-level qualifier that is used to create the DSNPREFIX parameter. In the example, DBSMS.CKIHC is specified as the HLQ.
   - **DESCR**
     
     The description of the system. In the example, test is specified as the DESCR.
Example

```
//S2 EXEC PGM=IHCHALDB,
//      REGION=80M,DYNAMNBR=99
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
//     DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESLIB DD DISP=SHR,DSN=your.SDFSRESL
//IMSDALIB DD DISP=SHR,DSN=your.MDALIB
//IMS DD DISP=SHR,DSN=your.DBDLIB
//TRACE DD SYSOUT**
//SYSPRINT DD SYSOUT**
//SYSUDUMP DD SYSOUT**
//IHCSYSIN DD *
//TABLE NAME(SYSTEM) FILE(DD1) FUNCTION(ADD)
//DD1 DD *
*SYSID *SYSNAM *RULE *HLQ *DESCR
IVP1  IMSP  14  Y DBSMS.CKIHC  test
```

Figure 46. Sample JCL for the SYSTEM table

Adding a user to the USER table of SYSTEM IVP1

You can add a user to the USER table of SYSTEM IVP1 by modifying the member IHCESAM2 in the SIHCSAMP file.

About this task

The rule for creating the user name is also used for creating the data set name, even when the data set name creation level is not on the USER level.

Procedure

1. Locate the sample JCL in the SIHCSAMP file, member IHCESAM2.
2. Modify the following parameters:

   **UID**
   
   The name of the user ID. TEST1 is specified as the UID.

   **RULE**
   
   The rules that are used to create the DSNPREFIX parameter. According to these rules, the DSNPREFIX parameter is created from the HLQ, the system, the user, the DBD, and the partition name.

   **U**
   
   The indicator that determines the content of the USER field when the data set name is created. You can specify one of the following values:

   **R**
   
   Uses the USR field.

   **D**
   
   Uses the UID field.

   **P**
   
   Uses a pattern to create the portion of the user ID. P is specified as the content of the USER field.

   **HLQ**
   
   The high-level qualifier that is used to create the DSNPREFIX parameter. DBSMS.CKIHC is specified as the HLQ.

   **Important:** Do not change this field if it is used at this level for creating the data set name.
**USR**

The definition of the user **TEST** is specified as the USR.

**Important:** Do not change this field if it is used at this level for creating the data set name.

**USER NAME**

The name of the user. **Test User** is specified as the USER NAME.

**Example**

```
//S2 EXEC PGM=IHCHALDB, 
 REGION=80M,DYNAMNBR=99 
/STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD 
/ DD DISP=SHR,DSN=your.SDFSRESL 
/DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL 
/IMSDALIB DD DISP=SHR,DSN=your.MDALIB 
/IMS DD DISP=SHR,DSN=your.DODB LIB 
/TRACE DD SYSOUT** 
/SYSPRINT DD SYSOUT** 
/SYSDUMP DD SYSOUT** 
/IHCSYSIN DD * 
   TABLE NAME(USER) FILE(DD2) FUNCTION(ADD) SYSTEM(IVP1) 
/ DD2 DD * 
 *UID *PATTERN*RULE * *HLQ *USR *USER NAME 
TEST1 **** 12345 P DBSMS.CKIHC TEST Test User 
```

*Figure 47. Sample JCL for the USER table*

**Adding an IMS ID to the IMSID table of SYSTEM IVP1**

You can add an IMS ID to the IMSID table of SYSTEM IVP1 by modifying the member IHCESAM3 in the SIHCSAMP file.

**Procedure**

1. Locate the sample JCL in the SIHCSAMP file, member IHCESAM3.
2. Modify the following parameters:

   **IMSID**
   
   The name of the IMS ID. In the example, **CKCK** is specified as the IMSID.

   **SYSID**
   
   The name of the system ID. In the example, **IVP1** is specified as the SYSID.

   **DESCR**
   
   The description of the IMS ID. In the example, **Same IMS data sets as IVP1** is specified as the DESCR.
Example

```
//S2    EXEC PGM=IHCHALDB,
      //      REGION=80M,DYNAMNBR=99
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
//      DD DISP=SHR,DSN=your.SDFSRESLB
//IMSALIB DD DISP=SHR,DSN=your.MDALIB
//TRACE DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//IHCYSYSIN DD *
//DD3    DD *

TABLE NAME(IMSID) FILE(DD3) FUNCTION(ADD)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSID</td>
<td>SYSID</td>
<td>DESCR</td>
</tr>
<tr>
<td>CKCK</td>
<td>IVP1</td>
<td>Same IMS datasets as IVP1</td>
</tr>
</tbody>
</table>
```

Figure 48. Sample JCL for the IMSID table

Adding a row to the DBD table
You can add a row to the DBD table by modifying the member IHCESAM4 in the SIHCSAMP file.

Procedure
1. Locate the sample JCL in the SIHCSAMP file, member IHCESAM4.
2. Modify the following parameters:

   **DBD**
   The name of the DBD. In the example, PHD02 is specified as the DBD.
   
   **RULE**
   The rules for creating the data set name. In the example, 1234 is specified as the RULE. According to the rule for constructing data set names, the DSNPREFX parameter is created from the HLQ, the system, the user, and the name of the DBD.
   
   **HLQ**
   The name of the high-level qualifier. In the example, DBSMS.CKIHC is specified as the HLQ.
Example

```plaintext
//S2 EXEC PGM=IHCHALDB,
   // REGION=80M,DYNAMBR=99
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
// DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//IMSDALIB DD DISP=SHR,DSN=your.MDALIB
//IMS DD DISP=SHR,DSN=your.DBDLIB
//TRACE DD SYSOUT**
//SYSPRINT DD SYSOUT**
//SYSUDUMP DD SYSOUT**
//IHCSYSIN DD *
   TABLE NAME(DBD) FILE(DBD) FUNCTION(ADD) SYSTEM(*) USER(*)
/ID3 DD *
   +DBD +RULE +HLQ
   PHD02 1234 DBSMS.CKIHC
```

Figure 49. Sample JCL for the DBD table

**Adding DD names to the DBDLIB table of SYSTEM IVP1**

You can add DD names to the DBDLIB table of SYSTEM IVP1 by modifying the member IHCESAM5 in the SIHCSAMP file.

**About this task**

If you want to assign the DBDs to more than one system, repeat the TABLE command with the name of a different system. If all systems have the same DBDLIBs, assign the different SYSTEM name to the global table.

**Procedure**

1. Locate the sample JCL in the SIHCSAMP file, member IHCESAM5.
2. Modify the following parameter:

   **DDNAME**

   This name is fixed. You can specify DDs of the non-HALDB as DBDLIB1 to DBDLIB5. However, you must specify the DD of the HALDB DBDLIB as DBDLIBH. In the example, DBDLIB1 is specified as the DD of the non-HALDB DBDLIB, and DBDLIBH is specified as the DD of the HALDB DBDLIB.
Example

Adding DD names to the IMSLIB table of SYSTEM IVP1
You can add DD names to the IMSLIB table of SYSTEM IVP1 by modifying the member IHCESAM6 in the SIHCSAMP file.

Procedure
1. Locate the sample JCL in the SIHCSAMP file, member IHCESAM6.
2. Modify the following parameters:

   **DDNAME**
   Only the following DD names are allowed:

   **MACLIB**
   Specifies the MACLIB data set. Use MACLIB2 to specify an additional macro library for DBDGEN.

   **RESLIB**
   Specifies the RESLIB data set. Use RESLIB2 and RESLIB3 if you need a concatenation to the RESLIB.

   **MDALIB**
   Specifies the MDALIB data set. Use MDALIB2 to specify more than one MDALIB.

Figure 50. Sample JCL for the DBDLIB table

```sql
//S2 EXEC PGM=IHCHALDB,
// REGION=80M,DYNAMnbr=99
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
// DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESLIB DD DISP=SHR,DSN=your.SDFSRESL
//IMSDALIB DD DISP=SHR,DSN=your.MDALIB
//IMS DD DISP=SHR,DSN=your.DBDLIB
//TRACE DD SYSOUT=
//SYSPRINT DD SYSOUT=
//SYSPRINT DD SYSOUT=
//IHCSYSIN DD *
TABLE NAME(DBDLIB) FILE(DDDD) FUNCTION(ADD) SYSTEM(IVP1)
//DDDD DD *
*DDNAME DSN
DBDLIB1 your.DBDLIB
DBDLIBH your.haldb.DBDLIB
```
Example

//S2 EXEC PGM=IHCHALDB,
// REGION=80M,DYNAMNBR=99
//STEPLIB DD DISP=SHR,DSN=your.SIHLOAD
// DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//IMSDALIB DD DISP=SHR,DSN=your.MDALIB
//IMS DD DISP=SHR,DSN=your.DBDLIB
//TRACE DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//IHCSYSIN DD *  
TABLE NAME(IMSLIB) FILE(DDIM) FUNCTION(ADD) SYSTEM(IVP1)
//DDIM DD *
*DDNAME *DSN
MACLIB your.SDFSMAC
RESLIB your.SDFSRESL
MDALIB your.MDALIB

Figure 51. Sample JCL for the IMSLIB table
Converting non-HALDB DBDs to HALDB DBDs

You can use the ISPF user interface to convert non-HALDB DBDs to HALDB DBDs, and later convert non-HALDB data sets to HALDB data sets.

Step 1. Converting non-HALDB DBDs to HALDB DBDs

You must convert non-HALDB DBDs to HALDB DBDs before you can convert non-HALDB data sets to HALDB data sets.

About this task

To convert non-HALDB DBDs to HALDB DBDs, complete the following steps:

Procedure

1. On the IMS HALDB Toolkit panel, select option 4, Convert DBD. The Update DBD libraries panel is displayed, as shown in the following figure.

   ![Update DBD libraries panel](image1)

   Figure 52. Update DBD libraries panel

2. Verify the DBD libraries or modify them, and press Enter. The Update IMS libraries panel is displayed, as shown in the following figure.

   ![Update IMS libraries panel](image2)

   Figure 53. Update IMS libraries panel

3. Verify the IMS libraries or modify them, and press Enter. The DBD Conversion panel is displayed, as shown in the following figure.

   ![DBD Conversion panel](image3)
4. Specify the name of the HALDB DBD that you want to convert.
5. Specify whether you want to convert from VSAM to OSAM or to a single data set. You can select both types of conversions, neither type of conversion, or just one type of conversion.
6. Press Enter to begin the conversion.

Results

The source of the specified DBD in the non-HALDB library is rebuilt in HALDB format. Then, the source is recompiled and the HALDB DBD is saved in the specified HALDB DBD library.

What to do next

Keep the non-HALDB DBDs and the HALDB DBDs because you will need them to convert non-HALDB data sets to HALDB data sets.

Step 2. Converting non-HALDB data sets to HALDB data sets

To convert non-HALDB data sets to HALDB data sets, you need data set names for the non-HALDB.

Before you begin

Before you convert non-HALDB data sets to HALDB data sets, you must convert non-HALDB DBDs to HALDB DBDs. Also, you must register the definitions of the converted HALDB DBDs to the RECON data sets.

About this task

You can load the database with an application program. However, you also must create the HALDB data sets.

Attention: After you convert non-HALDB data sets to HALDB data sets, do not delete the non-HALDB DBD because you might want to convert the data sets from other users later. However, conduct new testing with the new HALDB DBDLIB.

You do not need JCL for the new data sets because they are dynamically allocated.

Procedure

To convert non-HALDB data sets to HALDB data sets:
1. On the IMS HALDB Toolkit panel, select option 5 and press Enter. The Update IMS libraries panel is displayed, as shown in the following figure.
2. Verify the IMS libraries or modify them, and press Enter. The Update DBD libraries panel is displayed, as shown in the following figure.

3. Verify the DBD libraries or modify them, and press Enter. The Dataset Conversion panel is displayed, as shown in the following figure.

4. Specify the name of the HALDB DBD. The JCL that is required to transform the non-HALDB data sets to HALDB data sets is generated.

   **Important:** For the non-HALDB data sets, you must include the DD statements, including those DD statements for secondary indexes. If data sets are missing, you are prompted to specify the DD statements.
Part 7. Application support

IMS HALDB Toolkit provides functions that enable IMS batch applications to make better use of the HALDB environment.

Topics:
- Chapter 13, “Partition selection API,” on page 207
- Chapter 14, “Testing partition selection exits,” on page 209
- Chapter 15, “Loading logical children (PROCLOPT=L),” on page 215
- Chapter 16, “Creating a DFSHALDB statement,” on page 217
- Chapter 17, “Loading a single partition,” on page 223
- Chapter 18, “Deleting a single partition,” on page 229
- Chapter 19, “Merging two databases into one HALDB,” on page 235
Chapter 13. Partition selection API

The partition selection API provides a callable interface to assign a key to a partition.

This API allows applications to split their input on a partition boundary for parallel processing. The API returns the partition name and the partition number. The API does not need to run in a DL/I environment.

You can use the partition selection API with an application that processes partitions in parallel. When an application wants to process partitions in parallel, there might be a need to split the application’s input on partition boundaries.

However, the application does not know those boundaries, especially when those boundaries can be changed. The partition selection API resolves this problem by receiving the database root segment key and returning the partition name and number.

Invoking the partition selection API

The following definition specifications are required to invoke the partition selection API:

- Define HANDLE as a 4-byte binary field
- Define PARTNAME as an 8-byte character field
- Define PARTNUM as a 2-byte binary field
- Define RETCODE as a 4-byte binary field
- Define RSNCODE as a 4-byte binary field

The application must be linked as AMODE 31.

API calls

The following calls are appropriate for the partition selection API:

Initial (one time) call

CALL IHCUAPI using ("INIT",HANDLE,RETCODE,RSNCODE,dbname)

The API sets HANDLE to be used in subsequent calls.

Selection call

CALL IHCUAPI using
("SEL",HANDLE,RETCODE,RSNCODE,yourkey,PARTNAME,PARTNUM)

The caller provides yourkey.

The API returns PARTNAME and PARTNUM.

Termination (last) call

CALL IHCUAPI using ("TERM",HANDLE,RETCODE,RSNCODE)

RETCODE and RSNCODE are returned for each call.
## Return and reason codes

*Table 28. Return and reason codes*

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>N/A</td>
<td>Function complete</td>
</tr>
<tr>
<td>16</td>
<td>256</td>
<td>Invalid function</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be INIT,SEL,TERM</td>
</tr>
<tr>
<td>257</td>
<td></td>
<td>Invalid DBD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The specified DBD is not a valid DBD</td>
</tr>
<tr>
<td>258</td>
<td></td>
<td>Not a HALDB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The specified DBD is not HALDB</td>
</tr>
<tr>
<td>259</td>
<td></td>
<td>No handle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The HANDLE is not valid</td>
</tr>
<tr>
<td>260</td>
<td></td>
<td>Partition not found</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The key does not belong to any partition</td>
</tr>
<tr>
<td>261</td>
<td></td>
<td>Invalid TCB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TERM call has a different TCB than INIT</td>
</tr>
<tr>
<td>512-599</td>
<td></td>
<td>Internal error</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contact IBM Software Support</td>
</tr>
</tbody>
</table>
Chapter 14. Testing partition selection exits

The test partition selection exit utility allows testing of partition selection exits without requiring the DBD to be previously converted to HALDB.

About this task

Additionally, the distribution results of a partition selection exit can be tested. The DBD can currently be HALDB.

The input for this utility is an HD unload file. This file must have DFSURGU1 as the ddname.

The output of this utility is a distribution list.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEPSEL.
2. Specify the JCL DD statements. For a list of DD statements, see “PSEXIT DD statements” on page 210.
3. Specify the PSEXIT command parameters. For a list of PSEXIT command parameters, see “PSEXIT command parameters” on page 213.

Example

```/*------------------------------------------------------------
/*
/* Test partition selection exit
/*
/*------------------------------------------------------------
//PSEXIT EXEC PGM=IHCHALDB,
  // REGION=50M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
  // DD DISP=SHR,DSN=your.SDFSRESL
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
//IMS DD DISP=SHR,DSN=your.DBDLIB
//RECON1 DD DISP=SHR,DSN=your.RECON1
//RECON2 DD DISP=SHR,DSN=your.RECON2
//RECON3 DD DISP=SHR,DSN=your.RECON3
//DFSURGU1 DD DISP=SHR,DSN=your.unload.file
//MSGPRINT DD SYSPUT
  //SYSUDUMP DD SYSPUT
  //IHCSYSIN DD *
  //PSEXIT DD(db(nnn)) -  /* <= specify DBD
        PARTNM(n) -  /* <= no of partitions
        KEYLEN(xx) -  /* <= for IHCPSELO
        KEYOFF(xx) -  /* <= for IHCPSELO
        PSNAME(IHCPSELO) */ <= partition select
  //KEYS DD *  /* <= partition high keys
  AAAA
  KKKK
  JJJJ

Figure 58. Sample JCL member IHCEPSEL
PSEXIT DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the PSEXIT command.

Table 29. DD statements and record format for the PSEXIT command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>DFSURG1</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>KEYS</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCLIST</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**STEPLIB**

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR,DSN=ihcload
//    DD DISP=SHR,DSN=imstools
//    DD DISP=SHR,DSN=reslib
```

where:

- `ihcload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

**DFSRESLB**

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.
IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.

**IMS**

This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

**IMSDALIB**

This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested.

Dynamic allocation of the database data sets is attempted in the following order:

1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

**RECONx**

These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

**Attention:** If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

**DFSURGU1**

This statement is required. The statement defines the DBD unload file that the DBD parameter specifies.

**IHCSYSIN**

This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “PSEXIT command parameters” on page 213.

**KEYS**

This statement is required for selection exits that require a string for partition selection. If the selection routine does not need keys or strings, use the PARTNUM parameter instead.

The statement defines the KEYS file that contains a list of keys. Use this statement with the KEYLEN and the KEYOFF parameters.

**MSGPRINT**

This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.
IHCLIST

This statement is optional. The statement defines the IMS HALDB Toolkit output data set for reports. For more information about the reports generated in this data set, see “IHCLIST data set” on page 303.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE

This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

SYSUDUMP

This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
**PSEXIT command parameters**

The following table summarizes the **PSEXIT** command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the name of the DBD to be tested. The unload file (DFSURGU1) must be for the specified DBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>KEYLEN</td>
<td>Optional</td>
<td>Specifies the length of the subkey when the IHCPSEL0 selection exit is used. IHCPSEL0 allows you to select a subset of the key. You can use the exit to specify the offset within the key (KEYOFF) and the length of the key portion (KEYLEN).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong> In the following example, the root key is 8 bytes. Position 5 and 6 have the state code. You want to separate the DB by state. You would specify KEYLEN(2) KEYOFF(5) and provide the state codes in the KEYS file. For example: //KEYS DD * AL AZ WY</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(nnn)</em> where <em>nnn</em> is 1 to root keylen -1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>KEYOFF</td>
<td>Optional</td>
<td>Specifies the offset within the root segment key to use when the IHCPSEL0 selection exit is used. IHCPSEL0 allows you to select a subset of the key. You can use the exit to specify the offset within the key (KEYOFF) and the length of the key portion (KEYLEN).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong> In the following example, the root key is 8 bytes. Position 5 and 6 have the state code. You want to separate the DB by state. You would specify KEYLEN(2) KEYOFF(5) and provide the state codes in the KEYS file. For example: //KEYS DD * AL AZ WY</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(nnn)</em> where <em>nnn</em> is 1 to root keylen -1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>PARTNUM</td>
<td>Optional</td>
<td>Specifies the number of partitions required.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This parameter must be specified only for selection routines that do not need keys or strings. Specify the strings for all other selection routines in the KEYS file. The number of keys in the KEYS file defines the number of partitions. For more information about the KEYS file, see &quot;KEYS&quot; on page 211.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(1 - 999)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Requirement</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>PSNAME</td>
<td>Required</td>
<td>Specifies the name of the partition selection exit you have written.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(exit_name)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Chapter 15. Loading logical children (PROCOPT=L)

A logical child is a segment that has a physical parent and a logical parent.

A logical relationship allows for a different view of segments in a database by providing a new set of pointers (LP,LТ,LCH). HALDB only uses the LP equivalent as an EPS pointer. The LP-EPS pointer requires the ILK of the logical parent, which is not known at that time.

HALDB does not allow loading of logical child segments because there is no WF1 work file, and the extended pointer set (EPS) of the logical parent and the paired segment are unavailable at load time.

WF1 is the work file created during reorganization reload or by a database load application.

In addition, load segments in the hierarchy below the logical child segment (variable intersection data) are not allowed.

IMS HALDB Toolkit improves this situation by allowing these load segments with the following restrictions:
- The logical child segment must contain the LPCK in the IO area.
- The application will receive a BB status code.
  The segments (1 child and dependents) are inserted later.
- JCL is structured as follows:
  - Step 1 is the load program. File DFSURWF1 is used to store the delayed segments.
  - Step 2 is prefix update. This step inserts all delayed segments. All WF1 data sets created by step 1 must be included in the DFSURWF3 file.
- You must change your JCL as follows:
  - Use IHCWRC00 instead of DFSRRC00 to accommodate steps 1 and 2
  - Include the SIHCLOAD target load library in the STEPLIB

At the end of the load (prefix update), all databases are labeled IC needed unless they are defined as NONRECOV.
Chapter 16. Creating a DFSHALDB statement

The DFSHALDB create utility provides dynamic management of the DFSHALDB statement by providing a key for the starting partition and a key for the ending partition.

About this task

Batch applications can process a subset of a HALDB. To accomplish this task, the applications must specify a DFSHALDB DD statement and a parameter statement that specifies the program communication block (PCB), the starting partition, and the number of consecutive partitions.

When the partition structure changes, this statement must be modified. The DFSHALDB create utility makes this change obsolete.

When you specify a key range, the utility transforms the key range into a starting and ending partition. Therefore the application does not require a change when partitions are split or consolidated.

This utility runs as an additional step in front of the specified application.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEDFSH.

2. Specify the JCL DD statements. For a list of DD statements, see "IHCUDFSH DD statements" on page 219.

3. Specify the IHCUDFSH command parameters. For a list of IHCUDFSH command parameters, see "IHCUDFSH command parameters" on page 222. You must specify the DBD and the PCB to be used in the RUN command.
Example

```bash
//*------------------------------------------------------------
//*
//* Create a DFSHALDB statement from keys.
//* The file (DFSHALDB) must be used in
//* the subsequent IMS step.
//*
//*------------------------------------------------------------
//S2 EXEC PGM=IHCALDB,
// REGION=80M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
// RECON1 DD DISP=SHR,DSN=your.RECON1
// RECON2 DD DISP=SHR,DSN=your.RECON2
// RECON3 DD DISP=SHR,DSN=your.RECON3
// IMS DD DISP=SHR,DSN=your.DBLIB
// TRACE DD SYSOUT**
// SYSAEND DD SYSOUT**
// MSGPRINT DD SYSOUT**
// IHCYSIN DD *
// RUN PGM(IHCUDFSH) -
//   DBD(IIIIII) -
//   PCB(2) -
//   KEYS DD *
08001000 <<< low key
98008000 <<< high key

//DFSHALDB DD DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,1),
//   DSN=&&DFSHAL

Figure 59. JCL for creating a DFSHALDB statement
```
IHCUDFSH DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the IHCUDFSH command.

Table 31. DD statements and record format for the IHCUDFSH command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>KEYS</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>DFSHALDB</td>
<td>Output</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB
This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```//STEPLIB DD DISP=SHR,DSN=ihclload // DD DISP=SHR,DSN=imstools // DD DISP=SHR,DSN=reslib```

where:

- `ihclload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

IMS
This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

IMSDALIB
This statement is optional. The statement specifies a partitioned data set or
data sets that contain the dynamic allocation members for the database
data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested. Dynamic allocation of the database data sets is attempted in the following order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

**RECONx**
These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

**Attention:** If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

**IHCSYSIN**
This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “IHCUDFSH command parameters” on page 222.

**KEYS**
This statement is required. The statement specifies the low and the high key. The utility refers to the key range and transforms the key range into a starting and ending partition.

**DFSHALDB**
This statement is required. The statement defines a file to be used in a subsequent step.

Specify the following line:
```
//DFSHALDB DD DSN=&&HAL,DISP=(OLD,DELETE)
```

**MSGPRINT**
This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

**TRACE**
This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

**SYSUDUMP**
This statement is required only if a dump is requested by IBM Software
Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
IHCUDFSH command parameters

The following table summarizes the IHCUDFSH command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Defines the DBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>PCB</td>
<td>Required</td>
<td>Defines the PCB number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(pcb_number)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Chapter 17. Loading a single partition

Databases with secondary indexes must insert the secondary indexes in random sequence when loading.

About this task

You can improve performance (elapsed time) when loading partitions that contain secondary indexes by using the load a single partition utility. The utility delays the insertion of secondary indexes until the end of the load process, at which point it sorts the indexes and loads the indexes sequentially.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCELOAD.
2. Specify the JCL DD statements. For a list of DD statements, see "HALDBLOAD DD statements" on page 224.
3. Specify the HALDBLOAD command parameters. For a list of HALDBLOAD command parameters, see "HALDBLOAD command parameters" on page 227.

Example

```cobol
//JOBNAME   JOB   00,000,CLASS=A,MSGCLASS=X
//******************************************************************************************
//*                                                                                      
//* LOAD a single partition                                                                
//*                                                                                      
//******************************************************************************************
//DROP      EXEC PGM=IHCHALDB,                                               
//          REGION=60M       
//STPLIB    DD DISP=SHR,DSN=your.SIHCLOAD   
//          DD DISP=SHR,DSN=your.SDFSRESL  
//RECON1    DD DISP=SHR,DSN=your.RECON1   
//RECON2    DD DISP=SHR,DSN=your.RECON2   
//RECON3    DD DISP=SHR,DSN=your.RECON3   
//DFSRESLB  DD DISP=SHR,DSN=your.SDFSRESL  
//IMS       DD DISP=SHR,DSN=your.DBBLIB   
//MSGPRINT  DD SYSOUT=*                     
//SYSUDUMP  DD SYSOUT=*                     
//IHCYSIN   DD *                           
HALDBLOAD  DBD(dbdbname) PARTITION(partname) - 
           PSB(pdbname) -             
           PCB(1) -                   
           PGM(pgmname)               
/*
```

Figure 60. JCL for delaying the load of a single partition containing secondary indexes
HALDBLOAD DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the HALDBLOAD command.

Table 33. DD statements and record format for the HALDBLOAD command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>PGMLIB</td>
<td>Input</td>
<td>RECFM=U</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLIB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```bash
//STEPLIB DD DISP=SHR,DSN=ihcload
// DD DISP=SHR,DSN=imstools
// DD DISP=SHR,DSN=reslib
```

where:

- `ihcload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLIB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.
IMS  This statement is required. The statement describes the library that contains
the DBDs of the database that you are processing.

This data set must reside on DASD.

IMSDALIB  This statement is optional. The statement specifies a partitioned data set or
data sets that contain the dynamic allocation members for the database
data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is
always referred to by DBRC, and DBRC determines the data sets to be
allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only
when the dynamic allocation for the original data sets is requested.

Dynamic allocation of the database data sets is attempted in the following
order:

1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECON

These statements are optional. The statements define the database recovery
control (DBRC) RECON data sets. RECON data sets dictate which
databases and IMS online systems to use during the processing.

Attention:  If you specify IMSDALIB DD dynamic allocation, do not use
these RECON DD statements. If you used JCL to allocate RECON1 as a
spare data set, you must restart the database.

PGMLIB  This statement is optional. The statement defines the APF-authorized
application program. You can use either STEPLIB or PGMLIB to specify the
application program.

IHCSYSIN  This statement is required. The statement defines a control statement or
data set that contains control statements that specify the functions of IMS
HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see
“HALDBLOAD command parameters” on page 227.

MSGPRINT  This statement is optional. The statement defines the IMS HALDB Toolkit
output data set for progress messages issued during the job. For more
information, see “[MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output
job. IMS HALDB Toolkit dynamically allocates this data set if you do not
provide this DD statement.

TRACE  This statement is optional. The statement defines the output data set that
IMS HALDB Toolkit uses to write diagnostic trace records. This data set
might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output
job. If you do not provide this DD statement, it is dynamically allocated to
SYSOUT by IMS HALDB Toolkit.
SYSUDUMP

This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
HALDBLOAD command parameters

The following table summarizes the HALDBLOAD command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the master DBD name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>PARTITION</td>
<td>Required</td>
<td>Specifies the partition name to be loaded.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(partname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>PCB</td>
<td>Optional</td>
<td>Specifies the PCB. You can specify a PCB name, or a PCB number as specified in the PSB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> *(pcbname</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> If omitted, the default is to use the first PCB for the DBD with PROCOPT=L</td>
</tr>
<tr>
<td>PGM</td>
<td>Required</td>
<td>Specifies the program name to load the partition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(pgmname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>PSB</td>
<td>Required</td>
<td>Specifies the name of the PSB to be used to load the partition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(psbname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
Chapter 18. Deleting a single partition

The drop partition utility provides an alternative approach for correcting a pointer error after a single database partition is deleted.

About this task

When a single partition is deleted, all secondary indexes for the primary database are invalid. The indexes now contain records for non-existent segments. This condition is known as a pointer error. The secondary indexes must be corrected. You can correct the indexes by rebuilding the indexes using an index builder tool (such as IMS Index Builder).

You could alternatively remove those index records that point to the deleted partition. The drop partition utility provides this solution. This process is recoverable and fast and might be more efficient than rebuilding the indexes.

Alternatively, you can create the JCL by using the ISPF user interface.

Procedure

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHEDROP.
2. Specify the JCL DD statements. For a list of DD statements, see "REORGIND DD statements" on page 230.
3. Specify the REORGIND command parameters. For a list of REORGIND command parameters, see “REORGIND command parameters” on page 233.

Example

```//JOBNAME JOB 00,000,CLASS=A,MSGCLASS=X
//*****************************************************************************/
//* Drop partition
//*[*****************************************************************************
//DROP EXEC PGM=IHCHALDB, REGION=60M
//STEPLIB DD DISP=SHR,DSN=your.SIHCLOAD
// DD DISP=SHR,DSN=your.SDFSRESL
// RECON1 DD DISP=SHR,DSN=your.RECON1
// RECON2 DD DISP=SHR,DSN=your.RECON2
// RECON3 DD DISP=SHR,DSN=your.RECON3
// DFSRESLB DD DISP=SHR,DSN=your.SDFSRESL
// IMS DD DISP=SHR,DSN=your.DBDLIB
//MSGPRINT DD SYSOUT**
//SYSUDUMP DD SYSOUT**
//IHCSYSIN DD *
//REORGIND DBD(PHDO2) DROP(PHDO22)*/
```

Figure 61. JCL for deleting a single partition
REORGIND DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the REORGIND command.

Table 35. DD statements and record format for the REORGIND command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Required</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR, DSN=ihcloud
// DD DISP=SHR, DSN=imstools
// DD DISP=SHR, DSN=reslib
```

where:

- `ihcloud` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.
IMS This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

IMSDALIB This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested.

Dynamic allocation of the database data sets is attempted in the following order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECONx These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

Attention: If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

IHCSYSIN This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “REORGIND command parameters” on page 233.

SYSPRINT This statement is optional. The statement defines the statistics output data set and output from other utilities that are executed under the control of IMS HALDB Toolkit.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide it.

MSGPRINT This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.
This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

SYSUDUMP
This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
# REORGIND command parameters

The following table summarizes the `REORGIND` command parameters.

**Table 36. REORGIND command parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD</td>
<td>Required</td>
<td>Specifies the master DBD name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(dbdname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
<tr>
<td>DROP</td>
<td>Required</td>
<td>Specifies the name of the partition to be removed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong> <em>(partname)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong> None.</td>
</tr>
</tbody>
</table>
### Chapter 19. Merging two databases into one HALDB

The merge DBD utility allows you to merge identical HALDBs, or combine user-partitioned databases into one multi-partitioned HALDB.

**About this task**

The following requirements for this utility and task apply:

- Both DBDs must currently be HALDB
- No secondary index is allowed
- No logical relationships are allowed
- Both DBDs must be identical
  
  If compression is used, the compression exit must support both DBDs.
- If you use your own partition selection exit, ensure that it is supported by IMS HALDB Toolkit, or use the supplied exit, IHCPSEL0
  
  The partition string is copied to the target ("to") DBD.
- DBDs must use high keys that reflect their data
  
  IMS HALDB Toolkit must be able to merge the partition definition into one DBD without changing defined high keys.

IMS HALDB Toolkit verifies that key ranges do not conflict. If they conflict, the merge process stops. However, because the merge is performed using temporary RECONs, the database will not have been changed. During post-processing, the DBRC is updated. The source ("from") DBD is deleted, and its partitions become part of the target DBD.

You must perform an image copy of the new partitions while the source DBD is offline, unless the source partitions are to be added to the end of the target DBD. In that case, the source DBD might be kept online.

When user partitioning (multiple identical databases on different key ranges) has been used, the conversion first transforms the databases to HALDB, and then merges them together using this process.

This task is necessary when you require secondary indexing that was not possible when using multiple databases.

**Procedure**

1. Locate the sample JCL for this utility in the SIHCSAMP file, member IHCEMERG.
2. Specify the JCL DD statements. For a list of DD statements, see "HALDBCOPY DD statements" on page 237.
3. Specify the HALDBCOPY command parameters. For a list of HALDBCOPY command parameters, see "HALDBCOPY command parameters" on page 240.

**Example**

The following example shows the JCL for merging user-partitioned HALDBs into one HALDB.
////******************************************************************************
//** Merge two identical HALDBs
//******************************************************************************
//MERGE EXEC PGM=IHCHALDB,
// //  REGION=60M
//STEPLIB DD DISP=SHR,DSN=your.SINCHAL
// //  DD DISP=SHR,DSN=your.SDFSRESL
//RECON1 DD DISP=SHR,DSN=your.RECON1
//RECON2 DD DISP=SHR,DSN=your.RECON2
//RECON3 DD DISP=SHR,DSN=your.RECON3
//DFSRESLB DD DISP=SHR,DSN=your.SDFSRES
//IMS DD DISP=SHR,DSN=your.DBDLIB
//MSGPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//IHCSYSIN DD *
//HALDBCOPY TODBD(todbd) FROMDBD(frdbd)

Figure 62. Sample JCL for merging two DBDs into one DBD
HALDBCOPY DD statements

DD statements are used to identify the source of input and the placement of output information.

The following table shows DD statements and record format for the HALDBCOPY command.

Table 37. DD statements and record format for the HALDBCOPY command

<table>
<thead>
<tr>
<th>DD name</th>
<th>Use</th>
<th>Format</th>
<th>Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>DFSRESLB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IMS</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Required</td>
</tr>
<tr>
<td>IMSDALIB</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>RECONx</td>
<td>Input</td>
<td>Same as IMS</td>
<td>Optional</td>
</tr>
<tr>
<td>IHCSYSIN</td>
<td>Input</td>
<td>LRECL=80</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>MSGPRINT</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>TRACE</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Output</td>
<td>LRECL=133</td>
<td>Optional</td>
</tr>
</tbody>
</table>

STEPLIB

This statement is required. The library data sets pointed to from the STEPLIB DD statement must always be APF-authorized, even if a DFSRESLB DD statement is provided. Authorized library data sets include:

- IMS HALDB Toolkit load library
- IMS utilities that are available for IMS HALDB Toolkit
- IMS RESLIB library
- Any other libraries

The libraries can appear in any order.

```
//STEPLIB DD DISP=SHR,DSN=ihcload
// DD DISP=SHR,DSN=imstools
// DD DISP=SHR,DSN=reslib
```

where:

- `ihcload` is the name of the library that contains the IMS HALDB Toolkit load modules.
- `imstools` is the optional name of the library that contains the IMS reorganization utilities load modules.
- `reslib` is the name of the library that contains the IMS load modules.

If you do not specify an IMSDALIB DD statement in the JCL, you must specify the MDALIB in the STEPLIB for dynamic allocation purposes.

DFSRESLB

This statement is optional. The statement points to an authorized library that contains IMS SVC modules.

IMS HALDB Toolkit dynamically allocates this DD if you do not provide it.
IMS  This statement is required. The statement describes the library that contains the DBDs of the database that you are processing.

This data set must reside on DASD.

IMSDALIB  This statement is optional. The statement specifies a partitioned data set or data sets that contain the dynamic allocation members for the database data sets of non-HALDB and the RECON data sets.

If DBRC is active and the IMSDALIB DD statement is coded, IMSDALIB is always referred to by DBRC, and DBRC determines the data sets to be allocated for each RECON DD.

For the database data sets of non-HALDB, IMSDALIB is referred to only when the dynamic allocation for the original data sets is requested. Dynamic allocation of the database data sets is attempted in the following order:
1. Dynamic allocation members in the IMSDALIB concatenation
2. Dynamic allocation members in JOBLIB or STEPLIB concatenation

RECONx  These statements are optional. The statements define the database recovery control (DBRC) RECON data sets. RECON data sets dictate which databases and IMS online systems to use during the processing.

Attention: If you specify IMSDALIB DD dynamic allocation, do not use these RECON DD statements. If you used JCL to allocate RECON1 as a spare data set, you must restart the database.

IHCSYSIN  This statement is required. The statement defines a control statement or data set that contains control statements that specify the functions of IMS HALDB Toolkit.

For more information about the format of IHCSYSIN DD statements, see “HALDBCOPY command parameters” on page 240.

SYSPRINT  This statement is optional. The statement defines the statistics output data set and output from other utilities that are executed under the control of IMS HALDB Toolkit.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide it.

MSGPRINT  This statement is optional. The statement defines the IMS HALDB Toolkit output data set for progress messages issued during the job. For more information, see “MSGPRINT data set” on page 302.

The data set can reside on DASD, or it can be routed through the output job. IMS HALDB Toolkit dynamically allocates this data set if you do not provide this DD statement.

TRACE  This statement is optional. The statement defines the output data set that IMS HALDB Toolkit uses to write diagnostic trace records. This data set might be required for problem diagnosis.
This data set can reside on DASD, or it can be routed through the output job. If you do not provide this DD statement, it is dynamically allocated to SYSOUT by IMS HALDB Toolkit.

**SYSUDUMP**

This statement is required only if a dump is requested by IBM Software Support. The statement defines a dump data set. If the IMS HALDB Toolkit detects an error and ends with a U0999 abend, this dump is not necessary. However, if any other system or user abend occurs, this data set might be required for problem diagnosis.

This data set can reside on DASD, or it can be routed through the output job.
HALDCOPY command parameters

The following table summarizes the HALDCOPY command parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROMDBD</td>
<td>Required</td>
<td>Specifies the DBD, including all partitions, to be merged into the TODBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td>TODBD</td>
<td>Required</td>
<td>Specifies the DBD to receive all partitions specified in the FROMDBD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Format</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Default value</strong></td>
</tr>
</tbody>
</table>
Part 8. Using the ISPF user interface

You can create the JCL for many of the IMS HALDB Toolkit batch utilities by using the ISPF user interface.

Topics:

• Chapter 20, “Setting up the environment for the ISPF user interface,” on page 243
• Chapter 21, “Processing a project,” on page 259
• Chapter 22, “Converting to HALDB,” on page 277
• Chapter 23, “Maintaining HALDB,” on page 281
• Chapter 24, “Creating JCL statements for database utilities,” on page 287
• Chapter 25, “Other ISPF utilities,” on page 289
• Chapter 26, “Using ISPF DBRC support functions,” on page 291
Chapter 20. Setting up the environment for the ISPF user interface

The JCL for many of the IMS HALDB Toolkit utilities can be created using the ISPF user interface.

Topics:
- “Starting the ISPF user interface” on page 244
- “Creating an IMS environment” on page 245
- “Creating projects” on page 255
- “Deleting an environment” on page 257
Starting the ISPF user interface

You can use the CLIST to start the ISPF user interface.

Before you begin

Before you run IMS HALDB Toolkit as an ISPF application, you must perform a one-time setup task. Ensure that the steps in the topic "Setting up the ISPF interface for IMS HALDB Toolkit" in the IMS Database Utility Solution: Overview and Customization have been completed.

Procedure

Issue the following command to start the ISPF user interface:

ex 'your.CLIST(IHCCHAL)'

The IMS HALDB Toolkit main menu panel is displayed:

```
Command ===>

  Select Function  Current Settings
  Environment:     IMS Version: 14  Tool 1100

0 Setup Utilities
1 Add-Delete-Change Environment
2 Select an Environment
3 Projects in Progress

4 Convert from Full Function to HALDB
5 Convert entire DD library to HALDB
6 Split or Consolidate HALDB Partitions

7 Database Utilities
8 Other Utilities
9 DBRC for HALDB
```

Figure 63. Main menu panel
Creating an IMS environment

You create an IMS environment to keep conversion projects and definitions for different IMS systems in one place. Help is available for each panel.

About this task

An IMS environment consists of library definitions such as RESLIB, DBDLIB, MDALIB and, perhaps, RECON. Projects are connected to an environment.

Environments typically have names such as TEST or PROD. There is no limit to the number of environments you can create. An environment must be completely defined before it can be used by a project.

Databases can belong to multiple IMS systems. However, only one RECON should represent a database.

Parameters and data sets are uniquely defined for each IMS environment, and most parameters can have an ask-me-later setting. This setting allows you to delay a decision until it is needed and applicable.

Procedure

1. Start the ISPF user interface.
   The IMS HALDB Toolkit main menu is displayed.

   Figure 64. Main menu panel

   2. Select option 1 from the Main Menu and press Enter.
      The Environment Maintenance panel is displayed.
3. To add a new environment, select option 2 from the Environment Maintenance panel and press **Enter**.
   The Define a New Environment panel is displayed:

4. Enter an environment name and a description that provides useful information. Press **Enter**.
   The Project Definition Settings panel is displayed.
   You can now define project-related definitions that are specific to the environment.

5. On this panel make the following specifications:
   - Whether to perform partitioning evaluation separately or during the unload process
   - How you would prefer JCL to be generated (as a single JOB for each phase or as one multi-step JOB)
Whether you would like the data sets of the database that is being converted, to be deleted. If the data sets are to be deleted, specify at which stage the deletion occurs.

When you have finished specifying all of the panel information, press **Enter**. The Data sets panel is displayed.

---

**Command ==>**

<table>
<thead>
<tr>
<th>Data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RESLIB</strong> : <em>your.SDFSRESL</em></td>
</tr>
<tr>
<td>:</td>
</tr>
<tr>
<td><strong>MACLIB</strong> : <em>your.SDFSMAC</em></td>
</tr>
<tr>
<td>:</td>
</tr>
<tr>
<td><strong>LOADLIB</strong> : <em>your.SIHCLOAD</em></td>
</tr>
<tr>
<td><strong>DFSVSAM</strong> : <em>your.JCL</em></td>
</tr>
<tr>
<td><strong>Member name</strong>: <em>VSAMP</em></td>
</tr>
<tr>
<td><strong>MDALIB</strong> : <em>your.MDALIB</em></td>
</tr>
<tr>
<td><strong>MDALIB</strong> 1:</td>
</tr>
<tr>
<td><strong>MDALIB</strong> 2:</td>
</tr>
</tbody>
</table>

**RECON MDA located in:** 4 1 **RESLIB**
2 **MDALIB** 1
3 **MDALIB** 2
4 **Specify RECON data sets**

**Figure 68. Data sets panel**

6. From the Data Sets panel, specify the data sets to use when generating the JCL.

   If the RECONs are not located in either the RESLIB or MDALIB data sets, you can specify them explicitly with option 4. Otherwise, the Specify RECON Data set names panel is skipped.

7. If you selected option 4 to specify RECON data sets, the Specify RECON Data Set Names panel is displayed.

---

**Command ==>**

**Specify RECON Data set names:**

**RECON1** *your.RECON1*
**RECON2** *your.RECON2*
**RECON3** *your.RECON3*

**Figure 69. Specify RECON Data Set Names panel**

8. Specify the RECON data set names to use and press **Enter**.

   The DBD Libraries panel is displayed.
9. Specify the DBD load libraries to use and press **Enter**.
The DBD Conversion Rules panel is displayed.

10. Specify the conversion rules to use, and press **Enter**.
The Save Source Statements panel is displayed.
Press the HELP key for explanation of items on the panel.
11. Specify where you would like to save the generated DBD source and the IDCAMS source statements, and press Enter.

The Primary DBD Partitioning Rules panel is displayed.

Figure 72. Save Source Statements panel

12. Specify conversion rules for the primary DBD, and press Enter.

The PSINDEX Definition Rules panel is displayed.

Figure 73. Primary DBD Partitioning Rules panel
13. Specify the conversion rules for secondary index DBDs, and press Enter.
   The Partition Naming Rules panel is displayed.

   The Space Allocation Rules for PHDAM and PHIDAM panel is displayed.
15. Specify space allocation rules for both PHDAM and PHIDAM, and press Enter.

The Space Allocation Rules for PHDAM and PHIDAM panel is displayed.

Figure 76. Space Allocation Rules for PHDAM and PHIDAM panel


The Space Allocation Rules for PSINDEX Data Sets panel is displayed.

Figure 77. Space Allocation Rules for PSINDEX Data Sets panel
17. Specify space allocation rules for HALDB ILDS data sets, and press Enter. 
The JCL Defaults panel is displayed.

18. Specify the JOBCARD JCL defaults and the allocations rules to use during the 
    unload process and press Enter. 
The Utility Setup for UNLOAD panel is displayed.
19. Specify the specific rules to use for the UNLOAD utility and press Enter. The Utility Setup for RELOAD panel is displayed.

20. Specify the specific rules to use for the RELOAD utility and press Enter. The Utility Setup for BACKUP panel is displayed.
21. Specify the specific rules to use for the image copy utility and press Enter.

**Results**

After you have completed all panels, a dialog box displays indicating that the environment was successfully created.

If the environment was successfully created, press Enter and the first panel for creating a project will be displayed.

You can create the project now or later.
Creating projects

After defining an environment, you can create one or more conversion projects which are associated with that environment.

About this task

Note: If you have just added an environment, you should already be on the Convert Full function to HALDB panel.

Procedure

1. Select option 4 from the IMS HALDB Toolkit Main Menu, and press Enter. The Convert from Full Function to HALDB panel is displayed.

   Figure 83. Convert Full function to HALDB panel

   2. Select option 1 (Add-Delete-Change Project), and press Enter. The Project Maintenance for Environment panel is displayed.

   Figure 84. Project Maintenance for Environment panel

   3. To add a new project, select option 2 and press Enter. The Define a New Conversion Project panel is displayed.

Chapter 20. Setting up the environment for the ISPF user interface  255
4. Specify a useful name and description for the project, and press **Enter**.

The Project Tracking and Staging KSDS Allocation panel is displayed.

5. Specify the KSDS allocation rules, and press **Enter**.

An ISPF edit panel is displayed with generated JCL that you can submit to create the required KSDS data set. If necessary, edit the job, then submit it.

6. After the job runs successfully, press the **END** key (PF3).

The next panel indicates that the environment definitions have been copied, and asks you if you want to change any options.

If you are satisfied with the environment options for the environment connected to this project, press **Enter**.

If you want to change any options, you can do by specifying Y and pressing **Enter**. Panels from the environment definition are displayed again.

After the options are changed, a panel is displayed indicating that the project was created and that the environment options have been updated.
Deleting an environment

An IMS environment allows you to keep conversion projects and definitions for different IMS systems in one place. You can also delete existing environments that are no longer needed.

Procedure

1. Select option 1 (Add-Delete-Change Environment) from the IMS HALDB Toolkit Main Menu and press Enter.
   The Environment Maintenance panel is displayed.

   ![Image of Environment Maintenance panel]

   Figure 88. Environment Maintenance panel

2. Select option 1 to see a list of all environments and press Enter.
   The Environment List panel is displayed. The panel shows an example of a list of environments.

   ![Image of Environment List panel]

   Figure 89. Environment List panel

3. You can select the one to delete by entering an s next to PROD environment.
   Press Enter.
   The next panel that is displayed is a verification panel that confirms the deletion.
If $Y$ is specified, a dialog box is displayed which indicates that the environment was deleted.
Otherwise, you are returned to the list of environments and no environment is deleted.

Figure 90. Verification panel for deleting environments

If $Y$ is specified, a dialog box is displayed which indicates that the environment was deleted.
Otherwise, you are returned to the list of environments and no environment is deleted.
Chapter 21. Processing a project

You can process a project to perform maintenance and conversion tasks.

Topics:

- “Selecting a project” on page 260
- “Processing a project using the Project Work panel” on page 262
- “Creating and submitting JCL” on page 265
- “Using SYNC points to halt or continue to the next phase of a project” on page 266
- “Processing a project using a partition selection exit” on page 268
Selecting a project

You can use the IMS HALDB Toolkit main menu to select conversion and maintenance options for projects.

You must select an environment before beginning a conversion so that IMS HALDB Toolkit knows which data sets and RECONs to use. If you do not currently have an environment or you want to switch to a different environment, select option 2 (Add-Delete-Change Environment) from the IMS HALDB Toolkit main menu.

To begin a conversion project from an existing environment, select option 4 (Convert from Full Function to HALDB) from the IMS HALDB Toolkit main menu.

To begin a maintenance project from an existing environment, select option 6 (Split or Consolidate HALDB Partitions) from the IMS HALDB Toolkit main menu.

Alternatively, you can select option 3 (Projects in Progress) to obtain a list of all projects for this environment. From this list, you can select the appropriate project.

Choosing a new or "last active" conversion project

To begin a conversion project from an existing environment, select option 4 (Convert from Full Function to HALDB) from the IMS HALDB Toolkit main menu.

The Convert Full Function to HALDB panel is displayed.

```
------------------------------- IMS HALDB Toolkit -----------------------------
Command ==> Convert Full Function to HALDB

  4  Select Function  Current Settings
      Environment: online
      Project: O-HD02
      IMS Version: 14

  1  Add-Delete Conversion Project
  2  Projects in Progress
  3  Select a Project
  4  Start or Continue with the Current Project
```

Figure 91. Convert Full Function to HALDB panel

Selecting option 4 on this panel automatically connects IMS HALDB Toolkit to the last active project.

Note: If no connection can be made, the message no project selected is displayed. In that case, select option 3 to select a project.

Choosing a new or "last active" maintenance project

To begin a maintenance project from an existing environment, select option 6 (Split or Consolidate HALDB Partitions) from the IMS HALDB Toolkit main menu.

The Split or Consolidate HALDB Partitions panel is displayed.
Selecting option 4 on this panel automatically connects IMS HALDB Toolkit to the last active project.

Note: If no connection can be made, the message no project selected is displayed. In that case, select option 3 to select a project.

Selecting a project from the environment

After you choose a new or "last active" project, the Project List for Environment your environment panel is displayed, which shows all projects in the specified environment.

In the following example, two types of projects are shown: maintenance (M) and conversion (C).

You can now select a conversion or maintenance project.

If the conversion or maintenance project you select has already been selected by a different user ID, you cannot select it.

After you select a project, the Project Work panel is displayed.
Processing a project using the Project Work panel

After you select a project to convert, the Project Work panel is displayed.

About this task

You can select any of the following options in any order that is necessary to help you complete your work.

Procedure

- If you select option 0 (Setup Parameter) from the Project Work panel, the Select Function panel is displayed. Select the Setup Parameter to specify any setup changes. However, some changes can only be performed at the start of a project.

You can use the Select Function panel to change setup parameters that were made during project definition or to include parameters that might have been omitted. Depending on which step the conversion process is performing, some changes might have no effect.

You should ensure that you complete all changes before selecting DBDs to be converted.
• Select the Continue with Current Project to continue with a project that was
interrupted. If the previous step was completed, conversion begins with the next
step. If this is the first time you are selecting option 1, IMS HALDB Toolkit
proceeds to the DBD selection phase.

• If you select option 2 (Show Status of Project) from the Project Work panel, the
DBDs in Project HDO2 panel is displayed. Select Show Status of Project to view
the step which is currently being processed. Use the Help key for explanations
of fields on the panel.

This panel is informational, only, and lists all DBDs that have been specified for
conversion.

In the example, suppose you selected HDO2 and HPV20. The secondary indexes
(indicated by X) and the primary index (indicated by P) are added,
automatically.

If the selected databases had logically related databases, they would also be
added automatically, and the process would continue until all related databases
were added.

If you had a complex relationship consisting of, for example, five primary DBDs
and twenty secondary indexes, all would be included by selecting only one.

```
Command ==> DBD(s) in Project HDO2
S DBD Type  Prim DBD  Log. Current Level  Current Status
-------------------------------------------------------------------------------
 HD02  D     Collect DBDs  Wait for User
 S1021 X  HD02  Collect DBDs  Wait for User
 S1023 X  HD02  Collect DBDs  Wait for User
 S1022 X  HD02  Collect DBDs  Wait for User
 S1024 X  HD02  Collect DBDs  Wait for User
 HDV1  D     Collect DBDs  Wait for User
 HPV20 P  HDV1  Collect DBDs  Wait for User
```

Figure 96. DBDs in Project HDO2 panel

• Select Recreate JCL of Current Phase to re-create JCL for all processes currently
in SUBMITTED status. This might be necessary if, for example, a batch function
did not complete and you are no longer in the panel to submit the JCL. After
re-creating the JCL, you might need to resubmit it.

• If you select option 4 (Restart at Current or Prior Level) from the Project Work
panel, a list of project phases is displayed in chronological order. Select Restart
at Current or Prior Level to restart a conversion project from the current or prior
level. This would typically be necessary if the JOB did not complete, and the JCL
is no longer available. This panel shows all steps within a current conversion
phase that are finished or in progress. You can go back to a previous step if the
step has not completed. After you have started the “reload” phase, it might not
be possible to go back. You cannot proceed to a future step. When you restart at
a prior level, the project restarts at the beginning of that step.

In the following example, the last project syncpoint is displayed at the top of the
panel; the current phase is displayed at the bottom.
If you select option 5 (Restart from Beginning) from the Project Work panel, a delete confirmation panel is displayed. Select Restart from Beginning to restart a conversion project from the beginning. All data currently held in the project file is deleted. Any changes made during the process up to this point, are not backed out.

Ask me later
If, when setting up your parameters, you specified Ask me later, the panels that correspond begin to display.

About this task
At this stage of the conversion process, other special panels might also display.
Creating and submitting JCL

When all questions have been answered regarding how to process a conversion project, IMS HALDB Toolkit creates JCL and presents an ISPF panel for you to submit it.

About this task

Figure 99. ISPF panel for submitting conversion JCL

When the JCL processes successfully, an indicator displays to confirm that the batch job has completed.

A phase can have more than one batch job; all the jobs must be completed before the next phase begins.
Using SYNC points to halt or continue to the next phase of a project

After certain phases complete, a sync point is reached. The sync point allows you to interrupt the project at a logical point.

About this task

Sync points specify whether to continue to the next step.

A typical reason to halt a project is when batch processing has not completed and you will be unavailable for a period of time.

Procedure

To halt or continue to the next phase from a project checkpoint:

1. At a project checkpoint, the Project Checkpoint panel displays and you can choose to start the next phase or halt the project.

   ![Figure 100. Project Checkpoint panel](image)
   - Specify Y to begin the next phase of the project.
   - Specify N to halt the project.

To halt or resume project conversion:

2. If you halt the project, a Project Halted panel is displayed.

   ![Figure 101. Project Halted panel](image)
   - The status panel automatically causes the Project Work panel to display.
   - Optional: To quit the project and continue later, by pressing the End key.
• Optional: To resume project conversion, select the appropriate option from the Project Work panel.

![Command panel](image)

**Figure 102. Project Work panel**

• Optional: To resume project conversion, select the appropriate option from the Project Work panel.

• Optional: To resume project conversion, select option 1 from the Project Work panel. The project continues from where it stopped.
Processing a project using a partition selection exit

You must use a partition selection exit to process the project, if the database records (root segments) are to be distributed by methods other than the root segment key. If the database records are to be distributed by a portion of the root key, which is not the leftmost portion, you must use a selection exit.

Before you begin

For example, if the regional identifier is in a lower portion of the key, and the data should be separated by region identifier.

Restriction on partition selection exits:

- Caution should be used if you want to specify a partition selection exit for the secondary index. Using the exit can result in index keys that are no longer consecutive from one index partition to the next, thus impacting applications that use PCBs with PROCSEQ.
- IMS HALDB Toolkit does not support a selection exit on PSINDEX.
- If the data is to be distributed using the leftmost portions of the root segment high key, no selection exit is required, and none should be provided.

About this task

Selection exits are called for initialization and sequencing.

Exits provide the following functions:

- IMS knows the number of partitions and provides the partition number as input to the exit.
- The exit has additional parameters including the keystring parameter from the partition definition. This parameter can be used to identify and sequence partitions.

Other methods are possible; however, they are prone to errors when the exit identifies a certain number of partitions, but does not receive that input from the DBRC definitions.

Procedure

1. Activate the partition selection exit for use.
2. Define partition boundaries.
3. Select or change the DBD.
4. Optional: Define other conversion options.

Results

If you use the IHPCSEL0 exit, you can activate the exit and define the partition boundaries as part of one task. If you supply your own partition selection exit, you must perform extra steps to make sure that your conversion process runs smoothly.

IHCPSEL0 exit

The IHCPSEL0 exit is provided with IMS HALDB Toolkit and works the same way as the high key works for partitioning without a selection exit.
IHCPSEL0 uses the DBRC keystrng parameter to identify the key portion and the high subkey for that partition. This enables flexibility for changing the partitioning method. The keystrng parameter is specified as 000;lll;kkkkk where:

- **ooo** is the numeric specifying the offset from the start of the root key. The smallest offset is 001.
- **lll** is the numeric specifying the length of the subkey. \( lll+ooo \) must not be larger than the root segment key.
- **kkkkk** is the high-level qualifier key of the subkey for this partition.

IHCPSEL0 performs certain checks during initialization. It verifies that all \( ooo \) and \( lll \) are the same for each partition. Then, it checks for duplicate high keys.

Keys can be defined shorter than \( lll \). In that case, they are padded with x"FF".

### Using a partition selection exit for conversion

After you create a project and before you select a DBD, you can activate a partition selection exit. You can also use this procedure to define the partition boundaries.

#### About this task

On the next panel (Specify a Partition Selection Exit), you are asked to specify the partition selection exit.

#### Procedure

1. Select option 1 under "Selection Exit" on the DBD Conversion Rules panel.
   
   You can also specify a partition selection exit for the secondary index; however, this might cause problems. The index keys will no longer be consecutive from one index partition to the next. This has an impact on applications that use PCBs with PROCSEQ. IMS HALDB Toolkit does not support a selection exit on PSINDEX.

   1. On the next panel (Partition Selection Exit), specify the partition selection exit.
You can use either IHCPSEL0, which is supplied, or provide your own selection exit.
If you use your own exit, you must provide strings for the partition or define the number of partitions.

---

Command ==> 

Partition Selection Exit for HIDV2

Specify a Partition Selection Module: IHCPSEL0
Specify a Key Offset from Start of the Key: 2
Specify a String Length: 4
Root Segment Key Length is: 8
(Offset length must not exceed root key)

1 Specify Strings
   1 Yes, prompt for strings
   2 No, number of partitions is: 0

Use the HELP key for more information

Figure 104. Partition Selection Exit panel

3. If you use IHCPSEL0, provide the subset high keys.
   IMS HALDB Toolkit provides the Specify Partition Keystrings panel and the Partition Keystrings (character) panel for you to enter subset high key information.

---

Command ==> 

Specify Partition Keystrings

1 Select
   1 Enter keystrings in character format
   2 Enter keystrings in hex format
   3 Provide dataset

Member (if dataset is partitioned)

Figure 105. Specify Partition Keystrings panel

---

Command ==> 

Partition Keystrings for HIDV2
Character Keystrings only

Max key length:  4
No. of keys   :  0

Specify Key:  499
Specify 'FF' for high value key

Figure 106. Partition Keystrings (character) panel
4. When you are finished entering keys, press the End key. The Keystrings for xxxxxx panel displays all of the keys that you entered.

5. Optional: Use the Keystrings for xxxxxx panel to change any of the information that you entered.

   ![Keystrings for DBD HIDV2](image)

   **Figure 107. Verifying partition keystings**

   IMS HALDB Toolkit performs a final check for the partition selection exit.

6. Optional: If you encounter the Partition Selection Exit xxxxxxxx is not currently in your RESLIB panel, copy the exit into your RESLIB. Partition boundary calculations are then performed.

   ![Partition Selection Exit](image)

   **Figure 108. Confirming existence of partition selection exit**

### Changing the DBDs

After DBDs and partition boundaries are selected, you can begin conversion by unloading the primary DBDs.

### About this task

New DBDs and partitions must be defined, and new data sets must be allocated. Then, reloads can be performed and conversion will be complete.
Because the DBDs have changed, an online change must be performed before making the new HALDBs available. The conversion process makes all changes except online changes, although it creates required JCL and control statements.

You submit the JCL as specified in your setup options or during project definition. You can change the options according to the needs of your project.

**Creating partition boundaries**

After DBDs are selected, you should locate effective partition boundaries. This section describes ways to create partition boundaries.

**About this task**

You can accomplish this by locating the high-level key of each partition that you want. You must select partition boundaries for all participating databases, including secondary indexes.

The index record size increases significantly because of the increase in the pointer size from 4 to 28 and the addition of the root segment key.

**Procedure**

- Use a partition selection exit.
- Specify the high key for a partition.
- Set partition boundaries automatically.

**Defining the number of partitions by specifying high keys**

After DBDs are selected, you can define the number of database partitions you want to have by specifying high keys for the partitions.

**About this task**

The number of partitions will be the number of high keys you specify. You must select partition boundaries for all participating databases, including secondary indexes. You need not specify the keys in sequence. The highest key defined is replaced as x'FF'.

The index record size increases significantly because of the increase in the pointer size from 4 to 28 and the addition of the root segment key.

**Procedure**

1. On the Primary DBD Partitioning Rules panel, select option 4 Specify High Keys.
2. To specify keys, select option 1 or 2 depending on whether you want character or hex format.

3. Optional: Select option 1 and the Specify Partition Keystrings Character Keystrings only panel is displayed. Enter one key at a time; or, if you have a file containing the keys, you can use it here. The file must have as many records as the number of keys you want, the keys must be the first position of the record, and the file must have fixed record length.
4. When you have entered all keys, press the **End** key. The Keystrings for *xxxxx* panel displays all keys that were entered.

5. To change this list, select option D, C, or I.

6. Press the **End** key to begin processing, when the list is final.

7. Verify partitioning results. You can view results using the output list from the data collector.

---

**Setting partition boundaries automatically**

There are two methods to set partition boundaries automatically: specifying the partition size (recommended) and specifying the number of partitions.

**Procedure**

1. To specify partition size, select option 2 in the 'Partitions' section of the Primary DBD Partitioning Rules panel.

   This method ensures that no partition will be larger than the maximum size allowed.
2. Optional: To specify the number of partitions, select option 1 in the "Partition" section of the Primary DBD Partitioning Rules panel. If you use this method, a resulting partition might be larger than the 4 GB maximum. In that case, IMS HALDB Toolkit resets the size to 2 GB.

3. Optional: To specify partition boundaries for a secondary index, select option 1 or 2 in the PSINDEX Definition Rules panel.

**Specifying additional conversion options**

You can specify additional options during the conversion process.

**About this task**

For example:

- Change the DBD name.
Changing the name impacts post-processing. You must define the new DBDs to your online system, change all PSBs, and ensure that the applications will be changed if they are aware of the DBD name.

- **Convert from VSAM to OSAM.**
  OSAM has advantages not available in VSAM, especially when you use OSAM sequential buffering (OSB).

- **Convert to a single data set group.**
  Some databases contain multiple data set groups to minimize problems with database size.
  When you use IMS HALDB Toolkit, this adjustment is unnecessary. You need only one data set group.
Chapter 22. Converting to HALDB

If the database is offline, you can use the ISPF user interface to convert the database to a HALDB.

The following sections explain how to accomplish this task.

Topics:
- “Considerations for converting to HALDB” on page 278
- “Converting to HALDB using the ISPF user interface” on page 279
Considerations for converting to HALDB

The following considerations apply when you convert to a HALDB by using the ISPF user interface.

When databases are offline, conversion options are available by using ISPF panels or by using the batch conversion method.

Converting full-function databases requires that IMS HALDB Toolkit evaluate partition sizes and boundaries. Partitioned databases which use the PDB/PDF product are considered full-function databases and can be converted as such.

The conversion process for PDBs/PDFs uses partition high keys (for HIDAM/HISAM to PHIDAM) or the number of partitions (HDAM to PHDAM).

You can also choose to select the partition boundaries again. In this case, the partition definition from the PDB is ignored and the database is treated in the same way as any full-function database.
Converting to HALDB using the ISPF user interface

You can use the ISPF user interface to convert your full-function databases to HALDB.

Procedure
1. To display the DBD Conversion Selection panel, select option 1 (Continue with Current Project) from the Project Work panel. If you did not select DBDs, or if you cleared the project data set, this step is necessary.
2. Select one of the provided DBD libraries, or enter a specific DBD library and press Enter to see a list of DBDs.
   All DBDs and related databases must be converted together. You must select, at minimum, the primary DBD.
   All related DBDs and indexes will be automatically included. In addition, you can add DBDs that are not related (from an IMS perspective), and they will be converted at the same time.

3. Select the primary DBD to be converted; related DBDs are automatically added to the list.

Figure 115. DBD Conversion Selection panel
4. Press the **End** key to display the status panel showing all selected DBDs.
5. View your selections, and press the **End** key to display the following DBD Selection for Project xxxxx panel, which allows you to confirm that the DBD list is complete.

**Figure 116. Selection panel for DBDs to be converted**

6. Continue processing the project as described in Chapter 21, “Processing a project,” on page 259.
Chapter 23. Maintaining HALDB

The maintenance function from the ISPF user interface helps you maintain HALDBs after you convert them from full-function. The maintenance function is used to split or consolidate partitions.

You can use ISPF to perform maintenance tasks after databases are converted. The databases must be offline before beginning any maintenance task. The databases must be taken offline manually.

To maintain a database online, you must use the batch process.

You can perform the following tasks using the ISPF user interface:
- Select DBDs for maintenance
- Select and maintain partitions
- Create and run JCL

The following sections explain how to accomplish these tasks.

Topics:
- “Splitting or consolidating HALDB partitions” on page 282
- “Selecting DBDs for maintenance” on page 284
- “Selecting and maintaining partitions” on page 285
Splitting or consolidating HALDB partitions

You access the maintenance panel by selecting option 6 from the IMS HALDB Toolkit main menu.

About this task

The Split or Consolidate HALDB Partitions panel is displayed.

The panel shows the current environment in which you are working and the IMS version for which it is defined. If you are connected to a project which is eligible for maintenance, the project name is also displayed.

---

IMS HALDB Toolkit

Command ==> Split or Consolidate HALDB Partitions

Select Function

Current Settings
Environment:
   Project:
   IMS Version: 14

1  Add-Delete HALDB Maintenance Project
2  Projects in Progress
3  Select a Project
4  Start or Continue with the Current Project

Figure 118. Split or Consolidate HALDB Partitions

The panel offers the following functions:

• Select option 1 to maintain projects that are defined within the specified environment.
• Select option 2 to display all projects that are defined within the environment.
• Select option 3 to select a project. If no project is preselected, you must select a project before proceeding.
  A list of projects is displayed; however, only those defined as maintenance projects (Type: M) are eligible for selection.
  After you select a project, the name is displayed on the project line.
• Select option 4 to connect to the project that is displayed under Current Settings and to display the Project Work panel.
  In ISPF mode, Help is available for using the fields on this panel.
<table>
<thead>
<tr>
<th>Command ====&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Work Panel</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>0  Select Function</td>
</tr>
<tr>
<td>0  Setup Parameter</td>
</tr>
<tr>
<td>1  Continue with Current Project</td>
</tr>
<tr>
<td>2  Show Status of Project</td>
</tr>
<tr>
<td>3  Recreate JCL of Current Phase</td>
</tr>
<tr>
<td>4  Restart at Current or Prior Level</td>
</tr>
<tr>
<td>5  Restart From Beginning</td>
</tr>
<tr>
<td>Project Desc: Consolidate partitions</td>
</tr>
<tr>
<td>Project KSDS: your.project.consoli.phidv2</td>
</tr>
</tbody>
</table>

**Figure 119. Project Work panel**
Selecting DBDs for maintenance

If you select option 1 on the Split or Consolidate HALDB Partitions panel, you are asked to select the DBD.

About this task

DBDs are extracted from the RECON.

The HALDB DBDs panel is displayed.

Select the DBD to be processed. Only one DBD can be selected at-a-time.

```
------------------------------- IMS HALDB Toolkit -----------------------------
Command ==>                   
HALDB DBDs                  
Sel   DBD   Organization       
-----------------------------------
PHDV1 PHIDAM,VSAM            
PHIDV2 PHIDAM,VSAM           
PPAIR2 PHIDAM,OSAM           
PSHV21 PSINDEX               
PSHV22 PSINDEX               
QHV21 PSINDEX                
QHV22 PSINDEX                
QIDV2 PHIDAM,VSAM            
YHV21 PSINDEX                
YHV22 PSINDEX                
YIDV2 PHIDAM,VSAM            
ZEY120 PHIDAM,OSAM           

*****************************************************************************

Figure 120. Selecting HALDB DBDs

The panel is displayed for selecting the DBD library which contains the selected DBD, as shown in the following example.

Select the appropriate DBD library or type the data set name. The DBD library you choose will be associated with the project.

```
------------------------------- IMS HALDB Toolkit -----------------------------
Command ==>                   
Select one of the provided DBD libraries or specify a DBD library for HALDB DBD PHDV1
  your.DBDLIB
  1
  2
  3
  4
  5
  6
  7
  8
  9

Specify a DBDLIB dataset name

Figure 121. Selecting HALDB DBD library

The next step is to select a partition.
Selecting and maintaining partitions

You use the Select Partitions for Maintenance panel to select and maintain partitions. All partitions are displayed in key sequence.

About this task

The current size of the partition and the space used are displayed to assist you in deciding which partitions to select.

You can select multiple partitions. However, if you select more than one, they must be in key sequence and be selected or cleared one-at-a-time.

IMS HALDB Toolkit considers multiple partitions as one entity and creates new partitions based on the goals that you defined in the project setup.

---

```
Command ==> Select Partitions for Maintenance Master DBD: PHDV1
"S" to select a partition
"D" to deselect a partition

<table>
<thead>
<tr>
<th>Sel</th>
<th>Partition</th>
<th>Curr Space (CYL)</th>
<th>Allocated</th>
<th>Used</th>
<th>High key</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PHDV13</td>
<td>2</td>
<td>2</td>
<td>03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PHDV14</td>
<td>2</td>
<td>2</td>
<td>05</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PHDV12</td>
<td>2</td>
<td>2</td>
<td>06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PHDV10</td>
<td>2</td>
<td>2</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PHDV11</td>
<td>2</td>
<td>2</td>
<td>x’FF’</td>
<td></td>
</tr>
</tbody>
</table>
```

*Figure 122. Displaying all partitions in key sequence*

Restriction on partition selection exits

Only partition selection exit IHCPSEL0 (provided with this product) is supported.

DBDs with other partition selection exits are not supported.
Chapter 24. Creating JCL statements for database utilities

IMS HALDB Toolkit can create JCL for database utilities. You can then run reorganization or backup and recovery utilities for HALDBs or selected partitions.

Procedure

1. From the IMS HALDB Toolkit main menu, select option 7 (Database Utilities) and press Enter.
   The Database Utilities panel is displayed.

   ![Database utilities main menu](image)

   This panel allows you to select utility functions for performing actions on selected databases or partitions.

   **Option 1**
   This function reads the PSINDEX data sets and determines whether the EPS pointers are valid. If necessary, ILDS data sets from the primary database partitions are used to update the index pointer. This function requires the primary database to be offline.

   **Options 2 - 7**
   These functions create JCL to perform the action of the selected utility. Either IMS Tools products or standard utilities are used. On the main menu, you can select option 0 (Setup Utilities) to specify which utilities and associated parameters IMS HALDB Toolkit should use.

   **Option 8**
   If your CI and CA splits are a high number, this function reorganizes the PSINDEX. This function requires the PSINDEX database to be offline.

   **Option 9**
   This function is used to remove a partition. If the database has secondary indexes, the function also removes corresponding index records. This method is faster than rebuilding indexes for remaining partitions.
It is still possible to undo the drop at this time. Because a dropped partition has only been "disabled", it can be reactivated. However, secondary indexes must be rebuilt. If the database has been updated, you must unload and reload to reactivate the dropped partition.

Option A
This function allows you to load a single partition (PCB with PROCOPTL). Using this function can cause performance issues if secondary indexes are inserted. The insertion of secondary indexes is delayed until the partition has been loaded. The index records are then inserted sequentially. The application does not see status codes related to insertion of the records. However, if index loading fails, the job ends.

Option B
This function enables user-partitioned databases to transform to a single HALDB. The following restrictions apply:
• DBDs must be identical.
• The key range of the partitions must not overlap.

This function imports selected partitions into the target master DBD. Data set names are changed according to the rules in the target database. This function checks the key consistency of neighboring partitions. The merge is performed only if all checks have been satisfied.

2. Select a function from the menu and press Enter.
   A panel displays that shows all HALDBs that are currently defined.

3. Select a DBD on which to perform a function and, if necessary, its partition. Press Enter.
   The JCL for that function is generated automatically.
Chapter 25. Other ISPF utilities

You can use Other Utilities functions provided by the ISPF user interface to perform other HALDB tasks.

From the IMS HALDB Toolkit main menu, select option 8 (Other Utilities) and press Enter.

The Utilities panel is displayed.

```
------------------------------- IMS HALDB Toolkit -------------------------------
Command ==>

Utilities
    Select Function       Current Settings
    1 Show DBD Source     Environment: ims14
    2 Reassemble DBD
    3 Partition Selection Exit Test Utility
    4 OSAM Multi volume
```

*Figure 124. Utilities panel*

Option 1 allows you to rebuild the source of your DBD. You are asked to specify the DBDLib and select a DBD. The DBD source will be displayed.

Option 2 is the same as option 1, but applies JCL to the DBD source to enable you to do a DBDGEN.

Option 3 generates the JCL for testing a partition selection exit.

Option 4 creates the JCL to handle a non-SMS OSAM Multi volume.
Chapter 26. Using ISPF DBRC support functions

You can use DBRC support functions provided by the ISPF user interface to show, clone, copy, and backup HALDB definitions and partitions.

About this task

To use DBRC utilities, select option 9 (DBRC for HALDB) from the IMS HALDB Toolkit main menu and press Enter.

The DBRC Support panel is displayed to allow you to select a function that is supported by DBRC.

Figure 125. DBRC Support panel

Topics:
- “Showing partition definitions” on page 292
- “Cloning DBRC definitions” on page 293
- “Copying DBRC partitions” on page 295
- “Backing up DBRC DBD definitions” on page 297
Showing partition definitions

You can use the DBRC Support panel to show partition definitions.

Procedure

1. From the IMS HALDB Toolkit main menu, select option 9 (DBRC for HALDB) and press Enter.
   The DBRC Support panel is displayed.
2. From the DBRC Support panel, select option 1 (Show Partition Definitions) and press Enter.
   The Partition Definition in DBRC panel is displayed. The panel shows a list of all HALDB master DBDs and their partition definitions.

3. To see details about a DBD and its partitions, select a row and press Enter.
   For following example shows the result for option 1 (Show all HALDB DBDs and Partitions):

--- IMS HALDB Toolkit ---
Command =>
Partition Definition in DBRC
   Displayed as currently defined in RECON data sets
   Select
     1 Show all HALDB DBDs and Partitions
     2 Show DBD
     3 Show Partition

   RECON1: your.RECON1
   RECON2: your.RECON2
   RECON3: your.RECON3

Figure 126. Partition Definition in DBRC panel

--- IMS HALDB Toolkit --- Row 1 of 7
Command =>
HALDB DBDs and Partitions
Sel DBD Part Organization Sel.Exit
PHD02A PHD02A0 PHDAM,OSAM
   PHD02A1
   PHD02A2
   PHD02A3
   PHD02B0
   PHD02B1

******************************* Bottom of data *******************************

Figure 127. HALDB DBDs and Partitions panel
Cloning DBRC definitions

You can use the clone function from the DBRC Support panel to replicate HALDB DBRC definitions to other RECONs.

Procedure

1. From the IMS HALDB Toolkit main menu, select option 9 (DBRC for HALDB) and press Enter.
   The DBRC Support panel is displayed.

2. From the DBRC Support panel, select option 5 (Clone Partition Definition) and press Enter.
   The HALDB DBDs panel is displayed showing the list of HALDB DBDs.

3. Enter s to make a partition selection, along with any secondary indexes you want to include, and press Enter.
   The Enter Target RECON Data Set Names panel is displayed.

4. Specify the RECON data sets to which you plan to clone the HALDB DBRC definitions and press Enter.
   The Options for DBRC Cloning of DBD panel is displayed:
5. Specify the following parameters:
   - Specify a new data set name prefix.
   - If you include the secondary index databases, use the DBD name as an additional data set qualifier

IMS HALDB Toolkit creates the JCL to transport partition definitions of the selected DBD and any secondary indexes.
Submit the JCL to create the partition definition in your alternate RECONs.

**Note:** You should not use this function to transport image copies to partitions that have just been defined. Use the Copy function, instead. However, you can use any Load process to transport data to replicated partitions.
Copying DBRC partitions

You can use the copy function from the DBRC Support panel to replicate HALDB DBRC definitions to other RECONs.

About this task

Use this function to copy DBRC definitions and partition data sets to a different RECON.

The data transport process can unload or reload. However, this function is designed to allow transport with image copies.

Procedure

1. From the IMS HALDB Toolkit main menu, select option 9 (DBRC for HALDB) and press Enter.
   The DBRC Support panel is displayed.
2. From the DBRC Support panel, select option 6 (Copy Partitions to a Different RECON) and press Enter.
   The Enter Target RECON Dataset Names panel is displayed.

   Figure 131. Selecting the target RECON data sets

   Command ===>

   Enter target RECON Dataset names

   RECON1 your.new.recon1
   RECON2 your.new.recon2
   RECON3 your.new.recon3

3. Specify the RECON data sets to which you plan to copy the HALDB DBRC definitions and press Enter.
   The Copy Database to Different RECON panel is displayed.

   Figure 132. Specifying copy parameters

   Command ===>

   Copy database to different RECON. DBD is PHDO2

   Include secondary indexes
   Y Y Yes
   N No

   Enter new dataset name prefix
   your.hlq
   Add DBD name to dataset name prefix
   Y Y Yes
   N No

   Create IDCAMS for new datasets
   Y Y Yes
   N No
   Specify DATACLAS for new datasets
   Specify VOLSER for new datasets
4. Specify the following parameters:
   • Specify a new data set name prefix.
   • If you include the secondary index databases, use the DBD name as an
     additional data set qualifier.
IMS HALDB Toolkit can create the IDCAMS statements for the new files.
IMS HALDB Toolkit creates the JCL.
The first step creates a file containing DBRC statements and (optionally) a file
containing IDCAMS statements.
The DBRC statements create the HALDB DBD and its partitions. The partition
number is maintained so that image copies can be used.
Additionally, all complete image copy entries are transported to the new
RECONs.
VOLSER information is not copied.
You must have CATDS in your DBRC definition.
An optional third step uses the IDCAMS statements to allocate new files. File
sizes are copied from the original files.
You can now load your database into the new files by using the Recovery
function and accessing the new RECON.
Backing up DBRC DBD definitions

You can use the DBRC Support panel to back up DBRC DBD definitions.

Procedure

1. From the IMS HALDB Toolkit main menu, select option 9 (DBRC for HALDB) and press Enter.
   The DBRC Support panel is displayed.
2. Select option 7 from the DBRC Support menu.
   A list of HALDB master DBDs displays.
3. Select the DBD.
   A View panel displays the database and partition definition for the selected master DBD.
4. Select all rows.
5. Issue a CREATE command on the command line, and specify the file to which the definitions are to be backed up.
Part 9. Reference

The topics in this section provide you with technical references to help you interpret IMS HALDB Toolkit reports and understand the contents of the sample library file.

Topics:
- Chapter 27, “Sample reports,” on page 301
- Chapter 28, “Contents of the sample library file (SIHCSAMP),” on page 315
Chapter 27. Sample reports

This topic contains samples of reports that you might encounter as you use IMS HALDB Toolkit.

This topic contains only those reports created by IMS HALDB Toolkit. The reports created by the IMS and IMS Tools products are described in the manuals for those utilities.

Topics:
- “MSGPRINT data set” on page 302
- “IHCLIST data set” on page 303
MSGPRINT data set

The MSGPRINT data contains progress messages issued during the IMS HALDB Toolkit job.

Runtime Messages report

The following figure shows an example of the Runtime Messages report.

```
Runtime Messages
22:04:07,9 IHC010021 Utility driver started
22:04:07,9 IHC010001 IMS version is 13
22:04:08,0 IHC010011 Control cards used in this run
22:04:08,0 IHC010001 ANALYZE PART DBD(PHIDB1) -
22:04:08,0 IHC010001 PARTLIST(PHIDB11,PHIDB12) -
22:04:08,0 IHC010001 PARTNUM(3)
22:04:08,5 IHC010001 Verification started
22:04:08,5 IHC010001 IMS HALDB Toolkit level 1.1 01/26/18 01.18 07
22:04:08,7 IHC010001 List of partitions
22:04:08,7 IHC010001 PHIDB11
22:04:08,7 IHC010001 PHIDB12
22:04:09,3 IHC010001 Verification ended, return code is 0
22:04:09,3 IHC010001 Analyze started
22:04:09,3 IHC010001 Input: PHIDB11L DBSMS.DB.PHIDB1.L00001
22:04:09,4 IHC010001 Input: PHIDB11X DBSMS.DB.PHIDB1.X00001
22:04:09,4 IHC010001 Input: PHIDB11A DBSMS.DB.PHIDB1.A00001
22:04:09,4 IHC010001 Input: PHIDB12L DBSMS.DB.PHIDB1.L00002
22:04:09,4 IHC010001 Input: PHIDB12X DBSMS.DB.PHIDB1.X00002
22:04:09,4 IHC010001 Input: PHIDB12A DBSMS.DB.PHIDB1.A00002
22:04:09,7 IHC010791 HSSR region controller started
22:04:11,0 IHC010891 Analyzing DBD PHIDB1
22:04:11,2 IHC010841 64 database records read **FINAL**
22:04:11,3 IHC010791 HSSR region controller ended
22:04:11,3 IHC010001 Analyze ended, return code is 0
22:04:11,3 IHC010021 Utility driver ended
22:04:11,3 IHC010031 Highest return code is 0000
End of messages
```

Figure 133. Runtime Messages report
### IHCLIST data set

The IHCLIST data set contains several reports. The reports generated differ by which function was performed.

<table>
<thead>
<tr>
<th>Function</th>
<th>Reports</th>
</tr>
</thead>
</table>
| ANALYZEPART       | - Collect DBDs report  
                     | - Runtime Options In Effect report  
                     | - Unload report (not printed when the input data is unloaded data sets) |
| CONVERT           | - Collect DBDs report  
                     | - Runtime Options In Effect report  
                     | - Unload report                                                                 |
| MAINTAIN          | - Collect DBDs report  
                     | - Runtime Options In Effect report  
                     | - Unload report                                                                 |
| CHECK (HALDB      | HADB Analyzer report                                                  |
| Analyzer utility) |                                                                          |
| PSEXIT (Test Partition Selection Exit utility) | Partition Selection Exit report |

Chapter 27. Sample reports   303
Collect DBDs report

The Collect DBDs report shows the result of your partition settings. The report displays the number of partitions including their respective high keys. The total line is included at the top for reference.

**Partitioning for DBD GRE**

<table>
<thead>
<tr>
<th>Part</th>
<th>Roots %</th>
<th>All Segments %</th>
<th>Prefix Bytes %</th>
<th>Data Bytes %</th>
<th>Total Bytes %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>1</td>
<td>1,230,845</td>
<td>16.00</td>
<td>16,886,545</td>
<td>11.60</td>
<td>1,933,145,962</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>301,372,370</td>
<td>15.38</td>
<td>2,234,518,332</td>
</tr>
<tr>
<td>2</td>
<td>1,120,757</td>
<td>14.65</td>
<td>20,321,387</td>
<td>15.09</td>
<td>2,221,916,654</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>398,392,968</td>
<td>14.65</td>
<td>2,221,452,001</td>
</tr>
<tr>
<td>3</td>
<td>1,055,010</td>
<td>13.79</td>
<td>20,430,027</td>
<td>15.29</td>
<td>1,821,916,654</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>397,198,092</td>
<td>14.65</td>
<td>2,221,452,001</td>
</tr>
<tr>
<td>4</td>
<td>1,102,409</td>
<td>14.41</td>
<td>19,838,361</td>
<td>15.34</td>
<td>1,821,718,562</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>380,466,050</td>
<td>14.65</td>
<td>2,221,452,001</td>
</tr>
<tr>
<td>5</td>
<td>1,029,017</td>
<td>13.46</td>
<td>20,921,079</td>
<td>15.47</td>
<td>2,259,464,285</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>401,675,768</td>
<td>14.78</td>
<td>2,259,464,285</td>
</tr>
<tr>
<td>6</td>
<td>915,871</td>
<td>11.97</td>
<td>20,180,337</td>
<td>15.16</td>
<td>1,794,588,257</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>393,720,982</td>
<td>14.28</td>
<td>2,188,309,239</td>
</tr>
<tr>
<td>7</td>
<td>1,194,572</td>
<td>15.61</td>
<td>16,746,986</td>
<td>12.46</td>
<td>1,498,844,418</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>323,619,742</td>
<td>11.92</td>
<td>1,822,464,160</td>
</tr>
</tbody>
</table>

**Partitioning for DBD GRX01**

<table>
<thead>
<tr>
<th>Part</th>
<th>Roots %</th>
<th>All Segments %</th>
<th>Prefix Bytes %</th>
<th>Data Bytes %</th>
<th>Total Bytes %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>1</td>
<td>7,648,408</td>
<td>100.00</td>
<td>7,648,408</td>
<td>100.00</td>
<td>298,287,912</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>152,968,160</td>
<td>100.00</td>
<td>298,287,912</td>
</tr>
</tbody>
</table>

**Partitioning for DBD GRX02**

<table>
<thead>
<tr>
<th>Part</th>
<th>Roots %</th>
<th>All Segments %</th>
<th>Prefix Bytes %</th>
<th>Data Bytes %</th>
<th>Total Bytes %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>1</td>
<td>6,911</td>
<td>100.00</td>
<td>6,911</td>
<td>100.00</td>
<td>3,614,453</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>138,220</td>
<td>100.00</td>
<td>3,614,453</td>
</tr>
</tbody>
</table>

**Figure 134. Collect DBDs report (Part 1 of 2)**
The following descriptions apply to the fields in this report:

**Part** Displays the partition number. The partition number is assigned based on the order of high keys. Partition numbers in this report do not indicate partition IDs.

**Roots** Displays the number of root segments and its percentage of all root segments. This shows how many database records will be assigned to that partition.

**All Segments** Displays the number of segments in each partition and its percentage of all segments.

**Prefix Bytes** Displays the number of bytes (and its percentage of all bytes) that are required to store the prefix. This has particular significance for index databases. For example, the report in Figure 134 on page 304 shows that the GRX01 database has 153 MB for the prefix and 145 MB for data.

**Data Bytes** Displays the number of bytes (and its percentage of all bytes) required to store the data portion. This is the actual amount of data that will be stored on disk.

**Total Bytes** Displays the sum of prefix bytes and data bytes.

**Total** Displays the total number of bytes and segments in the database.

<table>
<thead>
<tr>
<th>Partitioning for DBD GRX03</th>
<th>Part</th>
<th>Roots</th>
<th>%</th>
<th>All Segments</th>
<th>%</th>
<th>Prefix Bytes</th>
<th>%</th>
<th>Data Bytes</th>
<th>%</th>
<th>Total Bytes</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>1</td>
<td>7,648,408</td>
<td>100.00</td>
<td>7,648,408</td>
<td>100.00</td>
<td>152,968,160</td>
<td>100.00</td>
<td>168,264,976</td>
<td>100.00</td>
<td>321,233,136</td>
<td>100.00</td>
</tr>
<tr>
<td>1</td>
<td>7,648,408</td>
<td>100.00</td>
<td>7,648,408</td>
<td>100.00</td>
<td>152,968,160</td>
<td>100.00</td>
<td>168,264,976</td>
<td>100.00</td>
<td>321,233,136</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

Key: ............ FFFFFFFFFFFFFFFF

<table>
<thead>
<tr>
<th>Partitioning for DBD GRX04</th>
<th>Part</th>
<th>Roots</th>
<th>%</th>
<th>All Segments</th>
<th>%</th>
<th>Prefix Bytes</th>
<th>%</th>
<th>Data Bytes</th>
<th>%</th>
<th>Total Bytes</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>1</td>
<td>3,182,169</td>
<td>100.00</td>
<td>3,182,169</td>
<td>100.00</td>
<td>63,643,380</td>
<td>100.00</td>
<td>89,100,732</td>
<td>100.00</td>
<td>152,744,112</td>
<td>100.00</td>
</tr>
<tr>
<td>1</td>
<td>3,182,169</td>
<td>100.00</td>
<td>3,182,169</td>
<td>100.00</td>
<td>63,643,380</td>
<td>100.00</td>
<td>89,100,732</td>
<td>100.00</td>
<td>152,744,112</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

Key: .................................... FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

**Figure 135. Collect DBDs report (Part 2 of 2)**
Runtime Options In Effect report

The Runtime Options In Effect report summarizes the runtime options that were applied to the job.

HALDB Conversion Analyzer

Runtime Options in effect

Primary DBD.........................: PHIDB1
Source for Analyzer...............: Database
Method of Operation................: Analyze andUnload
Database Type......................: HALDB
Partition............................: PHIDB11
Partition............................: PHIDB12
Partitioning for primary DBD.......: Partition Number
Number of Partitions...............: 3
Partitioning for secondary Indexes.: Partition Size
Size of a Partition...............: 1024

Figure 136. Runtime Options In Effect report

The following descriptions apply to the fields in this report:

Primary DBD
The name of the primary DBD.

Source for Analyzer
The input provided to the Analyzer. This field shows either Database or Unload file.

Method of Operation
Whether an unload was performed. This field shows either Analyze only or Analyze and Unload.

Database Type
The type of the database. This field shows either HALDB or Not HALDB.

Partition
The name of the partition that was processed.

Partitioning for primary DBD
The method used for partitioning the primary DBD. This field shows one of the following methods:
- Partition Number
- Partition Size
- High Keys
- Add empty partition

Partitioning for secondary Indexes
The method used for partitioning the secondary indexes. This field shows one of the following methods:
- Partition Number
- Partition Size
- High Keys
- Along with primary DBD
Unload report

The Unload report shows the number of segments and the segment size. The segment size is the true size as it is stored in the database.

It can be useful to know the average database record length when tuning your database. In this case, it is helpful for specifying the byte limit parameter of your PHDAM database.

A sample report is shown in the following example:

---

### Unload for DBD GRE

<table>
<thead>
<tr>
<th>Segment</th>
<th>Segment Count</th>
<th>Segment Data</th>
<th>Prefix and Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRA</td>
<td>7,648,481</td>
<td>1,541,514,136</td>
<td>1,373,247,554</td>
</tr>
<tr>
<td>GRA5</td>
<td>3,182,169</td>
<td>57,466,524</td>
<td>76,559,538</td>
</tr>
<tr>
<td>GRB</td>
<td>29,744,738</td>
<td>1,633,250,273</td>
<td>2,007,332,787</td>
</tr>
<tr>
<td>GRC</td>
<td>25,221,133</td>
<td>805,503,608</td>
<td>654,176,810</td>
</tr>
<tr>
<td>GRE</td>
<td>24,133,135</td>
<td>965,325,400</td>
<td>1,110,124,210</td>
</tr>
<tr>
<td>GRH</td>
<td>4,860,332</td>
<td>231,274,879</td>
<td>260,436,871</td>
</tr>
<tr>
<td>G</td>
<td>16,617,211</td>
<td>1,893,610,390</td>
<td>1,993,313,656</td>
</tr>
<tr>
<td>RL</td>
<td>1,998,779</td>
<td>95,669,886</td>
<td>107,662,560</td>
</tr>
<tr>
<td>GRO</td>
<td>18,415,591</td>
<td>1,214,534,914</td>
<td>1,325,028,460</td>
</tr>
<tr>
<td>GRQ</td>
<td>3,503,153</td>
<td>229,727,199</td>
<td>250,746,117</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>135,324,722</strong></td>
<td><strong>8,268,809,017</strong></td>
<td><strong>13,974,016,149</strong></td>
</tr>
</tbody>
</table>

Avg. database record length: 1,827.03

---

*Figure 137: Unload for DBD GRE report*

The following descriptions apply to the fields in this report:

**Segment**
- Displays the segment name

**Segment Count**
- Displays the number of segments of this type in the database

**Segment Data**
- Displays the total amount of data bytes for this segment type in the database

**Prefix and Data**
- Displays the total bytes of this segment type in the database

**Total**
- Displays the sum of the segment types

**Avg. database record length**
- Displays the result of dividing the total prefix and data bytes by the number of root segments
HALDB Analyzer report

This report shows analysis information gathered by the HALDB Analyzer utility.

Report for database PHDO2

Exceptions and errors

No errors or exceptions detected

Threshold detail

<table>
<thead>
<tr>
<th>Item</th>
<th>Low</th>
<th>High</th>
<th>Curr</th>
<th>Warn</th>
<th>Crit</th>
<th>Detail</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAPUSE</td>
<td>60</td>
<td>80</td>
<td>9</td>
<td>27</td>
<td></td>
<td></td>
<td>300</td>
</tr>
<tr>
<td>ROOTOVL</td>
<td>1</td>
<td>30</td>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>ROOTHOME</td>
<td>2</td>
<td>20</td>
<td>23</td>
<td></td>
<td>7</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>SYNONYM</td>
<td>n/a</td>
<td>n/a</td>
<td>10</td>
<td></td>
<td>3</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>SPLIT</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>PTRHEAL</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td>265</td>
</tr>
<tr>
<td>AVGFSE</td>
<td>1</td>
<td>30</td>
<td>0.9</td>
<td></td>
<td>393</td>
<td></td>
<td>421</td>
</tr>
</tbody>
</table>

Database datasets

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Extents</th>
<th>BLKSIZE</th>
<th>Records</th>
<th>Allo (KB)</th>
<th>Used (KB)</th>
<th>Overhead</th>
<th>Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHDO20L</td>
<td>1</td>
<td>4,096</td>
<td>0</td>
<td>720</td>
<td>720</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PHDO20A</td>
<td>1</td>
<td>4,096</td>
<td>329</td>
<td>1,440</td>
<td>1,316</td>
<td>5,792</td>
<td>1,325</td>
</tr>
<tr>
<td>PHDO21L</td>
<td>1</td>
<td>4,096</td>
<td>0</td>
<td>720</td>
<td>720</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PHDO21A</td>
<td>1</td>
<td>4,096</td>
<td>46</td>
<td>1,440</td>
<td>184</td>
<td>4,660</td>
<td>171</td>
</tr>
<tr>
<td>PHDO22L</td>
<td>1</td>
<td>4,096</td>
<td>0</td>
<td>720</td>
<td>720</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PHDO22A</td>
<td>1</td>
<td>4,096</td>
<td>46</td>
<td>1,440</td>
<td>184</td>
<td>4,660</td>
<td>171</td>
</tr>
</tbody>
</table>

Total 6,480 3,844

Database segments

<table>
<thead>
<tr>
<th>Segment Name</th>
<th>Segment Count</th>
<th>Total in RAA</th>
<th>av. length</th>
<th>Total bytes</th>
<th>Data bytes</th>
<th>Prefix bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>30</td>
<td>29 82</td>
<td>2,460</td>
<td>1,920</td>
<td>540</td>
<td></td>
</tr>
<tr>
<td>DEPO1</td>
<td>235</td>
<td>31 1,022</td>
<td>240,170</td>
<td>235,000</td>
<td>5,170</td>
<td></td>
</tr>
<tr>
<td>DEPO2</td>
<td>1,402</td>
<td>165 1,018</td>
<td>1,427,236</td>
<td>1,402,000</td>
<td>25,236</td>
<td></td>
</tr>
</tbody>
</table>

Total 1,667 225 1,669,866 1,638,920 30,946

Avg. database record length 55,662

Figure 138. HALDB Analyzer report (Part 1 of 6)
Report for partition PHDO20

Exception details
==================
Roots in partition...........: 24
Partition selection errors...: 0
Randomizer errors............: 0
ILKs in partition............: 1,325
Erroneous ILKs...............: 0
Number of missing ILEs.......: 0

Threshold detail
================
<table>
<thead>
<tr>
<th>Item</th>
<th>Low</th>
<th>High</th>
<th>Curr</th>
<th>Warn</th>
<th>Crit</th>
<th>Detail</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAPUSE</td>
<td>60</td>
<td>80</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>ROOTOVFL</td>
<td>1</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROOTHOME</td>
<td>2</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNONYM</td>
<td>n/a</td>
<td>n/a</td>
<td>12</td>
<td>3</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPLIT</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td>300</td>
<td>2500</td>
<td>1</td>
<td>327,134</td>
<td>1,327,134</td>
<td>1,327,134</td>
<td></td>
</tr>
</tbody>
</table>

Partition datasets
===================
<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Extents</th>
<th>BLKSIZE</th>
<th>Records</th>
<th>Allo (KB)</th>
<th>Used (KB)</th>
<th>Overhead</th>
<th>Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHDO20L</td>
<td>1</td>
<td>4,096</td>
<td>0</td>
<td>720</td>
<td>720</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PHDO20A</td>
<td>1</td>
<td>4,096</td>
<td>329</td>
<td>1,440</td>
<td>1,316</td>
<td>5,792</td>
<td>1,325</td>
</tr>
<tr>
<td>Total</td>
<td>2,160</td>
<td>2,036</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Partition segments
===================
<table>
<thead>
<tr>
<th>Segment Name</th>
<th>Segment Count</th>
<th>Total</th>
<th>avg. length</th>
<th>Total bytes</th>
<th>Data bytes</th>
<th>Prefix bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>24</td>
<td>24</td>
<td>82</td>
<td>1,968</td>
<td>1,536</td>
<td>432</td>
</tr>
<tr>
<td>DEP01</td>
<td>187</td>
<td>11</td>
<td>1,022</td>
<td>191,114</td>
<td>187,000</td>
<td>4,114</td>
</tr>
<tr>
<td>DEP02</td>
<td>1,114</td>
<td>54</td>
<td>1,018</td>
<td>1,134,052</td>
<td>1,114,000</td>
<td>20,052</td>
</tr>
<tr>
<td>Total</td>
<td>1,325</td>
<td>89</td>
<td></td>
<td>1,327,134</td>
<td>1,302,536</td>
<td>24,598</td>
</tr>
<tr>
<td>Avg. database record length</td>
<td>55,297</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Partition pointer
==================
<table>
<thead>
<tr>
<th>Segment</th>
<th>Type</th>
<th>Target</th>
<th>Total</th>
<th>Nonzero</th>
<th>Pcnt</th>
<th>Other block</th>
<th>Pcnt</th>
<th>Other in RAA</th>
<th>Pcnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>RAP</td>
<td>ROOT</td>
<td>100</td>
<td>21</td>
<td>21.00</td>
<td>5</td>
<td>23.80</td>
<td>5</td>
<td>23.80</td>
</tr>
<tr>
<td>ROOT</td>
<td>TF</td>
<td>ROOT</td>
<td>24</td>
<td>3</td>
<td>12.50</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>PCF DEP01</td>
<td>24</td>
<td>24</td>
<td>100.00</td>
<td>22</td>
<td>91.66</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Split data</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>DEP01</td>
<td>TF</td>
<td>DEP01</td>
<td>187</td>
<td>163</td>
<td>87.16</td>
<td>163</td>
<td>100.00</td>
<td>9</td>
<td>5.52</td>
</tr>
<tr>
<td>PCF DEP02</td>
<td>187</td>
<td>187</td>
<td>100.00</td>
<td>44</td>
<td>23.52</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>DEP02</td>
<td>TF</td>
<td>DEP02</td>
<td>1,114</td>
<td>927</td>
<td>83.21</td>
<td>229</td>
<td>24.70</td>
<td>8</td>
<td>0.86</td>
</tr>
<tr>
<td>Total</td>
<td>1,636</td>
<td>1,325</td>
<td>80.99</td>
<td>463</td>
<td>34.94</td>
<td>22</td>
<td>1.66</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 139. HALDB Analyzer report (Part 2 of 6)
Report for partition PHDO21
Exception details
=========================
Roots in partition..........: 3
Partition selection errors...: 0
Randomizer errors...........: 0
ILKs in partition..........: 171
Erroneous ILKs..............: 0
Number of missing ILEs.....: 0

Threshold detail
=================
<table>
<thead>
<tr>
<th>Item</th>
<th>Low</th>
<th>High</th>
<th>Curr</th>
<th>Warn</th>
<th>Crit</th>
<th>Detail</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAPUSE</td>
<td>60</td>
<td>80</td>
<td>3</td>
<td></td>
<td>3</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>ROOTOVFL</td>
<td>1</td>
<td>30</td>
<td>0</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROOTHOME</td>
<td>2</td>
<td>20</td>
<td>33</td>
<td></td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>SYNONYM</td>
<td>n/a</td>
<td>n/a</td>
<td>0</td>
<td></td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>SPLIT</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVGSE</td>
<td>1</td>
<td>30</td>
<td>0.7</td>
<td>36</td>
<td>46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPACE</td>
<td>300</td>
<td>2500</td>
<td>0</td>
<td></td>
<td>*</td>
<td>171,366</td>
<td></td>
</tr>
</tbody>
</table>

Partition datasets
==================
<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Extents</th>
<th>BLKSIZE</th>
<th>Records</th>
<th>Allo (KB)</th>
<th>Used (KB)</th>
<th>Overhead</th>
<th>Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHDO21L</td>
<td>1</td>
<td>4,096</td>
<td>0</td>
<td>720</td>
<td>720</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PHDO21A</td>
<td>1</td>
<td>4,096</td>
<td>46</td>
<td>1,440</td>
<td>184</td>
<td>4,660</td>
<td>171</td>
</tr>
<tr>
<td>Total</td>
<td>2,160</td>
<td>904</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Partition segments
==================
<table>
<thead>
<tr>
<th>Segment Name</th>
<th>Segment Count</th>
<th>in RAA avg. length</th>
<th>Total bytes</th>
<th>Data bytes</th>
<th>Prefix bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>3</td>
<td>3 82</td>
<td>246</td>
<td>192</td>
<td>54</td>
</tr>
<tr>
<td>DEP01</td>
<td>24</td>
<td>10 1,022</td>
<td>24,528</td>
<td>24,000</td>
<td>528</td>
</tr>
<tr>
<td>DEP02</td>
<td>144</td>
<td>56 1,018</td>
<td>146,592</td>
<td>144,000</td>
<td>2,592</td>
</tr>
<tr>
<td>Total</td>
<td>171</td>
<td>69 171,366</td>
<td>168,192</td>
<td>3,174</td>
<td></td>
</tr>
<tr>
<td>Avg. database record length</td>
<td>57,122</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Partition pointer
==================
<table>
<thead>
<tr>
<th>Segment</th>
<th>Type</th>
<th>Target</th>
<th>Total</th>
<th>Nonzero</th>
<th>Pcnt</th>
<th>Other block</th>
<th>Pcnt</th>
<th>Other in RAA</th>
<th>Pcnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>RAP</td>
<td>ROOT</td>
<td>100</td>
<td>3</td>
<td>3.00</td>
<td>1</td>
<td>33.33</td>
<td>1</td>
<td>33.33</td>
</tr>
<tr>
<td>ROOT</td>
<td>TF</td>
<td>ROOT</td>
<td>3</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PCF DEP01</td>
<td>3</td>
<td>3 100.00</td>
<td>2</td>
<td>66.66</td>
<td>1</td>
<td>33.33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split data</td>
<td>0</td>
<td>0 0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEP01</td>
<td>TF</td>
<td>DEP01</td>
<td>24</td>
<td>21 87.50</td>
<td>21 100.00</td>
<td>8</td>
<td>38.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCF DEP02</td>
<td>24</td>
<td>24 100.00</td>
<td>4</td>
<td>16.66</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEP02</td>
<td>TF</td>
<td>DEP02</td>
<td>144</td>
<td>120 83.33</td>
<td>27 22.50</td>
<td>9</td>
<td>7.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>298</td>
<td>171 57.38</td>
<td>55</td>
<td>32.16</td>
<td>19</td>
<td>11.11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 140. HALDB Analyzer report (Part 3 of 6)
Report for partition PHDO22
Exception details
=====================
Roots in partition...........: 3
Partition selection errors...: 0
Randomizer errors............: 0
ILKs in partition............: 171
Erroneous ILKs...............: 0
Number of missing ILEs.......: 0

Threshold detail
================
<table>
<thead>
<tr>
<th>Item</th>
<th>Low</th>
<th>High</th>
<th>Curr</th>
<th>Warn</th>
<th>Crit</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAPUSE</td>
<td>60</td>
<td>80</td>
<td>3</td>
<td>3</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>ROOTOVFL</td>
<td>1</td>
<td>30</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>ROOTHOME</td>
<td>2</td>
<td>20</td>
<td>33</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>SYNONYM</td>
<td>n/a</td>
<td>n/a</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>SPLIT</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AVGSE</td>
<td>1</td>
<td>30</td>
<td>0.8</td>
<td>37</td>
<td>46</td>
<td>0</td>
</tr>
<tr>
<td>SPACE</td>
<td>300</td>
<td>2500</td>
<td>0</td>
<td>*</td>
<td>171,366</td>
<td>171,366</td>
</tr>
</tbody>
</table>

Partition datasets
=====================
<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Extents</th>
<th>BLKSIZE</th>
<th>Records</th>
<th>Allo (KB)</th>
<th>Used (KB)</th>
<th>Overhead</th>
<th>Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHDO22L</td>
<td>1</td>
<td>4,096</td>
<td>0</td>
<td>720</td>
<td>720</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PHDO22A</td>
<td>1</td>
<td>4,096</td>
<td>46</td>
<td>1,440</td>
<td>184</td>
<td>4,660</td>
<td>171</td>
</tr>
<tr>
<td>Total</td>
<td>2,160</td>
<td>904</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Partition segments
=====================
<table>
<thead>
<tr>
<th>Segment Name</th>
<th>Segment Count</th>
<th>Segment Total in RAA</th>
<th>avg. length</th>
<th>Total bytes</th>
<th>Data bytes</th>
<th>Prefix bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>3</td>
<td>2</td>
<td>82</td>
<td>246</td>
<td>192</td>
<td>54</td>
</tr>
<tr>
<td>DEP01</td>
<td>24</td>
<td>10</td>
<td>1,022</td>
<td>24,528</td>
<td>24,000</td>
<td>528</td>
</tr>
<tr>
<td>DEP02</td>
<td>144</td>
<td>55</td>
<td>1,018</td>
<td>146,592</td>
<td>144,000</td>
<td>2,592</td>
</tr>
<tr>
<td>Total</td>
<td>171</td>
<td>67</td>
<td>171,366</td>
<td>168,192</td>
<td>3,174</td>
<td></td>
</tr>
</tbody>
</table>

Avg. database record length 57,122

Partition pointer
=====================
<table>
<thead>
<tr>
<th>Segment Type</th>
<th>Target</th>
<th>Total</th>
<th>Nonzero</th>
<th>Pcnt</th>
<th>Other block</th>
<th>Pcnt</th>
<th>Other in RAA</th>
<th>Pcnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>RAP</td>
<td>100</td>
<td>3</td>
<td>3.00</td>
<td>1</td>
<td>33.33</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>ROOT</td>
<td>TF</td>
<td>3</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>PCF</td>
<td>DEP01</td>
<td>3</td>
<td>3</td>
<td>100.00</td>
<td>2</td>
<td>66.66</td>
<td>1</td>
<td>33.33</td>
</tr>
<tr>
<td>Split data</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>DEP01</td>
<td>TF</td>
<td>24</td>
<td>21</td>
<td>87.50</td>
<td>21</td>
<td>100.00</td>
<td>8</td>
<td>38.09</td>
</tr>
<tr>
<td>PCF</td>
<td>DEP02</td>
<td>24</td>
<td>24</td>
<td>100.00</td>
<td>3</td>
<td>12.50</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>DEP02</td>
<td>TF</td>
<td>144</td>
<td>120</td>
<td>83.33</td>
<td>34</td>
<td>28.33</td>
<td>15</td>
<td>12.50</td>
</tr>
<tr>
<td>Total</td>
<td>298</td>
<td>171</td>
<td>57.38</td>
<td>61</td>
<td>35.67</td>
<td>24</td>
<td>14.03</td>
<td></td>
</tr>
</tbody>
</table>

Figure 141. HALDB Analyzer report (Part 4 of 6)
Report for index PSI021

Report for partition PSI0210

Exception details
===============
Total index records........: 23
Deleted index records.....: 0
Pointer healing needed....: 0
CI splits..................: 0
CA splits..................: 0
CA size (CIs).............: 180
Freespace CI (%).........: 0
Freespace CA (%).........: 0

Partition datasets
=================
<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Extents</th>
<th>BLKSIZE</th>
<th>Records</th>
<th>Allo (KB)</th>
<th>Used (KB)</th>
<th>Overhead</th>
<th>Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSI0210A</td>
<td>1</td>
<td>4,096</td>
<td>23</td>
<td>1,440</td>
<td>720</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>1,440</td>
<td>720</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Report for partition PSI0211

Exception details
===============
Total index records........: 7
Deleted index records.....: 0
Pointer healing needed....: 0
CI splits..................: 0
CA splits..................: 0
CA size (CIs).............: 180
Freespace CI (%).........: 0
Freespace CA (%).........: 0

Partition datasets
=================
<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Extents</th>
<th>BLKSIZE</th>
<th>Records</th>
<th>Allo (KB)</th>
<th>Used (KB)</th>
<th>Overhead</th>
<th>Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSI0211A</td>
<td>1</td>
<td>4,096</td>
<td>7</td>
<td>1,440</td>
<td>720</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>1,440</td>
<td>720</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 142. HALDB Analyzer report (Part 5 of 6)
Report for index PSIO22

Report for partition PSIO220

Exception details
================
Total index records..........: 96
Deleted index records.......: 0
Pointer healing needed.....: 0
CI splits...................: 1
CA splits...................: 0
CA size (CIs)...............: 180
Freespace CI (%)..........: 0
Freespace CA (%)...........: 0

Partition datasets
==================
DDNAME Extents BLKSIZE Records Allo (KB) Used (KB) Overhead Segments
---------------------------------------------------------------
PSIO220A 1 4,096 96 1,440 720 0 96
---------------------------------------------------------------
Total 1,440 720

Report for partition PSIO221

Exception details
================
Total index records..........: 139
Deleted index records.......: 0
Pointer healing needed.....: 0
CI splits...................: 1
CA splits...................: 0
CA size (CIs)...............: 180
Freespace CI (%)..........: 0
Freespace CA (%)...........: 0

Partition datasets
==================
DDNAME Extents BLKSIZE Records Allo (KB) Used (KB) Overhead Segments
---------------------------------------------------------------
PSIO221A 1 4,096 139 1,440 720 0 139
---------------------------------------------------------------
Total 1,440 720

Figure 143. HALDB Analyzer report (Part 6 of 6)
Partition Selection Exit report

This report shows information gathered by the partition selection test tool.

<table>
<thead>
<tr>
<th>All partitions</th>
<th>Segment</th>
<th>Pref-length</th>
<th>Segment</th>
<th>Pref-length</th>
<th>Data</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>before</td>
<td>after</td>
<td>Count</td>
<td>Bytes</td>
<td>Bytes</td>
</tr>
<tr>
<td>ROOT</td>
<td>18</td>
<td>18</td>
<td>30</td>
<td>540</td>
<td>1920</td>
<td>2460</td>
</tr>
<tr>
<td>DEP01</td>
<td>22</td>
<td>22</td>
<td>235</td>
<td>5170</td>
<td>235000</td>
<td>240170</td>
</tr>
<tr>
<td>DEP02</td>
<td>18</td>
<td>18</td>
<td>1402</td>
<td>25236</td>
<td>1402000</td>
<td>1427236</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>1667</td>
<td>30946</td>
<td>1638920</td>
<td>1669866</td>
</tr>
<tr>
<td>ILE records</td>
<td></td>
<td></td>
<td></td>
<td>265</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| ILE records    | 58      |             |         |            |      |       |

<table>
<thead>
<tr>
<th>Partition 1</th>
<th>Segment</th>
<th>Pref-length</th>
<th>Segment</th>
<th>Pref-length</th>
<th>Data</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>before</td>
<td>after</td>
<td>Count</td>
<td>Bytes</td>
<td>Bytes</td>
</tr>
<tr>
<td>ROOT</td>
<td>18</td>
<td>18</td>
<td>7</td>
<td>126</td>
<td>448</td>
<td>574</td>
</tr>
<tr>
<td>DEP01</td>
<td>22</td>
<td>22</td>
<td>51</td>
<td>1122</td>
<td>51000</td>
<td>52122</td>
</tr>
<tr>
<td>DEP02</td>
<td>18</td>
<td>18</td>
<td>298</td>
<td>5364</td>
<td>298000</td>
<td>303364</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>356</td>
<td>6612</td>
<td>349448</td>
<td>356060</td>
</tr>
<tr>
<td>ILE records</td>
<td></td>
<td></td>
<td></td>
<td>58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partition 2</th>
<th>Segment</th>
<th>Pref-length</th>
<th>Segment</th>
<th>Pref-length</th>
<th>Data</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td>before</td>
<td>after</td>
<td>Count</td>
<td>Bytes</td>
<td>Bytes</td>
</tr>
<tr>
<td>ROOT</td>
<td>18</td>
<td>18</td>
<td>23</td>
<td>414</td>
<td>1472</td>
<td>1886</td>
</tr>
<tr>
<td>DEP01</td>
<td>22</td>
<td>22</td>
<td>184</td>
<td>4048</td>
<td>184000</td>
<td>188048</td>
</tr>
<tr>
<td>DEP02</td>
<td>18</td>
<td>18</td>
<td>1104</td>
<td>19872</td>
<td>1104000</td>
<td>1123872</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>1311</td>
<td>24334</td>
<td>1289472</td>
<td>1313806</td>
</tr>
<tr>
<td>ILE records</td>
<td></td>
<td></td>
<td></td>
<td>207</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Database record distribution

<table>
<thead>
<tr>
<th>Part</th>
<th>Roots</th>
<th>%</th>
<th>Data</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>23.33</td>
<td>356060</td>
<td>21.32</td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>76.66</td>
<td>1313806</td>
<td>78.67</td>
</tr>
</tbody>
</table>

Figure 144. Partition selection test tool report
Chapter 28. Contents of the sample library file (SIHCSAMP)

This reference topic provides descriptions for the contents of the IMS HALDB Toolkit sample library file (SIHCSAMP).

The following members are included in sample library SIHCSAMP:

**IHCCHAL**
This member provides the sample skeleton CLIST for ISPF invocation.

**Requirement:** The ISPF data set names must match the target library names.

**IHCCHAL1**
This member provides the skeleton CLIST for invocation from control suite.

**IHCCKSDS**
This member provides the sample JCL for ISPF installation.

**IHCCSET**
This member provides the sample JCL for installation setup parameters that are used on a routine basis.

**IHCEACBL**
This member provides the sample JCL to create a list of PSBs that contain a specified DBD.

**IHCEANA**
This member provides the sample JCL to run the HALDB analyzer.

**IHCECLON**
This member provides the sample JCL to clone DBRC for test environments.

**IHCECOPY**
This member provides the sample JCL to copy a database to a different IMS (and RECON).

**IHCECVTN**
This member provides the sample JCL to convert to HALDB offline.

**IHCECVTY**
This member provides the sample JCL to convert to HALDB online.

**Note:** This member is not supported by IMS Database Utility Solution.

**IHCECVT1**
This member provides the sample JCL to convert to HALDB by specifying the number of partitions.

**IHCECVT2**
This member provides the sample JCL to convert to HALDB by specifying the partition size.

**IHCECVT3**
This member provides the sample JCL to convert to HALDB by specifying the high keys of the partition.
IHCEDBBK
This member provides the sample JCL to back up DBRC definitions for a specified DBD.

IHCEDBD
This member provides the sample JCL to re-create DBD source from DBDLIB.

IHCEDBRC
This member provides the sample JCL to relink the DBRC module DSPCRTR0.

IHCEDBRI
This member provides the sample JCL to set up HALDB data set name substitution outside of DBRC.

IHCEDFDS
This member provides the sample JCL to copy a database to a different IMS using DFSMSdss.

IHCEDFSH
This member provides the sample JCL to create DFSHALDB statement using keys.

IHCEDROP
This member provides the sample JCL to drop a single partition.

IHCEILK
This member provides the sample JCL to reset ILK.

IHCEIXHL
This member provides the sample JCL for the Index Pointer Healer.

IHCLOAD
This member provides the sample JCL to load a single partition.

IHCEMAI1
IHCEMAI2
IHCEMAI3
These members provide the sample JCL to run the analysis portion of the CONVERT and MAINTAIN commands.

IHCEMERG
This member provides the sample JCL to merge two HALDBs into one.

IHCEMTNN
This member provides the sample JCL to maintain HALDB partitions offline.

IHCEMTNY
This member provides the sample JCL to maintain HALDB partitions online.

Note: This member is not supported by IMS Database Utility Solution.

IHCEMTN1
This member provides the sample JCL to split two partitions into four partitions.

IHCEMTN2
This member provides the sample JCL to rearrange partitions using the partition size.
IHCEMTN3
This member provides the sample JCL to rearrange partitions using high keys.

IHCEOSAM
This member provides the sample JCL for OSAM multivolume allocation (non-SMS).

IHCEPSEL
This member provides the sample JCL for the partition selection test tool.

IHCEREOI
This member provides the sample JCL to reorganize a PSINDEX.

IHCESAM1
IHCESAM2
IHCESAM3
IHCESAM4
IHCESAM5
IHCESAM6
These members provide the sample JCL to create rules for substituting HALDB data set names outside of DBRC.

IHCESPLT
This member provides the sample JCL to split an unload file on a partition boundary.

IHCEUNIT
This member provides the sample JCL to show all DASD unit names and mark VIO.
Part 10. Troubleshooting

The topics in this section provide you with technical references to help you troubleshoot and diagnose IMS HALDB Toolkit problems.

Topics:

• Chapter 29, “Messages and codes,” on page 321
• Chapter 30, “Gathering diagnostic information,” on page 355
Chapter 29. Messages and codes

This section provides detailed information about IMS HALDB Toolkit messages and codes.

Topics:
- “Return and abend codes” on page 322
- “Trace file or trace panel” on page 323
- “ISPF messages” on page 323
- “Batch messages” on page 324
Return and abend codes

This reference section provides detailed information about IMS HALDB Toolkit return and abend codes.

You might encounter the following return or abend codes during processing.

A return or abend code other than 0 is an unusual condition.

An abend code of U 999 is issued after a subtask abends. The subtask abend code is also shown in the messages.

Either of the following return codes is issued:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Function completed successfully.</td>
</tr>
<tr>
<td>4</td>
<td>One or more of the following is true:</td>
</tr>
<tr>
<td></td>
<td>• Not all functions completed.</td>
</tr>
<tr>
<td></td>
<td>• You may need to perform an IMS online change</td>
</tr>
<tr>
<td></td>
<td>to complete the requested function.</td>
</tr>
<tr>
<td></td>
<td>• TAKEOVER(DELAY) was specified.</td>
</tr>
</tbody>
</table>
Trace file or trace panel

In a normal situation, a trace file is automatically attached to batch processing.

In the event of an error situation, you should attach the trace file to your error documentation to assist IBM Software Support in resolving the problem.

A trace panel is a screen containing trace information which depicts an error situation. When you submit error documentation, print the screen and attach it to the documentation to assist IBM Software Support.

For more information about the diagnostic information that you should attach when contacting IBM Software Support, see Chapter 30, “Gathering diagnostic information,” on page 355.

ISPF messages

For explanations of ISPF messages, see the appropriate ISPF publication.
Batch messages

This reference section provides detailed information about IMS HALDB Toolkit batch messages.

Message format

IMS HALDB Toolkit messages adhere to the following format:

IHC0nnnnx

Where:

IHC0  Indicates that the message was issued by IMS HALDB Toolkit

nnnn  Indicates the message identification number

x      Indicates the severity of the message:

A      Indicates that operator intervention is required before processing can continue.

E      Indicates that an error occurred, which might or might not require operator intervention.

I      Indicates that the message is informational only.

W      Indicates that the message is a warning to alert you to a possible error condition.

Each message also includes the following information:

Explanation:

The Explanation section explains what the message text means, why it occurred, and what its variables represent.

System action:

The System action section explains what the system will do in response to the event that triggered this message.

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.

---

IHC01000E  text

Explanation: The text of this message provides information about the error.

System action: None.

User response: None.

IHC01003I  text

Explanation: Displays the control statements that you supplied.

System action: None.

User response: None. This message is informational.

IHC01000E  Utility driver [started | ended | terminated] with error.

Explanation: This message indicates the start or termination of a batch run. Previous messages indicate a possible error reason.

System action: None.

User response: None. This message is informational.

---

IHC01001I  Control cards used in this run

IHC01003I  Highest return code is nnnn
Explanation: The highest return code of the utilities and functions is nnnn.

System action: None.

User response: None. This message is informational.

IHC01004E Error return code is nnnn, reason code rrrr

Explanation: A utility function ended with an error. The return code is indicated by nnnn, and the reason code is indicated by rrrr.

System action: The utility has been terminated.

User response: Evaluate the preceding error messages to determine the cause of the problem and resolve the error condition.

IHC01005E Expected continuation not received.

Explanation: The control statements are in error. A continuation was indicated, but no additional control statement was found.

System action: Processing terminates.

User response: Correct the control statement error and resubmit the job.

IHC01006E No commands found.

Explanation: The IHCSYSIN file was specified, but no control statements were specified.

System action: Processing terminates.

User response: Provide the necessary control statements.

IHC01007E Invalid command cmd found in line nnn.

Explanation: The command cmd is invalid. The control statement line is indicated by nnn.

System action: Processing terminates.

User response: Correct the control statement and resubmit the job.

IHC01008E Invalid keyword key found in line nnn.

Explanation: The keyword key is not valid for the specified command.

System action: Processing terminates.

User response: Correct the control statement and resubmit the job.

IHC01009E Invalid syntax for keyword key in line nnn.

Explanation: The keyword requires syntax rules. Syntax rules are: numeric, dname, dsname, or simply yes, or no. Message IHC01014I is issued to show the reason for the error.

System action: Processing terminates.

User response: Correct the control statement and resubmit the job.

IHC01010E Duplicate keyword key in line nnn.

Explanation: The keyword key is being used more than once in the same command.

System action: Processing terminates.

User response: Remove the duplicate keyword and resubmit the job.

IHC01011E No command found in line nnn.

Explanation: A command must start within the first 20 bytes of a control statement. No command was found.

System action: Processing terminates.

User response: Correct the control statement and resubmit the job.

IHC01012E Command cmd requires keyword key.

Explanation: The indicated keyword key is required for this command cmd.

System action: Processing terminates.

User response: Add the keyword and resubmit the job.

IHC01013E Keyword key and key are mutually exclusive.

Explanation: Mutually exclusive keywords have been specified.

System action: Processing terminates.

User response: Correct the control statement and resubmit the job.

IHC01014I Reason is rsn

Explanation: This message explains control statement errors. The variable rsn provides information about the control statement errors.

System action: None.

User response: None. This message is informational.
IHC01015E  Keyword key requires keyword key
Explanation: Both of the specified keywords are required and must be specified.
System action: Processing terminates.
User response: Add the missing keyword and resubmit the job.

IHC01016I  Command ccccc started/ended/terminated
Explanation: Command ccccc has either started or ended.
System action: None.
User response: None. This message is informational.

IHC01017E  Nothing to schedule
Explanation: After all control cards have been checked, there was no command to be processed.
System action: Processing terminates.
User response: Provide control statements.

IHC01020E  DBD nnnnn is in error, reason: rrr
Explanation: The DBD decoder found errors for DBD nnnnn. These errors are probably due to one of the following reasons rrr:
  • DBD has logical relation.
  • The DBD version is not supported.
  • The DBD type is not supported.
  • DBD nnnnn is not a valid DBD.
  • The DBD name is different from the member name. The member name in the DBD library and the name inside the DBD are different. The name inside the DBD is the valid name.
  • DOS does not support this DBD.
  • The index DBD has no primary DBD.
• The index DBD has more than one target DBD. IMS HALDB Toolkit does not support multiple target DBDs.
• The target DBD is an IMS HALDB Toolkit DBD, but the index DBD is not.
• The index DBD is an IMS HALDB Toolkit DBD, but the target DBD is not.
System action: Processing stops.
User response: Correct the invalid DBDs and restart the process.

IHC01021E  IMS or DBDLIB DD-statement missing
Explanation: The JCL is missing one or both of these statements.
System action: Processing stops.
User response: Add the IMS DD statement describing the appropriate DBDLIBs.

IHC01018E  IMS or DBDLIB DD-statement missing
Explanation: The JCL is missing one or both of these statements.
System action: Processing stops.
User response: Add the IMS DD statement describing the appropriate DBDLIBs.

IHC01022E  RESLIB not in STEPLIB concatenation
Explanation: The IMS SDFSRESL is not available to this job step.
System action: Processing stops.
User response: Add the IMS SDFSRESL to the STEPLIB.

IHC01023E  Unsupported level of IMS is being used
Explanation: IMS HALDB Toolkit is run under an unsupported version of IMS.
System action: Processing stops.
User response: Run IMS HALDB Toolkit with a supported version of IMS.

IHC01024E  DBD is not a PSINDEX DBD
Explanation: A HALDB function was requested, but the selected DBD is not a PSINDEX DBD.
System action: The function (step) ends.
User response: Select a PSINDEX DBD. This is a secondary index DBD which has been converted to HALDB.

IHC01025E  DBD nnnnn not found
Explanation: DBD nnnnn was requested, but is not in the DBD library.
System action: The job step ends.
User response: Ensure that you are using the correct DBD library; the IMS DD statement is pointing to those files.
IHC01026E KEYLEN + KEYOFF exceeds root key length of nnn

Explanation: The PSEXIT command is used to show the results of the partition selection exit IHCPSEL0.

System action: The job step ends.
User response: Correct the KEYLEN, KEYOFF, or both specifications, and restart the process.

IHC01027E DBD xxxxx is already HALDB

Explanation: The PSEXIT command is used with full function databases to test the functionality and distribution of a partition selection exit. The specified DBD is already HALDB.

System action: The job step ends.
User response: None.

IHC01028E No keys specified

Explanation: The partition selection exit, IHCPSEL0, requires that key strings be specified. No keys were found.

System action: The job step ends.
User response: Provide keys in the //KEYS DD statement.

IHC01029E Error during initialization of nnnnn

Explanation: The partition selection exit, IHCPSEL0, returned an error during its initialization phase.

System action: The job step ends.
User response: If the exit is IHCPSEL0, the reason is located in the messages job log. Otherwise, check with the author of the identified exit to determine the reason for the error.

IHC01030E DBRC not operational

Explanation: DBRC was requested; however, DBRC initialization failed.

System action: Processing stops.
User response: Keep the output listing and contact IBM Software Support.

IHC01031I DBRC subsystem nnnnn signon/signoff successful/failed

Explanation: This message is displayed at the start or the end of the DBRC connection. nnnnn indicates the specified subsystem name.

System action: None.
User response: If no request to DBRC is required, this message might be appropriate. However, if subsequent errors are displayed, the reason might be an error in DBRC connection.

IHC01032E HALDB master DBD nnnnn is not in the RECON

Explanation: The master DBD definition of a HALDB is kept in the RECON. A HALDB DBD named nnnnn was requested. The RECON does not have a record for this DBD.

System action: Processing stops.
User response: Correct the DBD name if this is the error. Otherwise, you might be using the wrong RECON.

IHC01033E DBD nnnnn is not HALDB

Explanation: The DBD nnnnn is expected to be a HALDB DBD. However, this is not the case.

System action: Processing stops.
User response: You might be using the wrong DBDLIB. If not, you must convert the specified DBD to HALDB.

IHC01034E Partition nnnnn is not in master DBD dddd

Explanation: Partition nnnnn was requested. However, nnnnn is not defined within its master DBD dddd.

System action: Processing stops.
User response: Either the partition name or the DBD name is wrong. You might also be using the wrong RECON.

IHC01035I Partition pppp authorized nnn

Explanation: Partition pppp has been authorized with DBRC. nnn is the authorization scope. It may be read, read exclusive, update, or exclusive.

System action: The DBRC authorization for the indicated partition has been performed.
User response: None. This message is informational.

IHC01036E Partition pppp not authorized, reason is nn

Explanation: Partition pppp could not be authorized with DBRC. The reason code nn can be found in message DFS047I.

System action: Processing stops.
User response: Make the database available by issuing the appropriate IMS commands.
IHC01037I  Partition pppp authorization released

Explanation:  Partition pppp is no longer authorized for the current function.

System action:  None.

User response:  None. This message is informational.

IHC01038E  Error during START of ISPF, reason code is xx

Explanation:  The ISPF application failed during startup. The code can be 20 to 24.

System action:  The application terminated.

User response:  Apply the appropriate action based on the reason code:

- Reason code 20: A previous error occurred.
  Save the trace screen and contact IBM Software Support.
- Reason code 21: A trace screen was displayed.
  Save the trace screen and contact IBM Software Support.
- Reason code 22 to 24: An internal error occurred.
  Contact IBM Software Support.

IHC01039E  orphaned split data is mnn

Explanation:  When IMS Online Reorganization Facility scanned the primary database to build a shadow copy of the primary index, it detected that there were data portions of split root segments that did not have the corresponding segment portions.

System action:  The IMS Online Reorganization Facility job continues.

User response:  The number of orphaned split data segments should be small. If you receive this message and the number that is displayed as mnn is 10 or more, it indicates that the primary database might have a problem. Check the primary database.

IHC01040I  Data set dsname not found

Explanation:  Dynamic allocation returned an error when allocating the data set named dsname. The data set was not found.

System action:  If the data set is essential, the process terminates with error.

User response:  The data set name comes from the MDALIB or from the RECON. Correct the data set name in those places and resubmit the job.

IHC01041I  Data set dsname in use

Explanation:  Dynamic allocation returned an error when allocating the data set. The data set is allocated in an exclusive state by a different job.

System action:  The process terminates.

User response:  Wait until the other job releases the data set and resubmit your job.

IHC01042E  Subtask task id has terminated abnormally.

Explanation:  The functions of IMS Online Reorganization Facility run as subtasks. Some of them might be in parallel. The subtask did not complete normally.

System action:  The process is terminated. The final abend code will be U999.

User response:  Collect the job output, including the dump, and contact IBM Software Support.

IHC01043E  Abend code is code

Explanation:  This message follows IHC01042E to display the abend code.

System action:  The process terminates.

User response:  Collect the job output, including the dump, and contact IBM Software Support.

IHC01044E  Unexpected end of service task task

Explanation:  The service task task terminated unexpectedly.

System action:  The process terminates.

User response:  Collect the job output, including the dump, and contact IBM Software Support.

IHC01045I  Return code is nnnn

Explanation:  The command ended with return code nnnn.

System action:  None.

User response:  None. This message is informational.

IHC01046E  Data set nnnn is migrated

Explanation:  The data set named nnnn could not be accessed.

System action:  Processing stops.

User response:  Recall the prior data set and rerun it.
Partition selection module not found

Explanation: The PSEXIT command could not find the specified partition selection exit.

System action: Processing stops.

User response: Provide the library containing the exit in the STEPLIB.

Partition selection size set to 2048MB

Explanation: Simulation, conversion, or maintenance of a HALDB was requested, specifying a fixed number of target partitions. However, one of the following conditions occurred:

ANALYZE
The size of the partitions would exceed 8 GB.

CONVERT
The size of the partitions would exceed 4 GB.

MAINTAIN
- For a VSAM or OSAM HALDB whose maximum data capacity is 4 GB, the size of the partitions would exceed 4 GB.
- For an OSAM HALDB whose maximum data capacity is 8 GB, the size of the partitions would exceed 8 GB.

System action: The selection of a fixed number of partitions is ignored. Processing continues with partition size set to 2 GB. This increases the number of partitions beyond the requested number.

User response: None. The simulation, conversion, or maintenance is valid. It is not necessary to restart with a different parameter.

Pointer error detected. Index must be rebuilt.

Explanation: The index pointer healer detected that an index EPS pointer requires healing. The corresponding ILE (the record in the target partition ILDS) was not found.

System action: Index pointer healer stops.

User response: Rebuild the invalid index or ILDS. If the index is invalid, rebuild it using either recovery or a PSINDEX rebuild tool. If the ILDS is invalid, rebuild it using the ILDS tool.

DD statement ddbname not found in JCL

Explanation: The DD statement named ddbname is required. It is not present in the JCL.

System action: Processing terminates.

User response: Provide the specified DD statement and resubmit the job.

HD unload file is from HALDB database

Explanation: The PSEXIT command is using an unload file which is already from a HALDB.

System action: Processing stops.

User response: Provide an unload file from a full-function database.

Error from Sort

Explanation: Standard sort is used. The sort returned with an error.

System action: Processing terminates.

User response: Collect the sort job output, including the dump, and contact IBM Software Support.

VSAM func error DD=ddname, RC=rc, RPL feedback=fdbk.

Explanation: The indicated VSAM function experienced an error.

System action: Processing terminates.

User response: Collect the VSAM job output, including the dump, and contact IBM Software Support.

Key table xxxxx not in project KSDS

Explanation: This is a should-not-occur situation.

System action: Processing stops.

User response: Keep the output listing and contact IBM Software Support.

Errors in control cards.

Explanation: This message is preceded by other messages that identify errors in control statements.

System action: Processing terminates.

User response: Correct control statement errors and resubmit the job.

Project KSDS is invalid

Explanation: The project KSDS has been provided through the IHCPROJ DD statement. This data set does not contain the required project information.

System action: Processing stops.

User response: Keep the output listing, and contact IBM Software Support. This JCL has been created by a conversion or maintenance project.
IHC01057E  Data set dsname ignored
Explanation: The IDCAMS rebuild process is attempting to retrieve the data set attributes from existing data sets. The specified data set name dsname could not be found.
System action: No IDCAMS statements are created for the target file.
User response: You must manually allocate the file that is to be copied.

IHC01058I  No pointer to heal in file ddname
Explanation: The pointer healer function was requested. The file indicated that its DD name did not have a pointer to heal.
System action: None.
User response: None. This message is informational.

IHC01059I  File ddname has nnn1 pointer. nnn2 needed healing.
Explanation: The pointer healer function was requested. The index file ddname has nnn1 pointer. However, only nnn2 pointer needed healing.
System action: None.
User response: None. This message is informational.

IHC01060I  Total records in index xxxx are nnnn
Explanation: The message displays the number of records in index xxxx.
System action: None.
User response: None. This message is informational.

IHC01061I  Initial load for DBD dbd
Explanation: The specified DBD is loading the database using a PROCOPT=L PSB.
System action: None.
User response: None. This message is informational.

IHC01062I  Loading index DBD dbd
Explanation: While the primary database is loaded (see 01061I), the secondary index dbd is also loaded.
System action: None.
User response: None. This message is informational.

IHC01063I  Loading index file ddname
Explanation: The PSINDEX file ddname is loaded as part of loading the secondary index.
System action: None.
User response: None. This message is informational.

IHC01064I  Number of records in file ddname is nnn
Explanation: The PSINDEX file ddname has been loaded. The number of records in this file is nnn.
System action: None.
User response: None. This message is informational.

IHC01065I  Application program pgm started/ended
Explanation: The application program pgm is participating in the PROCOPT=L support function. The message shows the start and the end of the program.
System action: None.
User response: None. This message is informational.

IHC01066I  PSB is psb
Explanation: The message shows the PSB used by this application.
System action: None.
User response: None. This message is informational.

IHC01067I  Application program return code is nnnn
Explanation: The application ended with the specified return code.
System action: None.
User response: None. This message is informational.

IHC01068I  IMS version is vers
Explanation: The message shows the IMS version that is currently being used.
System action: None.
User response: None. This message is informational.

IHC01069I  DDname ddn is not part of the DBD dbd
Explanation: The DD name ddn was found in the MDA member, but does not belong to the given DBD.
System action: None required; however, you might want to check the DBD to ensure that all other members belong there.
User response: None. This message is informational.
IHC01070E  DBD dbd is OSAM, but file allocated is VSAM
Explanation: During data set verification, a mismatch was found between the DBD type and the file type.
System action: Processing stops.
User response: Verify that both your MDA library and your DBD library are correct.

IHC01071E  DBD dbd is VSAM, but file allocated as OSAM
Explanation: During data set verification, a mismatch was found between the DBD type and the file type.
System action: Processing stops.
User response: Verify that both your MDA library and your DBD library are correct.

IHC01072E  File ddn is OSAM/KSDS/ESDS but should be OSAM/KSDS/ESDS
Explanation: The DD name ddn has an invalid file type.
System action: Processing stops.
User response: Assign the correct data set to the file indicated by ddn.

IHC01073I  IMS region controller started or ended
Explanation: This message is issued to indicate the start or the end of an IMS Batch region.
System action: None.
User response: None. This message is informational.

IHC01074I  Sorting index file ddn
Explanation: This part of the initial load support function. The PSINDEX file ddn is now being sorted and created.
System action: None.
User response: None. This message is informational.

IHC01075E  Database record key out of range
Explanation: The current function unloads a DBD participating in a user-partitioning conversion, when multiple identical DBDs are converted to one HALDB. The high-level qualifier key of each DBD has been specified. The unload function of this DBD now detects that a key is not within its boundaries.
System action: Processing stops.
User response: Begin the boundary selection process again. The initial boundary selection might be incorrect.

IHC01076I  Key: kkkkkkkkkkkkkkk
Explanation: This message is related to IHC01075E. It shows the key in process.
System action: Processing stops.
User response: Begin the boundary selection process again. The initial boundary selection might be incorrect.

IHC01077I  Low key: kkkkkkkkkkkkkkk
Explanation: This message is related to IHC01075E. It shows the lowest key that this DBD is allowed to accept.
System action: Processing stops.
User response: Begin the boundary selection process again. The initial boundary selection might be incorrect.

IHC01078I  High key: kkkkkkkkkkkkkkk
Explanation: This message is related to IHC01075E. It shows the highest key that this DBD is allowed to accept.
System action: Processing stops.
User response: Begin the boundary selection process again. The initial boundary selection might be incorrect.

IHC01079I  HSSR region controller started or ended.
Explanation: This message is issued during the unload phase. HSSR is used to assist the unload function.
System action: None.
User response: None. This message is informational.

IHC01080I  Start of ddnname listing.
Explanation: The specified file is being listed. The possible DD names are DFSVSAMP and DFSHALDB.
System action: None.
User response: None. This message is informational.

IHC01081I  End of ddnname listing
Explanation: This message indicates that the listing of the specified file has ended.
System action: None.
User response: None. This message is informational.

IHC01082I  Unloading DBD dbd
Explanation: The database dbd is currently being unloaded.
System action: None.
User response: None. This message is informational.
IHC01083I  Loading DBD dbd
Explanation:  The database dbd is currently being loaded.
System action:  None.
User response:  None. This message is informational.

IHC01084I  mmm database records read [FINAL]
Explanation:  This is a progress message that is issued every 250,000 database records (root segments). The indication of FINAL shows the total amount of database records.
System action:  None.
User response:  None. This message is informational.

IHC01085E  Error in partition part part, data set ddn
Explanation:  This message is accompanied by 01086E or 01087E.
System action:  Processing stops.
User response:  Keep the output listing, and contact IBM Software Support.

IHC01086E  Expected partition part, found part
Explanation:  The partitioning scheme is out-of-order.
System action:  Processing stops.
User response:  Keep the output listing, and contact IBM Software Support.

IHC01087E  Status code cc received
Explanation:  The status code cc was received as a response from an IMS call.
System action:  Processing stops.
User response:  The status code may indicate an unavailable database. If this is the case, make the database available and rerun. In all other cases, keep the output listing and contact IBM Software Support.

IHC01088E  Utility terminates with error
Explanation:  An error condition has occurred. Previous messages have been issued to explain the error.
System action:  Processing stops.
User response:  Proceed as instructed in previous messages that were issued to explain the error.

IHC01089I  Analyzing DBD dbdname
Explanation:  The specified DBD is analyzed.
System action:  None.
User response:  None. This message is informational.

IHC01090E  Last partition is already empty
Explanation:  The last partition does not contain any data.
System action:  The request to add an empty partition is discarded.
User response:  None.

IHC01091E  MDA dbdname DDname dname not found
Explanation:  The MDA member dbdname does not have the specified DD defined.
System action:  The process terminates.
User response:  Correct the specified MDA member and resubmit the job.

IHC01092I  DBDGEN complete for DBD dbdname
Explanation:  The specified DBD has been compiled and linked.
System action:  None.
User response:  None. This message is informational.

IHC01093E  DDname dname not found
Explanation:  The specified DD statement was not found in the JCL.
System action:  The process terminates.
User response:  Add the specified DD statement and resubmit the job.

IHC01094E  DBD dbdname has external logical relationship.
Explanation:  The specified DBD has a logical relationship to a different DBD. This type of relationship is not supported by IMS Online Reorganization Facility therefore, the specified DBD cannot be reorganized with IMS Online Reorganization Facility.
System action:  Processing terminates.
User response:  None.
IHC01095E  DBD dbdname has HALDB OLR active.

Explanation: IMS OLR is currently reorganizing this partition. IMS Online Reorganization Facility cannot run with OLR active. This situation can also occur when OLR is paused. The indication is that both the A-J and M-V data sets are active.

System action: Processing terminates.

User response: Reorganize a different partition or wait until OLR completes to resubmit your job.

IHC01096E  DBD dbdname is HALDB with logical relationship

Explanation: The specified DBD is a HALDB and also has a logical relationship. IMS Online Reorganization Facility can process a HALDB with a logical relationship only if all partitions are processed in the IMS Online Reorganization Facility job.

System action: Processing terminates.

User response: To process all partitions in one IMS Online Reorganization Facility job, specify the PARTITION(*) keyword and resubmit the job.

IHC01097E  DBD dbdname is not registered.

Explanation: The specified DBD is not in DBRC. IMS Online Reorganization Facility requires that the DBDs must be registered. This requirement also applies to the index DBDs.

System action: Processing terminates.

User response: Register this DBD and all its index DBDs and resubmit the job.

IHC01098E  DBD dbdname has conflicting DBRC definitions.

Explanation: The specified DBD does not match its DBRC definition.

System action: Processing terminates.

User response: Compare your DBD to a LIST.RECON of that DBD. Correct the errors and resubmit the job.

IHC01099E  DBD dbdname has dbrc status.

Explanation: The specified DBD has a DBRC status that does not allow the processing of this DBD.

System action: Processing terminates.

User response: Correct the DBRC exception and resubmit the job.

IHC01100I  process started in sub address space

Explanation: The specified process started in a dependent address space.

System action: None.

User response: None. This message is informational.

IHC01101I  process ended in sub address space, return code code

Explanation: The specified process ended in a dependent address space.

System action: None.

User response: None. This message is informational.

IHC01102E  Sub address space terminated abnormally

Explanation: The dependent address space did not end normally.

System action: The primary address space terminates. The messages will be displayed in the message file.

User response: This message is accompanied by another message that indicates the cause of the error. If you cannot resolve the problem, contact IBM Software Support.

IHC01103E  DBD dbdname does not have DBDS dname in DBRC

Explanation: The specified DBD does not have the DBDS record defined in DBRC. The DBD definition in DBRC is incomplete.

System action: The process terminates.

User response: Add the specified DBDS to the RECON. Use the INIT.DBDS function of the DBRC utility.

IHC01104E  Dynamic allocation failed, DSN dsname

Explanation: The data set that is indicated by dsname is probably a database data set. The data set name was obtained from the RECON. Additional messages explain the reason for the allocation failure.

System action: Processing terminates.

User response: Correct the reason for the allocation failure and resubmit the job.

IHC01105E  DSN dsname is too long.

Explanation: The length of the specified database data set name exceeds the maximum allowable length. Database data set names are appended with ".S", which means that database data set names can be a maximum of 42 characters.
System action: Processing terminates.
User response: The data set names must be shortened. If they cannot be shortened, the specified DBD cannot be processed by IMS Online Reorganization Facility.

IHC01106I Allocating the shadow files.
Explanation: The shadow files are being defined and allocated. This message precedes the IDCAMS list from the allocation.
System action: None.
User response: None. This message is informational.

IHC01107E Online subsystem imsid not available.
Explanation: The RECON subsystem records show this subsystem as an online IMS subsystem. However, IMS Online Reorganization Facility is not active in this subsystem.
System action: Processing terminates.
User response: Add the IMS Online Reorganization Facility load library to the online system STEPLIB. Verify that IMS Online Reorganization Facility has been properly set up.

IHC01108I AOI interface connected to imsid.
Explanation: The IMS Online Reorganization Facility job established operator command capability with the IMS subsystem that is named in the message.
System action: None.
User response: None. This message is informational.

IHC01109I No online subsystem active.
Explanation: No online IMS subsystem was active at this time.
System action: The process continues.
User response: None. This message is informational.

IHC01110E HIDAM root segment has compressed key.
Explanation: The root segment of a HIDAM or PHIDAM database is compressed. The compression is not only DATA, but also KEY. IMS Online Reorganization Facility does not allow compressed root keys for HIDAM databases.
System action: Processing terminates.
User response: The database must not have key compression. Remove the key compression by using the offline utilities and perform a DBD change between unload and reload.

IHC01111E Software for product not available.
Explanation: The named product is required for IMS Online Reorganization Facility to do the reorganization.
System action: Processing terminates.
User response: Add the program library for the specified product to the STEPLIB. Make sure that the library is APF authorized.

IHC01112I Reloading DD ddname DSN dsname
Explanation: The target data sets that are identified in this message will be used to load the database.
System action: None.
User response: None. This message is informational.

IHC01113E OPEN failed for DDname ddname.
Explanation: The OPEN operation failed for the DD name indicated by ddname.
System action: Processing terminates.
User response: Collect the output and contact IBM Software Support.

IHC01114I Secondary index build started or ended.
Explanation: This message indicates that a secondary index build process has either started or has ended.
System action: None.
User response: None. This message is informational.

IHC01115I DDname ddname found in JCL. It is deallocated.
Explanation: The specified DD statement is allocated internally.
System action: None.
User response: Remove the DD statement from the JCL and resubmit the job.

IHC01116I Shared index written with index DBD dbdname.
Explanation: The shared secondary index will be assigned to the first DBD.
System action: None.
User response: None. This message is informational.

IHC01117I Pre Reorganization Utility started or ended
Explanation: This message indicates that the prereorganization utility has either started or has ended.
IHC01118I  Prefix Resolution started or ended.
Explanation: This message indicates that the prefix resolution has either started or has ended.
System action: None.
User response: None. This message is informational.

IHC01119I  Prefix Update started or ended.
Explanation: This message indicates that the prefix update has either started or has ended.
System action: None.
User response: None. This message is informational.

IHC01120E  Primary DBD dbdname is NONRECOV.
Explanation: The primary DBD is defined as NONRECOV in DBRC. This is not permitted.
System action: Processing terminates.
User response: Change the DBD to RECOV in DBRC and resubmit the job.

IHC01121E  DBD dbdname is not supported for this function.
Explanation: The specified DBD has an unsupported DBD type. HSAM, DEDB, or Index DBDs are not supported.
System action: Processing terminates.
User response: The specified DBD cannot be reorganized with this tool.

IHC01122I  IC ddname specified but ignored. DBD is NONRECOV.
Explanation: The specified database DDNAME was specified to be image copied. However, the DBD is defined as NONRECOV. Therefore, an image copy is unnecessary.
System action: None.
User response: If you need an image copy, run the offline utility.

IHC01123E  IC ddname is not a valid DDname.
Explanation: The ICDDN keyword specifies a database DD statement. Neither the Primary DBD nor its index DBDs contain this DD.
System action: Processing terminates.
User response: Correct the ICDDN keyword and resubmit the job.

IHC01124E  IC ddname DDname icddn specified, but not in JCL.
Explanation: The ICDDN keyword specifies an image copy DD icddn for the database DD ddname. The icddn was not found in the JCL.
System action: Processing terminates.
User response: Provide a DD statement for the image copy and resubmit the job.

IHC01125E  Duplicate DDname ddname in ICDDN keyword.
Explanation: The ICDDN keyword is used to describe the image copies. The specified DDNAME is used more than once.
System action: Processing terminates.
User response: Provide a unique DD statement for each image copy and resubmit the job.

IHC01126E  Inconsistent shared index dbdname DBD. Reason rsn.
Explanation: Shared secondary indexes are being used. They must point to the same primary DBD to be supported by IMS Online Reorganization Facility. Additionally, all shared secondary indexes must have the same DBDS in DBRC.
System action: Processing terminates.
User response: IMS Online Reorganization Facility does not support shared secondary indexes that point to different primary DBDs.

IHC01127E  No IC1 found for ddname.
Explanation: The ICDDN keyword for the database DD ddname does not have a primary image copy specified.
System action: Processing terminates.
User response: You cannot specify a secondary IC without a primary IC. Change the ICDDN keyword and resubmit the job.

IHC01128E  IC ddname DSN for icddn too long.
Explanation: Dynamic allocation for image copy data sets is being used. The generated ddname is too long.
System action: Processing terminates.
User response: Create a shorter IC data set name and resubmit the job.
IHC01129E  IC DDname ddname is for database.
Explanation: The ICDDN keyword specified a database as target for image copy.
System action: Processing terminates.
User response: Provide a unique file for the image copy and resubmit the job.

IHC01130E  DSname for ddname is the same as for ddname
Explanation: The image copy data sets that are identified by the ICDDN keyword have different DD statements, but have the same data set name.
System action: Processing terminates.
User response: Provide different data sets for the image copies and resubmit the job.

IHC01131E  dsname for icddn is the same as for database ddname.
Explanation: The data set name that was used for an image copy is a database data set name.
System action: Processing terminates.
User response: Provide a different data set name for the image copy and resubmit the job.

IHC01132E  GDG gdgbase not defined.
Explanation: GDG was specified as a data set name for image copy. However, the GDG base does not exist.
System action: Processing terminates.
User response: Create the GDG base and resubmit the job.

IHC01133E  Index DBD dbdname for primary DBD dbdname in error.
Explanation: The index DBD that is indicated by dbdname for the primary DBD that is indicated by dbdname is in error.
System action: Processing terminates.
User response: Collect the output, the specified DBD source, and contact IBM Software Support.

IHC01134E  NEWDBD DD not in JCL.
Explanation: The NEWDBD keyword was specified in the control statements. The specified DD statement has not been provided.
System action: Processing terminates.
User response: Provide the DD statement that specifies the library that contains the changed DBDs and resubmit the job.

IHC01135E  No DBD specified.
Explanation: No DBD keyword was specified in the control statements.
System action: Processing terminates.
User response: Provide a DBD and resubmit the job.

IHC01136E  Multiple REORG commands specified
Explanation: More than one REORG command was found in IHCSYSIN.
System action: Processing terminates.
User response: Run each REORG as separate job step.

IHC01137E  DBD dbdname compare error. Reason is: rsn
Explanation: The NEWDBD keyword was specified in the control statement. The old and new DBD are compared. The new DBD contains changes that are not supported. The reason text provides an explanation.
System action: Processing terminates.
User response: The changes that can be made to the DBD are restricted when IMS Online Reorganization Facility is used. Use offline utilities for this reorganization.

IHC01138E  Primary DBD dbdname is not in NEWDBD.
Explanation: The NEWDBD keyword was specified in the control statements. However, the library does not have the primary DBD in it.
System action: Processing terminates.
User response: If you intended to make a DBD change, move your changed DBD to the file that is identified by the NEWDBD keyword.

IHC01139E  Post reorganization processing failed in phase: phase
Explanation: The takeover process failed.
System action: Processing terminates.
User response: Use the RESTART parameter to restart the takeover process. The database is still in prohibit authorization status. Do not change the prohibit authorization status. The takeover phase must complete. The takeover process will restart where the last operation stopped.
IHC01140I  Attempting to restart in phase phase
Explanation:  This informational message is issued at the point that the RESTART function continues the takeover process.
System action:  None.
User response:  None. This message is informational.

IHC01141I  Last phase completed was phase.
Explanation:  This message is issued during restart. The last phase completed is indicated by phase.
System action:  None.
User response:  None. This message is informational.

IHC01142E  Neither data set dsname or dsname was found.
Explanation:  During the takeover phase, the data sets are to be renamed. In this case, none of the data sets were found.
System action:  The process terminates.
User response:  This error typically occurs during a RESTART process. Because a database data set has been deleted, the database must be recovered.

IHC01143E  Index dbname not in NEWDBD library
Explanation:  A secondary index has been changed, but the new DBD has not been provided.
System action:  The process terminates.
User response:  Correct the changed DBD and resubmit the job.

IHC01144E  New primary DBD has external logical relationships.
Explanation:  The NEWDBD keyword has been specified. The new DBD added external logical relationships which are not supported.
System action:  Processing terminates.
User response:  Use offline utilities for this process.

IHC01145I  NEWDBD library now in effect.
Explanation:  This message is issued after unload has completed.
System action:  None.
User response:  None. This message is informational.

IHC01146I  Prior job jobname at time for DBD dbname in phase phase
Explanation:  This message indicates a pending restart. A previous job ran at the specified time for the specified DBD and stopped at the specified phase.
System action:  None.
User response:  None. This message is informational.

IHC01147I  (param) specified for this execution.
Explanation:  The RESTART function is used.
System action:  None.
User response:  None. This message is informational.

IHC01148I  DBD dbname has non-unique segments
Explanation:  The DBD has segment types that do not have a sequence field or that have a non-unique sequence field.
System action:  None.
User response:  None. This message is informational.

IHC01149E  IMSACB DD required with NEWDBD and ONLINECHANGE(Y).
Explanation:  The new DBD will be propagated to the online system, which requires the staging library ACBLIB.
System action:  Processing terminates.
User response:  Add the ACBLIB DD to the JCL and resubmit the job.

IHC01150E  Invalid attrib for ddbname DD
Explanation:  The attributes for a KSDS are invalid.
System action:  The process terminates.
User response:  Correct the attributes for the shadow data sets and resubmit the job.

IHC01151I  Temporary RECON created or deleted.
Explanation:  A HALDB partition was specified. A temporary RECON was created to define the partitions and the shadow data set names. This RECON is used for the reorganization process.
System action:  None.
User response:  None. This message is informational.
**IHC01152E**  NEWDBD specified with HALDB database.

**Explanation:** IMS Online Reorganization Facility does not support DBD changes for HALDB databases.

**System action:** Processing terminates.

**User response:** Use an offline utility and reorganize all partitions.

**IHC01153E**  *ddname* DD cannot be a temporary data set.

**Explanation:** A temporary data set was specified for the indicated DD.

**System action:** Processing terminates.

**User response:** Specify a permanent data set and resubmit the job.

**IHC01154E**  Unable to load module *modname*.

**Explanation:** The specified module *modname* was not found.

**System action:** Processing terminates.

**User response:** Make the user exit available in the JCL and resubmit the job.

**IHC01155I**  Rebuilding primary index.

**Explanation:** A HIDAM or PHIDAM database is being used. The primary index is created in the shadow data set.

**System action:** None.

**User response:** None. This message is informational.

**IHC01156E**  Error creating interim LOG

**Explanation:** IMS Online Reorganization Facility was unable to allocate a data set for use as a batch log during the apply process. The allocation parameters of this interim data set are based on the IEFRDER DD statement in the JCL.

**System action:** The IMS Online Reorganization Facility job terminates.

**User response:** Review the preceding messages to determine why the allocation failed. Resolve the allocation problems and resubmit job.

**IHC01157E**  Shadow data set for *ddname* too small

**Explanation:** The space allocation for the specified predefined shadow data set was insufficient.

**System action:** The process terminates.

**User response:** Increase the size of the specified data set and resubmit the job.

**IHC01158E**  Partition keyword required for HALDB

**Explanation:** A HALDB was specified; however, no partition was specified. You must specify the partition that you want reorganized.

**System action:** The process terminates.

**User response:** Specify a partition with the PARTITION IHCSYSIN DD statement keyword and resubmit the job.

**IHC01159E**  Invalid component DSN *dsname*

**Explanation:** The specified *dsname* is a component name of a VSAM cluster. The dsname is too long to use to create shadow data sets.

**System action:** The process terminates.

**User response:** Shorten the data set name and resubmit the job.

**IHC01160E**  Primary DSN *dsname* and shadow have different attributes

**Explanation:** Preallocated shadow data sets were used. The data set attributes of the shadow data sets must match the original data sets. The specified data set attributes and the attributes of the shadow data sets do not match.

**System action:** The process terminates.

**User response:** Provide the correct attributes. If this data set is an index data set, you probably specified the wrong keylen, key offset, or record length and resubmit the job.

**IHC01161E**  HALDB partition *name* is on M-V data sets

**Explanation:** The partition that is identified by *name* is currently on the M-V data sets. IMS Online Reorganization Facility was unable to create a copy of the data sets on the shadow file.

**System action:** IMS Online Reorganization Facility processing terminates.

**User response:** Use an offline reorganization or another IMS Online Reorganization to switch the active data sets, to make them the A-J data sets, and resubmit the job.

**IHC01162E**  Invalid subparameter for PTRCHECK keyword

**Explanation:** The PTRCHECK keyword has a Y or N in the first position and up to four additional parameters. The specified parameters are invalid.

**System action:** The process terminates.

**User response:** Provide valid parameters for the
IHC01163E  Pointer error detected
Explanation: IMS High Performance Pointer Checker detected a pointer error during the image copy process of the newly reorganized shadow database data sets.
System action: Processing terminates.
User response: Collect the job output, including the IMS High Performance Pointer Checker output, and contact IBM Software Support.

IHC01164E  Dynamic allocation failed, DD ddname
Explanation: IMS Online Reorganization Facility was unable to dynamically allocate a required DD. Messages that are issued prior to this one explain the reason for the allocation failure.
System action: Processing terminates.
User response: Correct the reason for the allocation failure and resubmit the IMS Online Reorganization Facility job.

IHC01165E  Invalid PARTITION specified
Explanation: A name was entered in the PARTITION keyword, but the partition name is not part of the master DBD.
System action: Processing terminates.
User response: Correct the partition name and resubmit the IMS Online Reorganization Facility job.

IHC01166I  DBD dbdname has internal logical relationship
Explanation: This is an informational message that informs you that the dbdname that you specified contains at least one internal logical relationship.
System action: Processing continues. IMS Online Reorganization Facility performs any prefix resolution and update processing that is required for the process that is being performed.
User response: None. This message is informational.

IHC01169E  More than 255 RAPS
Explanation: More than 255 RAPS have been specified for a partition, which exceeds the limit in IMS.
System action: The process terminates.
User response: Specify a smaller amount of RAPS.

IHC01170E  Byte limit too large
Explanation: The byte limit exceeded the maximum size, which is 16 MB.
System action: The process terminates.
User response: Specify a smaller byte limit.

IHC01171E  DBD dbdname has logical relationship
Explanation: The purpose of the current function is to convert an entire DBDLIB. However, this function does not allow for logically related DBDs.
System action: The DBD is excluded from this process.
User response: Use a different method to process this DBD.

IHC01172I  DBD dbdname was excluded, reason is: reason
Explanation: The current function is to convert an entire DBDLIB. The specified DBD will be excluded from this function due to one or more of the following reasons:
- DBD construct error
- DBD is not supported
- DBD is HALDB
- DBD has logical relation
- Primary index is missing
- Secondary index is missing
- Primary database is missing
- MDA library member missing
- MDA DDname not in member
- Data set not cataloged
- Index DBD in error
System action: The DBD is excluded from the conversion.
User response: Make the changes that are indicated by the reason and resubmit the job. Alternately, you can run the conversion as is, and work on those DBDs that failed at a later time.

IHC01173I  DBD dbdname is an index
Explanation: The specified DBD is a PSINDEX or a secondary index. The selected function does not allow for an index DBD.
System action: The process terminates.
User response: Specify a primary DBD.
IHC01174E Orphaned split segment found

Explanation: When scanning the primary database to build a shadow copy of the primary index, IMS Online Reorganization Facility detected that there was a root that had been split, and the pointer to the data portion was invalid.

System action: Processing terminates.

User response: When this message is issued, it likely indicates that there is a pointer error in the original database data set. Run IMS High Performance Pointer Checker or an equivalent tool against the original database to determine the pointer errors. Resolve the pointer errors and resubmit the IMS Online Reorganization Facility job.

IHC01175E All partitions required for NONRECOV DBD

Explanation: A nonrecoverable HALDB was specified in the DBD keyword. The command that was specified requires that IMS Online Reorganization Facility change the database to be recoverable during the IMS Online Reorganization Facility processing. IMS Online Reorganization Facility can be run only at the master DBD level for a HALDB, so all partitions must be included in the IMS Online Reorganization Facility processing.

System action: Processing terminates.

User response: Either remove the PARTITION keyword from the command so that all partitions are processed, or manually change the DBD to recoverable and resubmit the IMS Online Reorganization Facility job.

IHC01176E Data set found empty

Explanation: This message is preceded by IHC01085, which identifies the primary database DDname and DName. One record containing the partition ID and the reorg number should be found.

System action: The process terminates.

User response: Verify the load process of that partition. Run the partition initialization, and reload for that partition.

IHC01177E Shadow data set dsname defined as NOREUSE but data set is not empty

Explanation: The specified data set must be empty.

System action: The process terminates.

User response: Either specify the data set as REUSE or DELETE/DEFINE it prior to this process.

IHC01178E Dynamic allocation limit reached. Specify larger DYNAMNBR.

Explanation: The maximum dynamic allocation limit has been reached.

System action: The process terminates.

User response: Specify the DYNAMNBR parameter on your EXEC statement. A value of DYNAMNBR=999 typically resolves this problem.

IHC01179E VIO has been allocated to ddname

Explanation: The current process does not support VIO data sets.

System action: The process terminates.

User response: Specify a unit name that is not VIO. Use the IHCEUNIT member in SIHCSAMP file to locate other unit names that are non-VIO.

IHC01180E Sort work ddname has not enough disk space

Explanation: ddname cannot allocate enough sort work space. The system might be low on work space.

System action: The job fails.

User response: Override the DD with sufficient disk space so that the system can allocate enough sort work space for the DD.

IHC01181W ORF must take IC for ONLINE(Y) work. ICOFF(N) set.

Explanation: ICOFF(N) must be set to process an online database. IMS HALDB Toolkit changed the ICOFF keyword to ICOFF(N).

System action: ICOFF(N) is set and processing continues.

User response: None.

IHC01183E All partitions required for NOOSAM8G(Y)

Explanation: IMS HALDB Toolkit found the NOOSAM8G(Y) parameter on the MAINTAIN command but no PARTITION(*) parameter. You must specify PARTITION(*) when you specify the NOOSAM8G(Y) parameter.

System action: The process terminates.

User response: Either remove the NOOSAM8G(Y) parameter from the command or specify PARTITION(*).
IHC01184E NOOSAM8G(Y) cannot be specified for a database registered as NOOSAM8G in RECON

**Explanation:** The NOOSAM8G(Y) parameter cannot be specified for a database that is registered as NOOSAM8G in the RECON data sets.

**System action:** The process terminates.

**User response:** Remove the NOOSAM8G(Y) parameter from the command.

---

IHC01185E PARTSIZE must be less than 4096 for a database registered as NOOSAM8G in RECON

**Explanation:** The PARTSIZE value must be less than 4096 for a database that is registered as NOOSAM8G in the RECON data sets.

**System action:** The process terminates.

**User response:** Specify a value that is less than 4096 for PARTSIZE.

---

IHC01186E PARTSIZE must be less than 4096 when you specify NOOSAM8G(Y)

**Explanation:** The PARTSIZE value must be less than 4096 when you specify NOOSAM8G(Y).

**System action:** The process terminates.

**User response:** Either remove the NOOSAM8G(Y) parameter or specify a value that is less than 4096 for PARTSIZE.

---

IHC01188E DBRCOUT for a non-HALDB requires DSNPREF

**Explanation:** The database that is identified from the DBD parameter is a non-HALDB. When a non-HALDB is provided as the input for ANALYZEPART and the DBRCOUT parameter is specified, the DSNPREF parameter must also be specified.

**System action:** The process terminates.

**User response:** Specify the DSNPREF parameter and resubmit the job.

---

IHC01189E ddname is not APF authorized

**Explanation:** One or more libraries in the ddname concatenation are not APF authorized.

**System action:** The process terminates.

**User response:** APF authorize all the libraries that are concatenated to the ddname and rerun the job.

---

IHC01200E FROMDBD ddname1 TODBDD ddname2 compare failed: reason

**Explanation:** The merge process detected one or more errors. Potential reasons include:
- DBD names are identical
- Segment structure is different
- Segment definition is different
- DBD type is different
- DBD has secondary index
- DBD has logical relation
- DBD has different partition selection exits
- IHCPSEL0 is defined different

**System action:** The process terminates.

**User response:** None. Merging these two HALDBs is not possible.

---

IHC01201E DBD ddname is PSINDEX

**Explanation:** The specified DBD is an index DBD.

**System action:** The process terminates.

**User response:** A PSINDEX DBD cannot be specified for merging.

---

IHC01202E PGM name not found

**Explanation:** The specified program was not found.

**System action:** The process terminates.

**User response:** Correct the program name and resubmit the job.

---

IHC01203E PSB name not found

**Explanation:** The specified PSB was not found.

**System action:** The process terminates.

**User response:** Correct the PSB name and resubmit the job.

---

IHC01204E PSB not found or valid

**Explanation:** The specified PCB is not in the PSB or does not have PROCOPT=L.

**System action:** The process terminates.

**User response:** Correct the PSB/PCB parameter and resubmit the job.

---

IHC01205E DFSRESLB DD is required to MAINTAIN PSINDEX.

**Explanation:** When the MAINTAIN function processes PSINDEXes, the function uses DFSURGL0 (HD Reorganization Reload utility) in the DFSRESLB DD to reload databases. Because the DFSRESLB DD statement...
is not coded in the JCL, IMS HALDB Toolkit attempted to dynamically allocate the DFSRESLB DD statement using the IMS RESLIB library. However, the IMS RESLIB library containing IMS SVC modules was not found in the STEPLIB concatenation so IMS HALDB Toolkit did not allocate the DFSRESLB DD statement.

**System action:** Processing terminates.

**User response:** Specify the IMS RESLIB library containing IMS SVC modules to the STEPLIB DD statement or the DFSRESLB DD statement and rerun the job.

---

**IHC01301W Dynamic allocation failed, SVC 99 reason code ‘xxxx’**

**Explanation:** IMS Online Reorganization Facility was unable to dynamically allocate a data set. You can find information about the SVC 99 reason codes in the z/OS MVS Authorized Assembler Services Guide.

**System action:** If the data set is a required data set, then processing terminates. Otherwise, processing continues.

**User response:** Correct the reason for the allocation failure and resubmit the IMS Online Reorganization Facility job.

---

**Table 39. Message IHC01500W return codes and their meanings (continued)**

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>LOAD failed for required module. Verify that the complete IMS Online Reorganization Facility library is in the STEPLIB.</td>
</tr>
<tr>
<td>7</td>
<td>Unable to obtain ECSA.</td>
</tr>
<tr>
<td>8</td>
<td>Unsupported IMS release.</td>
</tr>
<tr>
<td>9</td>
<td>Logger exit initialization failed.</td>
</tr>
<tr>
<td>17</td>
<td>Unable to load the HRFYOFxx module or the HRFYOFxx module is not reentrant. Ensure that the IMS Online Reorganization Facility library is in the STEPLIB.</td>
</tr>
</tbody>
</table>

---

**IHC01500I Online Reorg Facility [initialization for insid | logger exit initialization] failed, RC=xxxx**

**Explanation:** During IMS control region startup, IMS Online Reorganization Facility was not initialized in the IMS subsystem, insid.

**System action:** IMS startup continues but IMS Online Reorganization Facility will be active for the IMS subsystem, insid, only when it is restarted.

**User response:** Review the previous error messages and try to correct the problem. If you are unable to correct the problem, contact IBM Software Support.

The following table lists the IHC01500W return codes and their meanings.

**Table 39. Message IHC01500W return codes and their meanings**

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Unable to locate SCD. Contact IBM Software Support.</td>
</tr>
<tr>
<td>3</td>
<td>Unable to locate IMS Online Reorganization Facility anchor. Logger exit initialization failed.</td>
</tr>
<tr>
<td>4</td>
<td>IEANTRT call failed. The error was returned from IEANTRT.</td>
</tr>
<tr>
<td>5</td>
<td>BLDL failed for the required module. Unable to load IMS Online Reorganization Facility module into ECSA. Verify that the complete IMS Online Reorganization Facility library is in the STEPLIB.</td>
</tr>
</tbody>
</table>

---

**IHC01501W DEBUG setup failed: reason, RC=xxxx**

**Explanation:** IMS Online Reorganization Facility was unable to start up additional debugging options. The reason specified explains why setup failed.

**System action:** Processing continues with DEBUG off.

**User response:** Report the reason and RC to IBM Software Support.

---

**IHC01502I Online Reorg Facility DEBUG status**

**Explanation:** This informational message indicates that the DEBUG services have been activated or stopped.

**System action:** None.

**User response:** None. This message is informational.
IHC01504E  Target IMS system insid is not active on this host

Explanation: An IMS Online Reorganization Facility maintenance utility was run on an MVS™ system and the target IMS subsystem, insid, is not active on that MVS system.

System action: The IMS Online Reorganization Facility job terminates.

User response: Resubmit the job on the same MVS system where the target IMS subsystem is active.

IHC01505E  Online Reorg Facility is not active in target IMS system

Explanation: An IMS Online Reorganization Facility maintenance utility was run but IMS Online Reorganization Facility is not active in the target IMS subsystem.

System action: The IMS Online Reorganization Facility job terminates.

User response: Verify that the target IMS subsystem is correct. IMS Online Reorganization Facility must initially be activated in an IMS subsystem with a restart of the IMS control region. Verify that this was done and that the IHC01500I message was issued to indicate that the IMS Online Reorganization Facility was initialized successfully.

IHC01506W  DEBUG status for IMS insid already state

Explanation: An IMS Online Reorganization Facility utility was run to set the DEBUG state. The DEBUG state for the IMS subsystem was already in the requested state.

System action: None. DEBUG status is left in previous state.

User response: Verify that the DEBUG status is in the appropriate state.

IHC01507I  DEBUG action request for IMS insid completed

Explanation: An IMS Online Reorganization Facility utility was run and has successfully changed the DEBUG state for an IMS subsystem.

System action: The requested DEBUG state is now in effect for the target IMS subsystem.

User response: None. This message is informational.

IHC01508W  Unable to install new maintenance: reason

Explanation: The IMS Online Reorganization Facility maintenance utility was unable to install new maintenance into an IMS online subsystem.

System action: The previous maintenance level is still in effect for the IMS subsystem.

User response: Correct the reason that new maintenance was unable to be installed and resubmit the job.

IHC01509W  IMS using ORF library datasetname, new maintenance being loaded from datasetname

Explanation: IMS Online Reorganization Facility maintenance utility is being executed with a different library than the IMS control region was initially started with.

System action: New maintenance is installed into the IMS control region from the different library.

User response: The next time the IMS subsystem is restarted it will activate IMS Online Reorganization Facility from the library in the control region STEPLIB. If you want the installation of the new maintenance level to be permanently installed into the IMS subsystem, the new maintenance level must either be copied into the STEPLIB data set for the IMS control region, or the library with the new maintenance level must be added to the STEPLIB of the IMS control region.

IHC01510I  Online Reorg Facility Vr.r maintenance successfully installed in insid

Explanation: This informational message indicates that a new maintenance level has successfully been installed into the insid online IMS subsystem.

System action: The new maintenance level is now in effect in the target IMS subsystem.

User response: None. This message is informational.

IHC01511E  Target member membername action reason

Explanation: An attempt to connect to the IMS Online Reorganization Facility or IMS Tools Online System Interface component in an IMS control region address space failed for the specified reason.

System action: The IMS Online Reorganization Facility job terminates.

User response: If the IMS subsystem is down, it must either be restarted or the SUBSYS record must be removed from DBRC. If the IMS subsystem is active, verify that either the IMS Online Reorganization Facility or IMS Tools Online System Interface
IHC01512I • IHC01518E

component was successfully initialized in that subsystem and that the component joined the same XCF group.

IHC01512I  Connected with target member membername

Explanation: This informational message indicates that the job has successfully connected to the corresponding member.

System action: Processing continues.

User response: None. This message is informational.

IHC01513E  ORF subtask failed RC=xxxx

Explanation: The IMS Online Reorganization Facility subtask in the IMS control region terminated unexpectedly.

System action: IMS Online Reorganization Facility processing in the IMS control region is no longer available.

User response: Report the problem to IBM Software Support. The IMS subsystem must be restarted to reactivate IMS Online Reorganization Facility in that IMS subsystem.

IHC01514I  capturetype capture for DBD(dbdname) action

Explanation: This informational message indicates that capturing log or change records for the DBD has either been started or has just ended.

System action: If capture has been activated, the captured records will begin being sent to the remote IMS Online Reorganization Facility utility. If capture has been stopped, no more change records for the DBD will be sent to the IMS Online Reorganization Facility utility.

User response: None. This message is informational.

IHC01515W  message_type from member membername

Explanation: An unexpected XCF message was received from the corresponding member. message_type shows one of the following values:

NEWMBR
The member that newly joined the XCF group. This type of message is typically received when an IMS subsystem is started during the reorganization.

LOSTCONN
The connection to the member was lost. This type of message is typically received when an IMS subsystem terminates during the reorganization or the reorganization job abnormally terminates.

System action: If the IMS Online Reorganization Facility job receives the message, that job terminates. If the message is received in the IMS control region, capturing changes for the remote IMS Online Reorganization Facility utility ends.

User response: None.

IHC01516E  XCF buffer limit exceeded

Explanation: The maximum amount of storage for incoming XCF messages has been exceeded; no further incoming XCF messages can be received.

System action: The IMS Online Reorganization Facility job terminates.

User response: The maximum amount of storage that is used for incoming XCF messages is controlled by the XCFMAX parameter in the HRFSET parameter.

This situation is usually the result of the IMS Online Reorganization Facility utility's inability to process the number of captured change records from the IMS control regions. It might be the result of an extreme amount of online activity and of the IMS Online Reorganization Facility utility running at a priority at which it cannot obtain any CPU cycles to receive pending XCF messages.

Resubmit the IMS Online Reorganization Facility job at a higher priority and at a less active time, or set the XCFMAX value slightly higher. This situation might also occur if the IMS Online Reorganization Facility utility does not receive the pending messages when it should. This situation would likely be an outstanding WTOR or CPU loop.

IHC01517E  Error encountered during capturetype capture for DBD dbdname:
FUNC=function RC=xxxx RSN=xxxx

Explanation: The IMS Online Reorganization Facility was unable to capture a change record for a DBD.

System action: Record capture is terminated for the DBD. The IMS Online Reorganization Facility job is notified of the records that were being captured. The IMS Online Reorganization Facility job abends.

User response: Report the problem to IBM Software Support.

IHC01518E  command call failed: RC=xxxx RSN=xxxx

Explanation: The IMS Online Reorganization Facility utility encountered an error when issuing the IMS command, command.

System action: The IMS Online Reorganization Facility utility abends.

User response: Review the return and reason codes that are described in IMS Application Programming EXEC DLI Commands for CICS® and IMS.
<table>
<thead>
<tr>
<th>Message Code</th>
<th>Description</th>
<th>Explanation</th>
<th>System Action</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHC01519I</td>
<td>Number of records captured was XXX</td>
<td>Number of records captured was XXX.</td>
<td>None.</td>
<td>None. This message is informational.</td>
</tr>
<tr>
<td>IHC01520E</td>
<td>Calltype call ended with 'statuscode status code'</td>
<td>When applying captured changes to the reorganized shadow data sets, the IMS Online Reorganization Facility utility encountered an unexpected status code.</td>
<td>The IMS Online Reorganization Facility job abends.</td>
<td>Report the problem to IBM Software Support.</td>
</tr>
<tr>
<td>IHC01521W</td>
<td>Error from modulename: FUNC=function RC=XXX RSN=XXX</td>
<td>An error was encountered while capturing records in the IMS control region.</td>
<td>The IMS Online Reorganization Facility job abends.</td>
<td>Report the problem to IBM Software Support.</td>
</tr>
<tr>
<td>IHC01522E</td>
<td>DBRC request request for DBD dbname DDN dbname failed, RC=XXX</td>
<td>A DBRC request failed.</td>
<td>The IMS Online Reorganization Facility job abends.</td>
<td>Review the return and reason code listed in the IMS Messages and Codes, Volume 4: IMS Component Codes to see whether it can be corrected. The IMS Online Reorganization Facility job can be restarted and as part of that restart. The DBRC request is also retried. If the error cannot be corrected, report the problem to IBM Software Support.</td>
</tr>
<tr>
<td>IHC01523E</td>
<td>DBRC request request for LOGDSN datasetname failed, RC=XXX</td>
<td>A DBRC request for a log data set failed.</td>
<td>The IMS Online Reorganization Facility job abends.</td>
<td>Review the return and reason code listed in the IMS Messages and Codes, Volume 4: IMS Component Codes to see whether it can be corrected. The IMS Online Reorganization Facility job can be restarted and the DBRC request will be retried. If the error cannot be corrected, report the problem to IBM Software Support.</td>
</tr>
<tr>
<td>IHC01524I</td>
<td>The following jobs jobs currently have DBD dbname allocated:</td>
<td>This message is issued with messages IHC01525I and IHC01526I.</td>
<td>None.</td>
<td>None. This message is informational.</td>
</tr>
<tr>
<td>IHC01525I</td>
<td>Jobname jobname IMSID imsid Reg# regionnumber</td>
<td>This message is issued with messages IHC01524I and IHC01526I.</td>
<td>None.</td>
<td>None. This message is informational.</td>
</tr>
<tr>
<td>IHC01526I</td>
<td>The jobs must be stopped before jobname can continue</td>
<td>This message is issued in conjunction with messages IHC01524I and IHC01525I when an IMS Online Reorganization Facility job needs to run the /STOP or /DBRECOVERY command on a DBD and it is unable to do so until the jobs that are listed relinquish control of the DBD. Message IHC01526A follows this message.</td>
<td>None.</td>
<td>None. This message is informational.</td>
</tr>
<tr>
<td>IHC01526A</td>
<td>(jobname): Waiting for BMPs (imsid): Reply 'RETR' or 'CANCEL'</td>
<td>This WTO message is issued after messages IHC01524I, IHC01525I, and IHC01526I. This message indicates that IMS Online Reorganization Facility is trying to issue the /STOP or /DBRECOVERY command on a DBD but the attempt to do so is failing because active BMP jobs exist.</td>
<td>None.</td>
<td>None. This message is informational.</td>
</tr>
</tbody>
</table>

*Chapter 29. Messages and codes 345*
**IHC01527W**  
Reply is an invalid reply  

**Explanation:** An invalid response to the previous WTOR was entered.  

**System action:** The previous WTOR is reissued.  

**User response:** Enter a valid reply.

---

**IHC01528I**  
The reply was reply  

**Explanation:** This informational message indicates the operator response for the previous outstanding WTOR.  

**System action:** Processing continues according to reply.  

**User response:** None. This message is informational.

---

**IHC01529W**  
Takeover processing delayed due to TAKEOVER(Delay)  

**Explanation:** TAKEOVER(Delay) was specified in the control statements. The IMS Online Reorganization Facility job ends without doing takeover processing.  

**System action:** Restart information is saved and the IMS Online Reorganization Facility job ends with RC=4. The original database is left in a DB running state with PROHIBIT AUTH status set in DBRC.  

**User response:** Determine the reason for delaying the takeover processing. When takeover processing is required, resubmit the IMS Online Reorganization Facility job with TAKEOVER(YES) specified.

---

**IHC01530I**  
Datasetname allocated to DD ddname  

**Explanation:** During restart, IMS Online Reorganization Facility allocates certain DD statements to the data sets that were in effect at the time the previous job terminated. If the ddname is in the JCL that is used to restart the job, it will be deallocated first.  

**System action:** Processing continues.  

**User response:** None. This message is informational.

---

**IHC01531I**  
T0request request for DBD dbdbname  

**Explanation:** This informational message indicates that IMS Tools System Interface requests have been made and have completed to an IMS control region.  

**System action:** Processing continues.  

**User response:** None. This message is informational.

---

**IHC01532I**  
Restart information save  

**Explanation:** This informational message is issued before takeover processing to indicate that restart information has been successfully saved in the IMS Online Reorganization Facility restart data set.  

**System action:** Takeover processing begins unless TAKEOVER(DELAY) was specified.  

**User response:** None. This message is informational.

---

**IHC01533W**  
T0request request for DBD dbdbname action RC=xxxx RSN=xxxx  

**Explanation:** An IMS Tools Online System Interface request to the target IMS subsystem failed.  

**System action:** The IMS Online Reorganization Facility job abends.  

**User response:** Review the IMS Tools Online System Interface return and reason codes. There might also be additional messages in the IMS control region that describe why the request failed. The most likely cause is the status of a DBD in the IMS control region. If the problem can be corrected, resubmit the IMS Online Reorganization Facility job. If you are unable to correct the problem, contact IBM Software Support.

---

**IHC01534E**  
Unknown segment segmentname in change record  

**Explanation:** When applying captured change records to the shadow data sets, IMS Online Reorganization Facility encountered a change record that involved a segment that is not in the DBD for which the IMS Online Reorganization Facility job is running.  

**System action:** The IMS Online Reorganization Facility job abends.  

**User response:** Verify that the DMB used in the online IMS subsystems corresponds to the DBD for which the IMS Online Reorganization Facility job is running.

---

**IHC01535E**  
DBRC command request for Partition partitionname failed, RC=xxxx  

**Explanation:** A DBRC request failed.  

**System action:** The IMS Online Reorganization Facility job abends.  

**User response:** Review the return and reason code listed in the *IMS Messages and Codes, Volume 4: IMS Component Codes* to determine whether it can be corrected. You can restart the IMS Online Reorganization Facility job and the DBRC request will be retarted. If the error cannot be corrected, report the problem to IBM Software Support.
Online Reorg Facility Version

Explanation: This informational message indicates the version, release, maintenance date, and maintenance time of IMS Online Reorganization Facility for the job or IMS control region.

System action: Processing continues.

User response: None. This message is informational.

Change capture already active for DBD

Explanation: An IMS Online Reorganization Facility job for a DBD was started and there is already an IMS Online Reorganization Facility job for the same DBD active in the IMS subsystem. Only one IMS Online Reorganization Facility job can be executing for a DBD or HALDB partition.

System action: The duplicate IMS Online Reorganization Facility job abends.

User response: Wait for the current IMS Online Reorganization Facility job for the DBD to end before resubmitting a subsequent IMS Online Reorganization Facility job if it is still needed.

AOI command request to insid failed,

Explanation: An IMS Online Reorganization Facility request that was issued to the IMS Tools Online System Interface failed.

System action: The IMS Online Reorganization Facility job terminates.

User response: Review the return codes for IMS Tools Online System Interface. If the return and reason codes indicate a problem that you can fix in the IMS subsystem, correct the problem and resubmit the job. If you cannot fix the problem, contact IBM Software Support.

Caught up with applying changes, waiting for TAKEOVER Window

Explanation: IMS Online Reorganization Facility has reached the point in the job where it can begin the takeover process. A TAKEOVER window was specified and the begin takeover time has not been reached.

System action: IMS Online Reorganization Facility begins idling until the begin takeover time that was specified is reached. During this idling time, any changes being applied to the original database are still captured and applied to the shadow database.

User response: If no action is taken, IMS Online Reorganization Facility begins the takeover process at the specified time. If you want the takeover performed sooner, you can enter a TAKEOVER command with an MVS MODIFY command. This will cause the takeover process to begin.

Takeover window has expired

Explanation: The end time that was specified by the TAKEOVER_WINDOW parameter has been reached and IMS Online Reorganization Facility is not yet ready to perform takeover processing for the job.

System action: IMS Online Reorganization Facility will take the action specified by the endaction operand of the TAKEOVER_WINDOW parameter.

User response: If WTOR was specified as the action to take when the takeover window expired, determine the action that you want to take and reply to message IHC01544I accordingly.
**IHC01544I**  Specify action to take: N - keep idling until next window, T - do TAKEOVER when ready, C - continue until TAKEOVER then ask again, A - abend job

**Explaination:** The specified TAKEOVER window has expired and WTOR was specified as the action to take.

**System action:** The IMS Online Reorganization Facility job continues processing and waits for an operator response.

**User response:** Determine the action that you want to take at this time and enter a response through the operator console. The options are:

- **N** Indicates that the IMS Online Reorganization Facility job continues processing and tries to perform takeover during the same window on the following day.
- **T** Indicates that you want to finish reorganizing the shadow data sets and perform takeover when the job reaches that point.
- **C** Indicates that you want to finish reorganizing the shadow data sets and then ask the operator what to do when the takeover point is reached.
- **A** Indicates that you want to abend the job.

**IHC01545I**  Ready for Takeover

**Explaination:** The IMS Online Reorganization Facility job is ready for takeover.

**System action:** The IMS Online Reorganization Facility job issues IHC01544I and waits for an operator response.

**User response:** See message IHC01544I.

**IHC01546W**  Unable to complete ONLINECHANGE – reason

**Explaination:** The Online Reorganization Facility job detected DBD changes. ONLINECHANGE(Y) was in effect, but IMS Online Reorganization Facility was unable to locate any IMSACBA or IMSACBB data sets to copy new ACBs into.

**System action:** The IMS Online Reorganization Facility job continues takeover processing but ends with RC=4. The databases are left in PROHIBIT AUTH state.

**User response:** The new ACBs have been generated into the IMSACB data set. You must manually copy these ACBs into the appropriate IMSACBA and IMSACBB data sets. After copying the ACBs, reset the PROHIBIT AUTH flag in DBRC. In future runs of IMS Online Reorganization Facility, if all online IMS subsystems are down, you can supply the appropriate IMSACBA and IMSACBB DD statements in the IMS Online Reorganization Facility job to identify the appropriate data sets.

**IHC01547E**  Backout failed for DBD dbname on imsid – original database is in inconsistent state

**Explaination:** When IMS Online Reorganization Facility was replicating changes that were made to the original database in the online IMS subsystems to the shadow database, the online IMS subsystem encountered an error when it attempted to backout some of the changes that were made to the original database.

**System action:** IMS stops the original database and leaves it with some changes that have not been backed out. The IMS Online Reorganization Facility job cannot determine which changes should or should not be applied to the shadow database, so the job terminates.

**User response:** Correct the problem with the original database and resubmit the IMS Online Reorganization Facility job.

**IHC01548E**  Change capture terminated with inflight changes

**Explaination:** IMS Online Reorganization Facility is terminating the apply phase; however, it cannot determine if some of the captured changes should be committed or if they should be backed out.

**System action:** The IMS Online Reorganization Facility job terminates abnormally.

**User response:** This is likely a logic error in IMS Online Reorganization Facility. Contact IBM Software Support.

**IHC01549E**  Unexpected return code (return code) from USEREXIT

**Explaination:** During the apply phase, IMS Online Reorganization Facility received the return code from the USEREXIT that was specified in the RELOAD.USEREXIT keyword. The return code is an invalid return code from the USEREXIT.

**System action:** The IMS Online Reorganization Facility job terminates abnormally.

**User response:** Correct the problem with the USEREXIT that caused it to return the invalid return code, and resubmit the IMS Online Reorganization Facility job.

**IHC01550E**  USEREXIT requested source segment (segment name) in HALDB database be deleted

**Explaination:** During the apply phase, the USEREXIT...
returned a return code that requested a source segment to be deleted. For HALDBs, IMS Online Reorganization Facility does not support deletion of index source segments.

System action: The IMS Online Reorganization Facility job terminates abnormally.

User response: Either resubmit the job without the USEREXIT, or change the USEREXIT so that it does not delete any index source segments.

IHC01551E Another ORF job already running for DBD dbdname

Explanation: IMS Online Reorganization Facility is already running for this DBD.

System action: The process terminates.

User response: None.

IHC01552E Error saving information in restart data set

Explanation: The restart data could not be saved.

System action: The process terminates.

User response: Contact IBM Software Support.

IHC01553E nn status code from nnnn call to HSSR for segment segname

Explanation: An unexpected status code was received.

System action: The process terminates.

User response: Contact IBM Software Support.

IHC01554E dsname not found in catalog

Explanation: A database data set that should have been in the catalog was not found.

System action: The process terminates.

User response: Contact IBM Software Support.

IHC01555E Information in catalog for dsname is different

Explanation: This message is followed by messages IHC01556E and IHC01557E. The allocated and cataloged data sets have discrepancies.

System action: The process terminates.

User response: Contact IBM Software Support.

IHC01556E type in catalog is/are: text

Explanation: See message IHC01555E.

System action: The process terminates.

User response: Contact IBM Software Support.

IHC01557E type in this job is/are: text

Explanation: See message IHC01555E.

System action: The process terminates.

User response: Contact IBM Software Support.

IHC01558W DBs are left in prohibit auth state

Explanation: The takeover process did not finish completely. The databases are left in a prohibit authorization (PROHIBIT AUTH = ON) state.

System action: The process ended.

User response: Complete one of the following tasks depending on the situation:

- If the takeover was delayed because you specified TAKEOVER(Delay), specify the RESTART(AUTO) keyword and resubmit the job. The process restarts from the Takeover phase to finish the job.
- If the takeover is incomplete because you specified ONLINECHANGE(N), see the preceding IHC01578W message.
- If the takeover is incomplete because of an error, correct the error. Then specify the RESTART(AUTO) keyword and resubmit the job or manually complete the remaining tasks.

IHC01559E DBD dbdname is not in prohibit auth state

Explanation: The specified DBD should still be in prohibit authorization state. However, it is not in that state anymore. The previous IMS Online Reorganization Facility operation did not finish normally.

This problem is probably detected during an IMS Online Reorganization Facility restart operation during which the database was activated by an operator command. The database might have been updated in between two IMS Online Reorganization Facility runs. The current status of the database is not known.

System action: The process terminates.

User response: Collect all output, including output from the previous runs, and check the IMS messages from the CTL and DLISAS region for information that might be related to this database. A time stamp recovery also might be necessary. Contact IBM Software Support.

IHC01560E SORT | IDCAMS failed, Return code is retcode

Explanation: The call to the specified utility failed. Additional error messages are issued.

System action: The process terminates.

User response: Refer to the additional messages for
information about how to resolve this error.

**IHC01561E**  RESTART data set can only be single volume

**Explanation:**  The file allocation of the restart data set is invalid.

**System action:**  The process terminates.

**User response:**  Resubmit the IMS Online Reorganization Facility restart data set definition.

**IHC01562E**  Unable to locate RECON data sets

**Explanation:**  The RECON data sets were not allocated.

**System action:**  The process terminates.

**User response:**  Provide RECON data sets by:

- Specifying RECONx DDs, or
- Specifying IMSDALIB that has the RECON MDAs

**IHC01563E**  Captured change records out of sequence

**Explanation:**  A sequence error was detected when the captured change records were applied.

**System action:**  The process terminates.

**User response:**  Resubmit the job and contact IBM Software Support to inform them about this error.

**IHC01564E**  Unable to inscmd DBD dbdname on IMS imsid

**Explanation:**  The specified IMS command failed on the indicated IMS system for this DBD. This message is followed by message IHC01565A and a WTOR to ask for an action.

**System action:**  Respond to the reply message.

**User response:**  Act on the WTOR.

**IHC01565A**  text

**Explanation:**  This message follows message IHC01564W and is self-explanatory.

**System action:**  None.

**User response:**  None.

**IHC01566E**  DBD dbdname has been allocated by unknown subsystem since IMS Online Reorganization facility stopped capturing changes

**Explanation:**  The database has been allocated since the changes have been applied, but before the takeover process has been started.

**System action:**  The function ends abnormally.

**IHC01567E**  Not all DBDs defined to imsid

**Explanation:**  The database definition in the online system is inconsistent. The primary DBD or some index DBDs might be missing.

**System action:**  The process terminates.

**User response:**  Correct the IMS GEN.

**IHC01568I**  DBDs not defined to imsid

**Explanation:**  The requested DBDs are not defined in the specified IMS system.

**System action:**  The XCF connection to this IMS system is terminated.

**User response:**  None. This message is informational.

**IHC01569I**  DLI batch job: jobname active

**Explanation:**  IMS Online Reorganization Facility detected that the database to reorganize is being accessed by a DLI batch job. This message is followed by message IHC01569A or IHC01571E.

**System action:**  None.

**User response:**  See the subsequent message IHC01569A or IHC01571E.

**IHC01569A**  (jobname): Waiting for DLI batch jobs to finish. Reply 'C' to cancel

**Explanation:**  This WTOR message is issued after message IHC01569I. This message indicates that the IMS Online Reorganization Facility job is waiting for the DLI batch jobs to stop.

**System action:**  Processing waits for the DLI batch jobs to stop. During the wait, the IMS Online Reorganization Facility job makes retry attempts in the background and deletes the WTOR message after all the DLI batch jobs stop.

**User response:**  Take one of the following actions:

- Type 'C' to cancel the reorganization job.
- Type nothing and wait for the listed DLI batch jobs to end, or manually end the DLI batch jobs so the reorganization job can proceed.

**IHC01570E**  Waiting for BMPs has expired

**Explanation:**  The number of retry attempts to stop BMP jobs reached the maximum number set by the BMPRETRY parameter in the base configuration module.
System action: The process terminates.

User response: Wait for the BMP jobs that are shown in the WTO message to end, and resubmit the job. You can increase the maximum number of retry attempts by specifying the BMPRETRY parameter in the base configuration module. If the database is used by long-running BMP jobs, consider enabling BMP job pause handling. For more information, see the topic "BMP pause feature" in the IMS Database Solution Pack IMS Online Reorganization Facility User’s Guide.

IHC01571E Waiting for DLI batch jobs has expired

Explanation: The number of retry attempts to wait for DLI batch jobs reached the maximum number set by the DLIRETRY parameter in the base configuration module.

System action: The process terminates.

User response: Wait for the DL/I batch jobs that are shown in the preceding message IHC01569I to end, and resubmit the IMS Online Reorganization Facility job. You can increase the maximum number of retry attempts by specifying the DLIRETRY parameter in the base configuration module.

IHC01572E Waiting for database pause has expired

Explanation: An attempt to pause the database has timed out.

System action: The process terminates.

User response: Resubmit the job. If the error persists, contact IBM Software Support.

IHC01573E Logical DBD dbname is not specified by the LOGICALDBD keyword.

Explanation: The database was updated by using a PCB that references the indicated logical DBD, but the LOGICALDBD keyword does not specify the logical DBD. Any logical DBD that applications use for update during the reorganization must be specified with the LOGICALDBD keyword.

System action: Processing terminates.

User response: Use the LOGICALDBD keyword to specify the logical DBD.

IHC01574E DBD dbname has insert, delete, or replace rules that are not supported.

Explanation: Unsupported path type is specified in the insert, delete, or replace rules of the indicated DBD. For HALDB databases that have internal logical relationships, the following restrictions apply to the insert, delete, and replace rules. The rules are specified by using the RULES= keyword of a SEGMENT statement in the physical DBD.

IHC01575E Insert rules L and V are not supported for logical parents. Only the P insert rule is supported.

User response: None. You cannot use IMS Online Reorganization Facility to process a HALDB DBD with these rules.

IHC01576E Internal message buffer allocation failed

Explanation: IMS Online Reorganization Facility cannot allocate the internal message buffers for receiving XCF messages.

System action: Processing terminates.

User response: Adjust the size of the internal message buffers by specifying the XCFMAX parameter in the base configuration module. For more information, see the topic "Base configuration parameters" in the IMS Database Solution Pack IMS Online Reorganization Facility User’s Guide.

IHC01577E Internal message buffers are exhausted

Explanation: The internal message buffers that receive XCF messages are full.

System action: Processing terminates.

User response: Increase the size of the internal message buffers by specifying the XCFMAX parameter in the base configuration module. For more information, see the topic "Base configuration parameters" in the IMS Database Solution Pack IMS Online Reorganization Facility User’s Guide.

IHC01578E ONLINECHANGE(N) required because PSBs must be rebuilt

Explanation: The specified DBD change involves changes to PSBs, which requires the ONLINECHANGE(N) keyword.

System action: Processing terminates.

User response: Specify ONLINECHANGE(N) and resubmit the job. After the job ends, perform any required post-processing tasks.

IHC01578W IMS online change is required

Explanation: IMS Online Reorganization Facility updated the staging ACB library with the new database definition, but the ACBs in online IMS subsystems are not updated.

System action: Processing continues.
User response: Use the Online Change Copy utility to copy the ACBs from the staging ACB library to the inactive ACB library. Then perform a standard IMS online change.

IHC01579W Unable to change DEBUG status: reason

Explanation: IMS Online Reorganization Facility could not change the DEBUG state for the target IMS subsystem.

System action: The previous DEBUG state is still in effect for the target IMS subsystem.

User response: Correct the condition reported in the message and resubmit the job.

IHC01580W Database database was not allocated on STA request

Explanation: Database was started on online IMS subsystems, but the database was not allocated due to a dynamic allocation failure. The database will be allocated when it is scheduled.

This message might be issued when a job step that follows the IMS Online Reorganization Facility job step attempts to statically allocate the database data sets. After the database data sets are reorganized, the job will have exclusive use of the database data sets until the job completes. Therefore, even if DISP=SHR is specified in the allocation job step, the online IMS subsystem fails to allocate the database data sets.

System action: Processing continues.

User response: If static allocation of database data set is requested in a subsequent job step, remove that job step and included it in another job.

IHC01581E Number of online IMS subsystems has changed

Explanation: The IMS Online Reorganization Facility job detected a change in the number of online IMS subsystems during reorganization. See the preceding IHC01515W message to determine whether IMS subsystem was started or terminated.

System action: Processing terminates.

User response: If you do not plan to start or shutdown any online IMS subsystems, resubmit the IMS Online Reorganization Facility job. If you received this message as a result of an error or a sudden termination of the IMS control region, correct the problem in the IMS control region and resubmit the IMS Online Reorganization Facility job.

IHC01582W One of dual image copy data sets was not created

Explanation: Dual image copy was requested but either the primary or the secondary image copy data set was not created due to an error in the image copy process.

System action: Processing continues.

User response: To create another copy, use the Create Image Copy (CRC) function of IMS High Performance Image Copy. For more information, see the IMS High Performance Image Copy User's Guide.

IHC01583E Unexpected return code from partition selection exit, FUNC=function, RC=rc

Explanation: The partition selection exit returned an unexpected return code. function is the function code, and rc is the return code from the exit.

System action: Processing stops.

User response: Correct the error in the partition selection exit, and resubmit the IMS Online Reorganization Facility job.

IHC01601I Notification for dbname [partname] has been sent to AD server

Explanation: A system notification was sent to the Autonomics Director server.

System action: Processing continues.

User response: None. This message is informational.

IHC01602W Notification failed, FUNC=function, RC=rc, RSN=rsn

Explanation: IMS Online Reorganization Facility failed to send the system notification to the Autonomics Director server.

System action: Processing continues.

User response: Ensure that the Autonomics Director server, which the ADXCFGRP keyword specifies, is started correctly.

IHC01603W Notification to AD was canceled: reason

Explanation: IMS Online Reorganization Facility did not send a system notification to the Autonomics Director server.

System action: Processing continues. But the system notification to the Autonomics Director server is not sent.

User response: Complete one of the following tasks depending on the reason indicated in the message:

IAVNTFY0 LOAD FAILURE

Ensure that the SHKTLOAD library of IMS
Tools Base is specified in the STEPLIB concatenations correctly.

**INITIALIZATION FAILURE**
Ensure that the XCF group name specified with the ADXCFGGRP keyword is correct and the specified Autonomics Director server is correctly configured.

**UNSUPPORTED FUNCTION**
Ensure that all the conditions for using the ADXCFGGRP keyword are met. For the conditions, see the topic "ADXCFGGRP keyword" in the IMS Database Solution Pack IMS Online Reorganization Facility User’s Guide.

**INTERNAL ERROR**
Contact IBM Software Support.

**IHC01604I** Utility history data for dbname [partname] stored

**Explanation:** IMS Online Reorganization Facility stored the utility history data for the indicated resource in the IMS Tools KB Sensor Data repository. If the database is a full-function database, dbname is the name of the database. If the database is a HALDB, dbname is the master DBD name, and partname is the name of the HALDB partition.

**System action:** Processing continues.

**User response:** None. This message is informational.

**IHC01605W** An error occurred in HKTEXST call, 
FUNC=function, RC=rc, RSN=rsn

**Explanation:** An error occurred during the IMS Tools KB job statistics API (HKTEXST) call. function is the function code, and rc and rsn are the return code and the reason code from the API.

**System action:** Processing continues, but the utility history data is not stored in the IMS Tools KB Sensor Data repository.

**User response:** Contact IBM Software Support.

**IHC01606W** Utility history data process was canceled. REASON: reason

**Explanation:** An error occurred and the utility history data process was canceled. reason is one of the following texts:
- initialization FAILURE
- INTERNAL ERROR

**System action:** Processing continues, but the utility history data is not stored in the IMS Tools KB Sensor Data repository.

**User response:** Complete one of the following tasks depending on the reason:

**IHC01611I** Reorganization starts for [database dbname 1 partition partname]

**Explanation:** The Conditional Reorganization Support Service (CRSS) determined that the database or the partition requires a reorganization. IMS Online Reorganization Facility is starting the reorganization process for the indicated database or partition.

**System action:** Processing continues.

**User response:** None. This message is informational.

**IHC01612I** Reorganization is not needed for [database dbname 1 partition partname]

**Explanation:** The Conditional Reorganization Support Service (CRSS) determined that the indicated database or partition does not need to be reorganized.

**System action:** The IMS Online Reorganization Facility process ends without reorganizing the database or the partition.

**User response:** None. This message is informational.

**IHC01613I** One or more exceptions are detected in [database dbname 1 partition partname]

**Explanation:** The Conditional Reorganization Support Service (CRSS) evaluated the indicated database or partition and detected one or more database exceptions.

**System action:** Processing continues.

**User response:** Complete the following steps:
1. Check the output from the job.

   If the reorganization policy was customized to send exception notification messages by using the z/OS WTO service, the TSO/E SEND command, or through email or texting, check those messages that are issued by the job. Those messages include a job summary message (policy evaluation summary message) that summarizes the result of policy evaluation performed by Policy Services during the job.

   For more information, see the topic "Checking the policy evaluation summary message" in the IMS Database Reorganization Expert User’s Guide. For information about finding the Diagnosis report that was created by this job and that is stored in the Output repository of IMS Tools Knowledge Base, see the topic "Finding Diagnosis reports by using
report completion codes" in the IMS Database Reorganization Expert User’s Guide.

2. Check the exceptions in the Diagnosis report and identify the cause of the database exceptions.

For more information, see the topic "Identifying the cause of database exceptions" in the IMS Database Reorganization Expert User’s Guide.

IHC01614I  No exception is detected in [database dbname | partition partname]

Explanation: The Conditional Reorganization Support Service (CRSS) evaluated the indicated database or partition and detected no database exceptions.

System action: Processing continues.

User response: None. This message is informational.

IHC01615I  [Database dbname | Partition partname] is reorganized because REORGDIAG is specified

Explanation: The indicated database or partition is reorganized regardless of its status because the job runs in Reorganization Diagnosis mode.

System action: Processing continues.

User response: None. This message is informational.

IHC01616W  An error occurred in BBECRAPI call, FUNC=function, RC=rc, RSN=rsn

Explanation: An error occurred during the Conditional Reorganization Support Service (CRSS) API call. function is the function code, and rc and rsn are the return code and the reason code from the API.

System action: Processing continues.

User response: The detail of the error or the warning is recorded in the journal messages that are issued by the CRSS. Correct the error and rerun the job.

IHC01617W  Conditional Reorg service is deactivated.

REASON: reason

Explanation: The Conditional Reorganization Support Service (CRSS) cannot continue the job because of the reason displayed.

System action: The processing continues without the CRSS. If the job runs in either Conditional Reorganization mode or Reorganization Diagnosis mode, IMS Online Reorganization Facility reorganizes the database regardless of database status, but the CRSS does not generate a Diagnosis report.

User response: Complete one of the following tasks depending on the reason:

BBECRI00 LOAD FAILURE
Ensure that the IMS Database Reorganization Expert load library is included in the STEPLIB concatenation.

INITIALIZATION FAILURE
Ensure that Policy Services is configured correctly.

ITKB INIT FAILURE
Ensure that the SHKTLOAD library of IMS Tools Base is included in the STEPLIB concatenation and that the Tools KB Server is configured correctly.

ITKBSERVER NOT SPECIFIED
Ensure that the ITKBSERVER keyword is specified in the HRFSYSIN DD statement.

UNSUPPORTED FUNCTION
Ensure that all of the following conditions for the CRSS are met:
- The primary database to reorganize is a full-function database or a single partition of a HALDB.
- The primary database to reorganize is not an index database.
- TAKEOVER(N) is not specified in the REORG HRFSYSIN DD statement.
- DBD change is not requested.

INTERNAL ERROR
Contact IBM Software Support.
Chapter 30. Gathering diagnostic information

Before you report a problem with IMS HALDB Toolkit to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all IMS HALDB Toolkit 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
- The trace file or trace panel. For more information, see "Trace file or trace panel" on page 323.
- 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
Part 11. Appendixes
Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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.

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 Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

**Trademarks**

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).
Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

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 322
accessibility 12
ANALYZE PART 15
command parameters 24
data flow 16
DBRC commands 37
DD statements 21, 49
examples 33
simulating conversion 17
simulating repartitioning 18
simulating repartitioning with unloaded data sets 19
unloaded data set 39
ANALYZE PART command parameters 24
API, partition selection 207
application support
creating a DFSHALDB statement 217
deleting a single partition 229
loading a single partition 223
loading logical children (PROCOP=L) 215
merging two DBDs into one DBD 235
partition selection API 207
testing partition selection exits 209

B
BACKUPDBRC
command parameters 114
DD statements 111
batch messages 324
batch update
creating rules for substituting HALDB data set names 193
maintaining HALDB without DBRC 193
substituting HALDB data set names 193

C
CHECK
command parameters 134
DD statements 131
codes 322
configuration
sample library file contents (SIHCSAMP) 315
usage and restrictions 8
CONVERT 45
command parameters 53
converting databases offline 47
data flow 46
examples 68
CONVERT command parameters 53
converting databases offline 47
converting to HALDB
ISPF user interface 279
offline 279
selecting DBDs 279
cookie policy 359, 361
COPYDBRC
command parameters 120
DD statements 118
creating a DFSHALDB statement 217

data flow
ANALYZE PART 16
CONVERT 46
MAINTAIN 76
data set names
creating rules for substituting HALDB data set names 193
creating unrelated data set names 193
substituting 183
data sets
converting non-HALDB to HALDB 202
database utilities, ISPF user interface
database utility functions
summary 287
DBDs
converting non-HALDB to HALDB 201
DBRC enhancement utilities 109
cloning DBRC definitions for test environments 115
copying HALDBs to a different RECON 123
DBRC request
specifying 180
DBRC support functions, ISPF user interface
backing up DBRC DBD definitions 297
cloning DBRC definitions 293
copying DBRC definitions 295
showing partition definitions 292
deleting a single partition 229
DFSHALDB statement, creating 217
diagnostic information
gathering 355
documentation
accessing 10
sending feedback 10

E
evironment, IMS 245
error messages 321

global tables 183

H
HALDB Analyzer 129
HALDB utilities
creating unregistered data sets 179
DBRC enhancement utilities 109
HALDB Analyzer 129
healing index pointers 161
recovery 155
reorganizing a PSINDEX 143
simulating HALDB without DBRC 179
splitting an unload file into single partition files 149
system utilities 167
HALDB without DBRC
installing 180
HALDBCOPY
command parameters 240
DD statements 237
HALDBLOAD
command parameters 227
DD statements 224
HALDBs
 copying to different IMS 125
healing index pointers 161

I
IHC ECLON 115
IHECCOPY 123
IHECDFS 125
IHCESAM1
SYSTEM table 194
IHCESAM2
USER table of SYSTEM IVP1 195
IHCESAM3
IMSID table of SYSTEM IVP1 196
IHCESAM4
DBD table 197
IHCESAM5
DBDLIB table of SYSTEM IVP1 198
IHCESAM6
IMSLIB table of SYSTEM IVP1 199
IHCUDBD
command parameters 171
DD statements 169
IHCUDFSH
command parameters 222
DD statements 219
IHCUSPLT
command parameters 154
DD statements 151
IHCWPSBL
DD statements 173
IMSID table
maintaining 188
ISPF messages 323
ISPF user interface
converting to HALDB 277, 278, 279
overview (continued)
creating rules for substituting HALDB data set names 184
database utilities 287
DBRC support functions 291
maintaining HALDB 281
maintaining HALDB without DBRC 184
other utilities 289
processing a project 259
setting up 244
setting up the environment 243
starting 244
substituting HALDB data set names 184
ISPF user interface configuration
creating a new IMS environment 245
creating projects 255
deleting an environment 257
legal notices
cookie policy 359, 361
notices 359
programming interface information 359
trademarks 359, 360
loading a single partition 223
loading logical children (PROCOPT=L) 215

MAINTAIN 75
command parameters 82
data flow 76
DD statements 79
examples 102
how it processes partitions 97
maintaining databases offline 77
MAINTAIN command parameters 82
maintaining databases offline 77
maintaining HALDB using MAINTAIN 75
maintaining HALDB, ISPF user interface
selecting and maintaining partitions 285
selecting DBDs for maintenance 284
splitting or consolidating HALDB partitions 282
merging two DBDs into one DBD 235
messages 321

notice 359

overview 1, 3
accessibility 12
application support 4
benefits 4
converting to HALDB 4
overview (continued)
DBRC handling 4
HALDB Analyzer 4
maintaining HALDBs 4
simulating HALDBs 4
system utilities 4
what does the product do 3

P
partition boundaries
creating partitions automatically 274
specifying high keys 272
using a partition selection exit for conversion 269
partition selection API 207
partition selection exit
changing the DBDs 271
creating partition boundaries 272
functions provided by an exit 268
IHCPSELO exit 269
locating partition boundaries 272
specifying additional conversion options 275
problems
diagnostic information about 355
programming interface information 359
project processing
creating and submitting JCL 265
project work panel 262
selecting a project 260
SYNC points 266
using a partition selection exit 268
project work panel
ask me later 264
restarting at the current or prior level 262
restarting from the beginning 262
selecting setup parameters 262
showing project status 262
projects 255
PSEXIT
command parameters 213
DD statements 210
PTRHEAL
command parameters 166
DD statements 163

R
reader comment form 10
RECON table
maintaining 188
recovery 155
reference
sample reports 301
reorganizing a PSINDEX 143
REORGIND
command parameters 148, 233
DD statements 145, 230
reports 301
Collect DBDs report 304
HALDB Analyzer report 308
Partition Selection Exit report 314
Runtime Messages report 302
Runtime Options In Effect report 306
reports (continued)
Upload report 307
RESETILK
command parameters 160
DD statements 157
return codes 322

S
sample library file contents (SIHCSAMP) 315
service information 9
SIHCSAMP sample library 315
simulating conversion 17
simulating databases 18
simulating HALDB using ANALYZEPART 15
ANALYZEPART command parameters 24
data flow 16
DD statements 21, 49
examples 33
repartitioning using unloaded data sets 19
simulating conversion 17
simulating databases 18
splitting an unload file into single partition files 149
support
required information 355
support information 9
system definition
modifying 185
system selection method
modifying 186
system utilities 167
creating a PSB list for selected DBDs 172
creating OSAM multi-volume data sets 175
listing VIO and unit names 176
rebuilding DBD source from DBDLIB 168
SYSTEMs
maintaining 189, 190, 191

tables
preinstalled 183
 technote 10
testing partition selection exits 209
trace file or trace panel 323
trademarks 359, 360
troubleshooting
batch messages 324
ISPF messages 323
return and abend codes 322
trace file or trace panel 323

U
unregistered data sets
global tables 183
installing HALDB without DBRC 180
unregistered data sets (continued)
  naming conventions  183
usage and restrictions
  database limitations  8
  logical relationships and secondary indexes  8
USERs
  maintaining  192